Automated Discovery and Analysis of Social Networks from Threaded Discussions

Anatoliy Gruzd, Caroline Haythornthwaite

Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, agruzd2@uiuc.edu, haythorn@uiuc.edu

Abstract

To gain greater insight into the operation of online social networks, we applied Natural Language Processing (NLP) techniques to text-based communication to identify and describe underlying social structures in online communities. This paper presents our approach and preliminary evaluation for content-based, automated discovery of social networks. Our research question is: What syntactic and semantic features of postings in a threaded discussions help uncover explicit and implicit ties between network members, and which provide a reliable estimate of the strengths of interpersonal ties among the network members? To evaluate our automated procedures, we compare the results from the NLP processes with social networks built from basic who-to-whom data, and a sample of hand-coded data derived from a close reading of the text.

For our test case, and as part of ongoing research on networked learning, we used the archive of threaded discussions collected over eight iterations of an online graduate class. We first associate personal names and nicknames mentioned in the postings with class participants. Next we analyze the context in which each name occurs in the postings to determine whether or not there is an interpersonal tie between a sender of the posting and a person mentioned in it. Because information exchange is a key factor in the operation and success of a learning community, we estimate and assign weights to the ties by measuring the amount of information exchanged between each pair of the nodes; information in this case is operationalized as counts of important concept terms in the postings as derived through the NLP analysis. Finally, we compare the resulting network(s) against those derived from other means, including basic who-to-whom data derived from posting sequences (e.g., whose postings follow whose). In this comparison we evaluate what is gained in understanding network processes by our more elaborate analysis.

Keywords

Social networks, named entity recognition, natural language processing, collaborative learning

1. Introduction

Online interactions of all kinds are generating a growing volume of text which creates problems for individuals trying to understand the internal dynamics of communities they may want to join. One set of texts that can benefit from further transparency in processes are those created by online learners. In the U.S. alone, over 3.5 million students were studying online in the Fall of 2006, and 20% of U.S. higher education students took at least one course in Fall 2006 (Allen & Seaman, 2007). The texts generated from all of these online courses represent a rich history of group interaction and an opportunity to explore and understand learner's behaviors in online settings. The problem is how to approach and make sense of the vast amount of data created by these communities and to use that information to help faculty and administration to understand online learning processes and to develop more appropriate and effective programs for the next generation of students. Unfortunately, current methods for processing and analyzing data from such communities are cumbersome, expensive and time consuming.

To address these issues, we are following a social network approach to study interaction patterns in e-learning communities. Interaction is essential in many approaches to networked learning. Although individuals may learn by retrieving information from online archives, dictionaries and encyclopedia, it is interaction with others from around the globe with similar, perhaps narrowly enjoyed, interests that fuels the benefits of networked learning. A social network view is in keeping with notions of collaborative learning, participatory culture, web 2.0, and learning through engagement with others (Bruffee, 1993; Cook & Brown, 1999; Jenkins, 2006; Koschmann, 1996; Lave & Wenger, 1991; Miyake, 2007). Thus, examining social networks – including the roles and positions of actors in a social network, their influence on others, and what exchanges support and sustain the network – is an important goal for understanding networked learning processes.

Gruzd, A. & Haythornthwaite, C. (2008). Automated discovery and analysis of social networks from threaded discussions. *International Network of Social Network Analysis Conference*, St. Pete Beach, FL, USA, January 22-27, 2008.

However, pursuing a social network approach raises a number of methodological issues. The first is how to examine and evaluate the network aspects of networked learning, including identifying what matters in terms of learning in the online interaction space. The second issue is how to do this on a scale that is adequate to give more than anecdotal results, and which keeps pace with the rapid production of text typical of networked learning settings.

To draw interaction data from online texts, two steps are needed. First, some form of automated processing is needed to reduce the large datasets to community and conversational essentials that show the relations of importance to group members; and second, assessment of these data extractions is needed to determine the usefulness and meaning of these measures to participants. When combined, these two aspects can provide useful representations of online conversations, from statistical reports to visualizations of data and interactions, each of which can help networked learners (instructors and students) better understand the social environment in which they are participants. This paper reports on our work addressing these two components. More specifically, we will discuss methodological issues and present our preliminary findings associated with the discovery of social networks from threaded discussions favored in online courses.

2. Dataset

The dataset includes all class-wide postings from bulletin boards for a required class for first term library and information science students taking their Masters degree online. Classes typically meet weekly in 'live' online sessions conducted through a combination of internet supported video, slides, and chat. Bulletin boards are then used over the week for discussion of topics initiated by the instructor. The class-wide bulletin boards are password protected and thus not openly available on the web. Postings from eight iterations of the class are available, each given by the same instructor, two per Fall 15-week term from 2001 to 2004. The eight classes involved 31 to 54 class members, the professor, and 3-4 teaching assistants. Together they posted 1200 to 2100 class-wide messages per term. Students also had small-group bulletin boards in use during these terms, and posted 2-3000 messages a term on these boards, but for privacy reasons these are not part of the dataset. The online learning system used at the time was an in-house application created and supported by the degree-granting school. Beyond bulletin boards, the community was also maintained via other online means, including email, and online chat during live weekly class sessions, and one campus visit per term. Table 1 gives the basic statistics for the four courses. Participants – students, instructor and 4-5 teaching assistants – posted 1207 to 2157 messages on the class-wide bulletin boards during the 15-week period.

Institutional Review Board permission was obtained for this work; procedures included alerting the class to the intended use of the data in the class-wide bulletin boards and describing the intended use. Students were given the option of contacting the researcher directly if they did not want to be directly quoted from the bulletin boards.

Table 1: Basic statistics for class-wide bulletin board postings, eight classes

	F01A	F01B	F02A	F02B	F03A	F03B	F04A	F04B
No. of Messages	1205	1581	1469	1895	1280	1242	1493	2157
No. of Participants	38	47	47	54	54	46	54	52
No. of Bboards	22	22	28	28	25	24	28	27
Avg.No.Symbols/Msg	1073	1056	864	898	1286	953	967	1058
Avg. No. Lines/ Msg	17	14	15	14	17	17	15	16

3. Determining Networks

In analyzing networked learning environments our aim is to make visible the interaction dynamics that are hidden in streams of linear text. Since interaction requires identifying participants, the first problem to solve is how to identify the actors, and then to derive "who is talking to whom." Later we want to add to that "what" they are talking about.

Typically, all that is evident as an overview of the list of bulletin board postings is the email address (or other identifier) of the poster, and the subject line. At first glance this seems to provide a simple mechanism for identifying who is talking to whom – a poster is answering the previous poster. This is one way to build the network, and we refer to this network as the *Chain Network* – built on the way messages chain to each other temporally.

3.1. Building Chain Networks

In constructing networks from the header information, there are still some decisions about the relevance and weight of earlier postings that need to be made. Table 2 presents some options. The overall question is what measure of influence or prominence should previous postings be given in considering the tie between posters. The basis of the social network perspective is consideration of the way each individual's behaviors affect the thinking and behaviors of others. We can readily expect that a first poster will influence postings that follow because of his or her primacy in addressing a topic presented for discussion. This post gets the ball rolling, provides an opinion to respond to, and discussion norms (or instructions) may dictate that subsequent posters pay attention to earlier postings. In promoting participation in online discussion, instructors may actively encourage reflective postings, i.e., discussion boards are

not just sites to submit individual assignments, but are instead places of activity, of to and fro of discussion. Although not all boards in all settings will be used as, or be successful as interactive, participatory discussion spaces, one thing every instructor would want to know if whether they have been used in this way. The chain data alone cannot actually determine this, and we return to ways of interrogating the actual interactive process below. Suffice it to say for now that in building the most representative chain network in what is expected to be a reflective, discussion-based forum, requires some consideration of the relation among previous posts.

In our formulation, we have considered several options. *Option 1 is the naïve solution, creating the network based on counting a tie to be present only between a message poster and the poster of the immediately preceding post.* In the options discussed here, the ties are treated as ***undirected***, i.e., discarding the direction of the connection between actors. In accepting undirected ties, we have reasoned as follows. We assume that an individual who first posts does not set out to influence any particular individual member of the class. Thus, the receiving node for a posting directed "out" from a first poster cannot be determined. Although such information might be in the post itself, for the chain network, which uses header information only, we do not want to assume more than is indicated by the available data, and thus we assume only a general intention to influence or engage with the class as a whole.

A responder does answer an individual, i.e., the previous poster, but we do not know without analyzing the text whether that response is an acknowledgement of influence, a challenge to it, or a completely new point unaddressed to the previous poster. Thus a determination of an intentional, directed tie from the responder to the previous poster cannot be made unequivocally from position in the posting sequence. Thus, we reason instead that juxtaposition is a sufficient, instrumental, indicator of a tie, but direction of the tie cannot reasonably be assumed. Thus a tie "in" to the previous poster is not assumed, only a tie based on sequencing within the message stream. Neither can a tie "out" from a secondary poster to subsequent posters be made for the same reasons as no tie "out" from the first poster is determined. Like the first poster, subsequent posters' influence extends to the class as a whole, and to subsequent, yet-to-be-heard posters as well.

Options 2 and 3 bring in consideration of the chaining effect of streams of posts. *Option 2 considers only the influence of the first poster as the prime mover of the discussion.* A weight of constant value, equal to or less than that assigned to the immediately preceding poster is assigned to the tie. Option 3 considers the influence of all previous posters, using weights of ordered diminishing size to weight the tie between the poster and all previous posters

Table 2: Chain network options

Options	Amy ← Bob ← Cathy ← David		
(1) Connect a sender to the immediately preceding poster only (undirected), e.g., a connection is counted only between poster David and Cathy is counted	0	0	1
(2) Connect a sender to the first (=thread starter) and immediately preceding poster, assigning a weight to both ties, e.g., a connection is counted between poster David & Amy, and David & Cathy.	<=1	0	1
(3) Connect a sender to *all* posters in the reference chain, assigning weights that decrease with 'distance' from the poster (e.g., reducing each by half)	.25	.5	1

3.2. Shortcomings of Chain Networks

These options are our starting point for examining interaction networks. They represent a logical set of criteria for building networks based on the posting chain information only. However, while these procedures provide some approximation of the conversational progress, there are a number of shortcomings of these techniques. In an asynchronous, many-to-many discussion board, individuals may address messages that appear much earlier than the immediately preceding posting, making the chain data a poor estimate of network interaction even if an accurate representation of the chain left in the textual artifact that results from use of the discussion boards. An individual may post in one apparent place in the message sequence, but refer in their message to one of more of the postings preceding their post, or to conversations and discussions that happened outside the discussion board (e.g., in our case, during the live sessions, or in different bulletin boards on the class discussion board). Further, an individual may seem to respond to one post, but in their text refer to several others, synthesizing and bringing together comments of others.

Examples of these kinds of issues taken from our dataset are given in Table 3 (names have been anonymized). The first example shows four individuals named in the text of the message who are directly addressed by the poster (Nick, Ann, Gina, Gabriel), but using only the previous poster information only one name (Gabriel) would be included in the network. The second example shows an unambiguous addressee (Gina) but if the network is built from ties across the entire chain history, extra people would be included in the network (Gabriel, Sam, and Eve as well as Gina); and if Gina is not the immediately preceding poster, a connection might not even be made to her. The

third example names a person who has not posted at all in this thread, and hence would not be identified at all by a chain network.

Table 3: Examples of differences between chain data and text data

Chain	Text
Previous post is by Gabriel, Sam replies:	'Nick, Ann, Gina, Gabriel: I apologize for not backing this up with a good source, but I know from reading about this topic that libraries…'
Previous posts by Gabriel, Sam, Gina, and Eva, then:	'Gina, I owe you a cookie. This is exactly what I wanted to know. I was already planning on taking 302 next semester, and now I have something to look forward to
Post by Fred:	'I wonder if that could be why other libraries around the world have resisted changing – it's too much work, and as Dan pointed out, too expensive.'

Each of these kinds of shortcomings in the chain network leads us to look at the text of the message for more detail on who is talking to whom, and about what. Our second approach uses natural language processing to identify and extract names from the text of the messages in order to build the who-to-whom network.

3.3. Name Networks

Identifying individuals from names within the text is not a straightforward issue for automation. Although we may have a master list of enrolled students, differences between names used and class lists is a common issue: Virginia becomes Gina; Michael John Smith or Michael J. Smith goes by John; Wendy Mason became Wendy Carpenter last term but her record remains in her former name; the instructor is identified as "Professor" rather than by their name; a student acquires and is referred to by a nickname ("JJ", MaryK) students with the same first name start being identified separately at some point in the term (e.g., Mary Kipley and Mary Donnelly both appear as Mary early on, becoming Mary K and Mary D in later posts).

To explore the variations that might be present, we hand coded a bulletin board containing 62 messages. This revealed a number of issues and conventions about the use of names. First, four categories of name use could be distinguished: those referring to participants in the class; those to non-participants, most commonly the author of a work under discussion; names appearing because of errors, e.g., incorrectly spelled names; and names appearing in the copy of an earlier posting appended to the new post. Table 4 gives some further specifics of name uses found within these categories.

Table 4: Name occurrences in bulletin board discussions

Usage	Network Participants
From	Person indicated in 'from' line of heading, always an email address (system generated)
Addressee	Direct reference to other ('I agree with you Todd')
Reference	Indirect reference to other ('Todd has a good point')
Self-Reference	Poster refers to themselves in some way (brain-dead student, high school teacher)
Signature	Name as given by the message author on their post
	Named Non-Participants
Subject	Subject of the discussion, 1-3 parts, e.g. Dewey, Brewster Kahle, Charles R. Darwin
Non-Group	Person not in the group, nor the subject, e.g., a former professor or mentor
	Errors
Error	New name appears because of error (e.g., Lackie as a subject instead of Leckie)
	Previous Posts and Copies
[Previous Posts]	If the previous message is included, indicates the previous poster ('Janice wrote: …') (system generated; could be edited deliberately or accidentally when)
[Copy]	Name appears because it is included within the previous message

Note: For our dataset, postings are first processed to remove copies of earlier messages. However, for some analysis copying behavior may be an aspect for investigation and hence names from copies would matter.

From these observations, after all personal names are found, what needs to be created is an *authority file* that ties name variations, nicknames, and incorrectly spelled names to a single identifier (e.g., email address). These names will play a role of nodes in the name network. Then, we need to find how these names/nodes are connected to each other to derive *who talks to whom* data. And, finally, to make the name network better reflect e-learning processes, we need to assign tie strength based on some pre-defined *relation* that believed to predict the success in e-learning communities. These three basic steps for building the name network is discussed in greater details in the following three sections.

3.3.1. Node Discovery using Personal Names

Previous work on automated name discovery

Personal name discovery from text is part of a broader task called Named Entities Recognition (NER). NER is a set of text mining techniques designed to discover *named entities, connections* and the types of *relations* between them

(Chinchor, 1997). In NER, a *named entity* is defined very broadly. It may be an organization, geographic location, country, etc. NRE is commonly used for the purpose of hiding sensitive data in private or secret documents such as personal medical records and vital government documents. These applications of NRE are usually referred to as anonymisation or pseudonymisation, also known as *automatic de-identification* (e.g., Sweeney, 2002; Medlock, 2006; Uzuner et al., 2007).

There are two primary approaches to find personal names in the text. The first and easiest approach is to compare each word from the text with a look-up dictionary of all possible personal names. If a word is in the dictionary, then it is considered to be a name. Examples of electronic dictionaries with English names include the publicly accessible US Census (http://www.census.gov/genealogy/names/dist.all.last), a commercial database from IBM (http://www-306.ibm.com/software/data/ips/products/masterdata/globalname), and a web resource "Behind the Name" (http://www.behindthename.com). Among researchers who relied on this approach are Harada et al. (2004), Sweeney (2004), Patman & Thompson (2003). While this approach is easy to implement and run, it will leave out names that are not in a dictionary. These may be names of foreign nationalities, informal variations of names, or nicknames. Additionally, this approach does not take into account that in different sentences a word may be a name or just a noun. For example, "*Page* asked for my help" and "Look at the *page* 23". To make sure that an algorithm will find "Page" in the first sentence above and ignore "page" in the second, some researchers may consider only capitalized words as potential candidates for personal names and ignore others. However, this restriction is not very practical with informal texts where names are often not capitalized.

The second, alternative approach to finding personal names does not require maintaining a dictionary of names. Instead, it applies linguistic rules or patterns to the content (e.g. word frequency and context words) and/or sentence structure (e.g. word position) to identify potential names. These linguistic rules/patterns can be derived manually (e.g., Kim & Woodland, 2000) or learned automatically from the corpus (e.g., Chen et al., 2002; Bikel et al., 1997; Day & Palmer, 1997). There are both advantages and disadvantages of using either the manual approach or automatic approach. The manually derived rules are considered to be faster and tend to produce fewer false-positive results than the automatically derived rules/patterns. If carefully compiled, the manually derived rules can be easily applied to a new dataset without a need to retrain an algorithm. However, like the dictionary-based approach, the manually derived rules/patterns are also susceptible to being incomplete and possibly missing some of the names. As for the automatically learned rules/patterns from the corpus, they do not require as much human input, but they are more likely to miscategorize words. They also often require retraining before application to a new corpus.

To improve the accuracy of both methods, they can be used in conjunction with each other. For example, one approach is to find all names based on the dictionary first, and then using linguistic rules/patterns, to find names that are not in the dictionary. Using such a hybrid approach, Minkov et al. (2005) reported 10-20% improvement in accuracy. A downside of a hybrid approach is that by combining various methods we also increase the time needed to run an algorithm.

The next section will describe our approach to personal name discovery. Following Minkov and her colleagues, we used a hybrid approach. But, instead of learning rules automatically, we derived rules manually based on our observations. As shown in the section on evaluation, the use of the manually derived rules can significantly reduce the processing time and reduce the number of false-positive results.

Our approach to personal name discovery

After reviewing and testing different software available to researchers to perform NRE-related tasks, we were not satisfied with their performance in terms of execution speed and accuracy. Furthermore, most of the available programs are trained on documents from newspaper or medical domains. This prompted us to develop our own algorithm. While developing our algorithm, we kept a couple of important criteria in mind. It must be able to 1) process messages in real-time, 2) understand informal online texts, 3) with minimum execution speed. These criteria are especially important to us because we planned to incorporate this algorithm into our Internet Community Text Analyzer (ICTA), a web-based system for automated analysis of online texts (see Haythornthwaite & Gruzd, 2007).

The algorithm works as follows. First, to avoid redundancy and wasted processing time, we removed any texts that belong to previous messages. This was done automatically using a string matching mechanism called *regular expressions*. We accomplished this by removing all lines from the messages that appeared after a pattern "<name> wrote:" and start with a colon ":". Second, we removed so-called "stop-words", such as *and*, *the*, *to*, *of*, etc.. There are many different versions of a "stop-words" list freely available on the Internet. The one we used is part of the Natural Language Toolkit (http://nltk.sourceforge.net), and it includes 571 words. Third, we normalized all remaining words by stripping all special symbols from the beginning and end of any word, including possessives (e.g '--*Nick*' or '*Nick's*' becomes *Nick*).

For all remaining words, to determine whether a word is a personal name, we relied on a dictionary of names as well as on a set of general linguistic rules derived manually. The dictionary that we used includes over 5,000 frequently used personal names and over 88,000 last names as reported by the 1990 US Census (available at http://www.census.gov/genealogy/names). However, since students in the class primarily addressed each other by their first names, we ignored any mentions of last names in the dataset. In addition to the dictionary, we also relied on two additional sources of personal names: a class roster (list of all class participants) (e.g., Matsuo et al., 2006) and the "From" field in the message header (e.g., Culotta et al., 2004). The use of the class roster appeared to be not as effective as we thought originally. This is primarily because students often did not used formal names from the roster to refer to each other, but nicknames and informal names (e.g., *Ren* for *Karen*, *Dan* for *Daniel*). Furthermore, the use of the class roster would limit the ability of the algorithm to perform well on texts produced by groups with unknown membership. The second additional source of names (the "From" field) proved to be more useful. In some cases, in addition to a poster's email address, the "From" field also includes his/her name enclosed within a set of round brackets. To recognize names from the "From" field of the message header, we used a simple string matching pattern that looks for only words found within the round bracket (if any). For example, the following record *"agruzd2@uiuc.edu (Anatoliy)"* will produce *Anatoliy*.

To recognize names that are not in the dictionary yet (e.g., nicknames, abbreviated names, unconventional names, etc) such as *CH* or *CarolineH*, we relied on the context words that usually indicate personal names such as titles (e.g., *Professor*, *Major*, *Ms.*) and greetings (e.g., *Hi* or *Dear*). In the future, we are planning to rely on other types of context words as well such as communication and motion verbs that usually express actions associated with humans (e.g. say, tell, warn, walk, run, etc). Such verbs can be obtained from various lexical resources such as VerbNet, EVCA, and VerbOcean (Chklovski & Pantel, 2004; Klavans & Kan, 1998).

To exclude personal names that are part of building or organization names like the "*Ronald Reagan Presidential Library*", we first ignored all sequences of more than three capitalized words, and second we removed phrases in which the last word was included in a pre-compiled list of prohibited words such as "Street" or "Ave".

Finally, for all words that we identified as potential names, we then attempted to determine the confidence level that the particular word is actually being used as a person's name in the text. This is accomplished by factoring in the commonality of a name in the US Census (if applicable) and whether or not the first character of a word is capitalized. For example, consider the word "*page*". According to the US Census, the name "*Page*" is possessed by 0.034% of the population sample. Therefore, its final score assigned by our algorithm will be $\frac{0.034}{p}$; where p is a parameter that will take a value of 1 if the word is capitalized or a value greater than 1 otherwise. This is done to "punish" non-capitalized words and reduce their confidence level of being a name. In the current version of the algorithm, we adopted a conservative approach by setting the value of p to 10. That is, the final score for non-capitalized "*page*" will be $\frac{0.034}{10} = 0.003$. Since it is less than a pre-set threshold of 0.0099 the word will be removed from the further consideration.

While the algorithm that we have described above is very thorough, it is still not capable of achieving 100% accuracy. This is because at this point in the process, incorrectly spelled names may be missed and some possible false-positive words may still be on the list. However, since accurate name extraction is a vital foundational building block in our primary work on automated inference of social networks, we needed to be as close to the 100% level of accuracy as possible. To reach that level of accuracy, we created a web-enable interface where researchers can also manually review and edit the list of extracted names created by our algorithm (see Figure 1). After running the name extractor, a researcher can use this interface to add names that were missed by the extractor or delete false-positive words. The algorithm will remember these words for future runs as well. To improve the readability of the extracted names, all names are displayed in the form of a *tag cloud*. The larger font size in the tag cloud indicates the higher frequency of occurrences of a particular name in the dataset. Clicking on any names from the tag cloud returns a list that shows all instances where that name was found along with 2-3 words preceding and following the name (see Figure 2), and from there a user can also go to the exact location in the text where a potential name was found. This is especially helpful for uncovering false-positive results. For example, in our experiments we were quickly able to verify that a word "*Mark*", a common name in the English language was not actually a name in that instance, but part of the term "*Mark* Up language".

The end result of our semi-automated name extraction exercise is a list consisting of all occurrences of personal names in the postings.

Figure 1: A web interface for editing extracted names: Top 30 names automatically extracted from the Internet Researchers' listserv for messages posted during October 2002

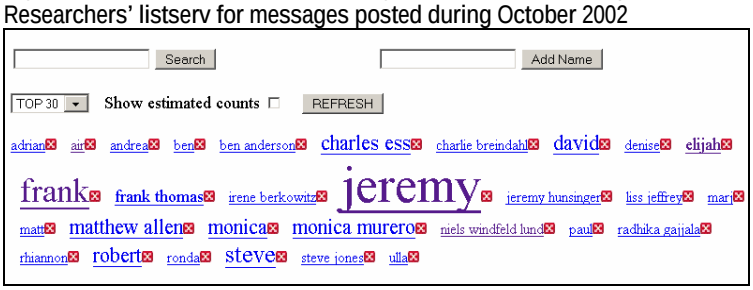
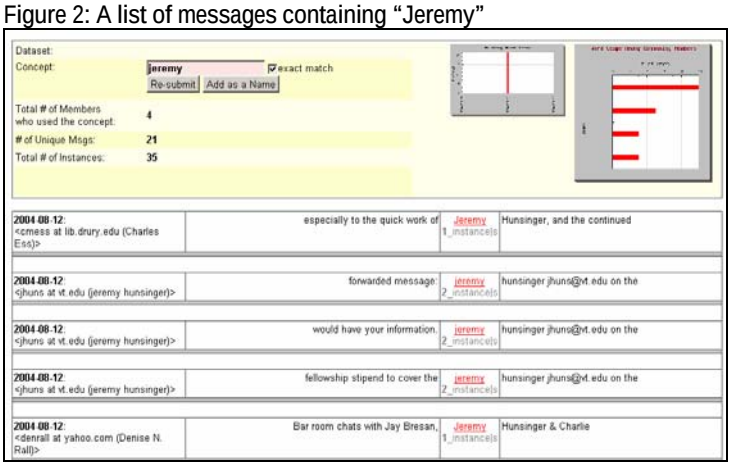


Figure 2: A list of messages containing "Jeremy"



Evaluation

To evaluate the accuracy and effectiveness of our automated approach (further referred to as *Local name extractor*), we compared it with another automated name extractor constructed based on Alias-I LingPipe[1], a state of the art toolkit for linguistic analysis. After selecting two non-intersecting sample subsets from our dataset: Subset A (**853** postings) and Subset B (**534** postings), we run both extractors and compare results. We decided to use evaluation measures traditionally used in the NER task: *P* precision and *R* recall. These measures are calculated in the following manner. Precision $P = \frac{T1}{T1 + F}$, defined as a ratio of all correctly identified names (T1) to all words labeled by the program as names (T1+F); where F is the number of false-positive results (words that were incorrectly labeled as names). And recall $R = \frac{T1}{T2}$, is defined as a ratio of all correctly identified names (T1) to all names in the dataset (T2). T2 is calculated by counting all distinct names found by both algorithms. In addition to measures mentioned above, we also evaluated the efficiency of both systems by recording and comparing their execution time. Table 5 shows the results of this evaluation.

Table 5: Comparing Local and LingPipe name extractors

	Subset A		Subset B	
	Local	LingPipe	Local	LingPipe
Total # names	1459	997	929	577
Total # correct distinct names discovered (T1)	331	340	195	176
False-Positive (incorrectly identified) (F)	45(12%)	227(40%)	39(16%)	238(57%)
# names in common	171		99	
# of missed names found by the alternate algorithm (D)	160	165	74	96
# of missed names of group members (M)	4	0	3	0
Total names in the dataset T2=T1+M+D	500	500	272	272
Precision P = T1/(T1+F)	0.88	0.60	0.83	0.43
Recall R = T1/T2	0.66	0.68	0.71	0.65
Execution time (in minutes)	3	52	2	34

[1]Alias-I LingPipe toolkit for linguistic analysis - http://www.alias-i.com/lingpipe

The results demonstrate that the *Local name extractor* returned far fewer false-positive results, than LingPipe: 12% and 16% versus 40% and 57% of the total number of extracted names in Subset A and B correspondently. In other words, a user would have to remove fewer incorrectly labeled words when using Local name extractor than with LingPipe. This fact is also supported by the higher values of precision *P* for Local name extractor: 1.46 times higher for Subset A and almost two times higher for Subset B. After the detailed examination of the results, we noticed that a larger number of mislabeled words by LingPipe are capitalized words such as names of software products (e.g., "Adobe Acrobat", "Dreamweaver"), words of exclamation and amazement such as "Aha", "Yeah", "Duh", "Wow", and greetings such as "Hi pg", "Hey", "Hello all". This is likely the result of LingPipe being trained originally on newswire corpora where words of exclamation and amazement as well as greeting are rare. For Local name extractor, the most common reasons for false-positive results was the selection of words from the name dictionary such as "major", "long", and "mark" that were not used as names in the text.

When examining recall values, both algorithms showed comparable results around 0.65 - 0.70. Recall indicates how many more words need to be added manually. Among the names that were missed by LingPipe were group names and nicknames (e.g., dw, ed) which are difficult to detect for any algorithms. But there were also missed names that should have been easy to find such as Wendy, Vincent, Scot, and Robert. As for Local name extractor, the most frequently missed names were solitary last names that were not precede or succeed by other contextual words. This can be explained by the fact that since group members in online communities usually refer to each other by their first names or nicknames, Local name extractor was not designed to recognize them. Our local name extractor also missed 4 and 3 names of group members in Subsets A and B correspondently due to the name foreign origins. These names were later added manually using the web interface. Despite this drawback, the execution time was 17 times faster than that of LingPipe. This fact coupled with the substantially higher precision makes the Local name extractor a very effective and efficient tool for the personal name extraction.

3.3.2. Automated Tie Discovery

After all network nodes, consisting of previously extracted personal names, are identified, the next step is to uncover if and how these nodes are interconnected. As mentioned previously, the posting headers by themselves are not very reliable source for '*who talks to whom*' data. Therefore, we will be relying on the content of messages to infer ties between people. There are two main methods in the literature for automated discovery of ties based on textual information. One is based on *semantic similarity* between so-called personal *profiles*. There are many different sources on how to measure *semantic similarity*, just to mention few: Kozima & Furugori (1993); Maguitman et al. (2005); Resnik (1999). Each personal *profile* describes a person's interests as a set of words or phrases. Words for somebody's profile are selected either manually by that person himself (e.g. Facebook profile) or pulled out automatically from a person's homepage/message or from parts of the text written about that person elsewhere. According to this method, two people are connected when the value of *semantic similarity* between their *profiles* is higher than a predefined threshold. In other word, people are considered to be connected when there is a substantial overlap of words found in their profiles. A variation of this method is often used for *expert* or *cooperator identification*. For example, Campbell et al. (2003) relied on both keywords submitted by users and the content of the users' emails to form what they call *expertise graph* which connects people based on their self-professed expertise. Another method is to use some sort of *co-occurrence metric* to calculate the number of times two names co-occur in close proximity in the text (e.g., Kautz et al., 1997). For example, Matsuo et al. (2006) relied on the number of co-occurrences of two people on web pages. Their solution used a simple, but elegant method to count co-occurrences by using a search engine. They counted the number of hits from an Internet search engine in response to a query consisting of two names joined via the Boolean operator AND. A major limitation of both methods is that they do not tell us much about the type of social relations, if any at all, between people. In other words, if two nodes are connected according to one of the methods mentioned above, it is not clear whether they are also connected socially, even though, they may have shared interests. Due to this limitation, we decided to develop our own tie discovery algorithm.

In what follows, we describe and evaluate our approach for automated discovery of ties from text. It is a two-step procedure. First, it derives *who talks to whom* from the content of postings. Then, it assigns tie strength across a pre-defined set of relations that believed to be predicting factors of the success in online communities. Some examples of these relations include *information sharing* (Mori et al., 2005), *trust* (Dutton & Shepherd, 2006; Matsuo et al., 2004), *information exchange* (Burnett & Buerkle, 2004; Burnett, 2000), *interactivity* (Rafaeli & Ariel, 2007; Rafaeli & Ravid, 1997; Rafaeli & Sudweeks, 1997), *diffusion of innovations* (Gregor & Jones, 1999; Rogers, 2003). For each *relation*, we will define how to measure it automatically based on the content of postings. In the current paper, we will only focus on *Information Exchange* that is especially important for e-learning communities.

Deriving "who talks to whom" data

For the purpose of our study, we are working with the assumption that the chance of two people sharing a social tie is proportional to the number of times each of them mentions the other in his/her postings either as an addressee or a subject person. As a way to quantify this assumption, our approach adds a nominal weight of 1 to a tie between a poster and all names found in the postings. After processing all postings, only those ties that have weights higher

than a pre-defined threshold (to be determined experimentally) are included as part of the *name network*. To demonstrate our approach, we will use a sample posting in Figure 3 below.

Figure 3: A sample posting, names are in bold

> From: wilma@bedrock.us
> Reference Chain: tank123@gl.edu, hle@gl.edu
>
> Hi **Dustin**, **Sam** and all, I appreciate your posts from this and last week […]. I keep thinking of poor **Charlie** who only wanted information on "dogs". […] Cheers, **Wilma**.

As indicated in the header, this posting is from *wilma@bedrock.us*, and it is a reply to the post by *tank123@gl.edu* (who actually started the thread) and *hle@gl.edu*. There are four names in the posting: *Dustin*, *Sam*, *Charlie*, and *Wilma*. According to our algorithm, there will be connections between the poster *wilma@bedrock.us* to each name in the postings:

wilma@bedrock.us - Dustin
wilma@bedrock.us – Sam
wilma@bedrock.us – Charlie
wilma@bedrock.us – Wilma

However, there are a few problems with this approach. First, *Wilma* is a poster; so there is no need for the *wilma@bedrock.us – Wilma* connection. Second, what will happen if more than one person has the same name? For example, suppose that there is more than one *Sam* in the group, how would we know which *Sam* is mentioned in this posting? Conversely, there could be situations where many different names can belong to one person. Furthermore, in the example above, *Charlie* is not even a group member; he is just an imaginary user. Ideally, we should not be connecting the poster with *Charlie*. To address these problems, a so-called *alias resolution* algorithm is often used. In the section below, we describe our approach to perform *alias resolution*.

Alias resolution via Name–Email associations

To disambiguate name aliases, we adopted a simple but effective approach that relies on the algorithm to associate names in the postings with email addresses in the corresponding posting headers (further referred as *name-email* associations). By learning *name-email* associations, our system will know that there are, for example, two *Nicks* because of the existence of two associations for *Nick* with two different email addresses. The easiest way to discover such *name-email* associations is to use a class list (also known as a *roster*) or University's online phonebook directory. However, for reasons mentioned previously, we decided not to rely on such resources. Instead, we developed a general algorithm that will learn all associations automatically. To simplify the task for the algorithm, we decided to ignore all names that clearly belong to non-group members. In our dataset, these are usually full names that appear in the middle one-third of the posting. Remaining names are considered to belong potentially to group members and are not excluded from the analysis.

In our algorithm, we relied on an assumption, similar to one used by Hölzer et al. (2005), that the higher number of collocations of two objects generally indicates a stronger association between them. In our case, the two objects are (1) a personal name from the body of the posting and (2) an email from the posting header. To improve the accuracy of associations, instead of counting collocations for all names and all emails, we associate a name with either a poster's email or with an email of potential addressee(s) (emails from the *reference chain*). As a point of clarification, we will refer to the association between a name and poster's email as *Association type P* (or just *Association P*), where *P* stands for *poster*. And the association between a name and addressee's email will be called *Association type A* (or just *Association A*), where *A* stands for *addressee*. In addition to counting the number of collocations, our four-step alias resolution algorithm is also based on assessing the confidence level for each association. The confidence level is assigned based on two criteria: a relative position of a name in the posting and a list of context words as described below.

Step 1. Determining Associations type P: A poster's name usually appears near the end of a posting in the signature. To find the Association P and estimate its confidence level, we first calculate how far a name is from the end of the posting using the following formula: $\frac{\text{pos}}{100}$, where *pos* is a relative position of a name inside the posting (in percent). This value is taken as an initial value of the confidence level. Next, we check if a name appears in close proximity (1-2 words) to one or more words or phrases that are commonly used in the signature such as 'thank you', 'best regards', 'cheers', or a sign of a new line. (We manually compiled a dictionary of these words/phrases before running the algorithm.) If yes, the value of the confidence level is doubled.

Step 2. Determining Associations type A: Next, we find and estimate the confidence level of Association A. To calculate the initial value of the association, we used the complement of the formula from Step 1: $1-\frac{pos}{100}$. This is because the closer the name is to the beginning of the posting, the more likely it is to be a part of the greeting. And if a word is part of the greeting, then it is more unlikely that the name belongs to a poster. In addition to measuring the position of the name in the posting, similar to what was done in Step 1, we compiled a dictionary of words that commonly appear with addressees. For example, words used in greetings include "hi", "hello", "dear", etc; in agreement - "agree with", "disagree"; and in references to others - "according to", "said that", etc. And again, if one of the words/phrases from this dictionary is found in a close proximity to the name, than the confidence level of Association A is doubled.

To demonstrate Step 1 and 2, we will once again use the sample posting as shown in Figure 3 above. The result of running Step 1 and 2 of our algorithm is shown in Table 6 below. The last two columns in Table 6 show the estimated confidence level for associations P and A.

Table 6: The results of running the algorithm

Words to the Left	Name	Words to the Right	Position	Context word	Association P	Association A
Hi	Dustin	Sam and	0	Yes	0	1 • 2 = 2
Hi Dustin	Sam	and all	0.01	Yes	0.01	0.99 • 2 = 1.98
Of poor	Charlie	who only	0.50	No	0.50	0.50
Cheers *	Wilma		0.88	Yes	0.88 • 2 • 2 = 3.52	0.12

* - indicates a new line

In the next step of our alias resolution, we will explain the procedure of selecting the strongest association for each name. For example, we need to decide whether *Sam* is a poster and thus, should be associated with the poster's email or is he a recipient of the posting and therefore, should be associated with an addressee's email.

Step 3. Choosing between Association P or A: Next, we compare and select an association with the highest confidence level. When the difference between values for P and A is insignificant (less than 10%), we reject both associations due to a lack of information to determine whether a name is a poster or an addressee in the message. If P is greater than A, then we assign that particular name to the poster's email found in the 'From' field of the header. Otherwise, the name is assigned to an addressee's email. In the example above, *Charlie* will be ignored due to the lack of evidence to support one association or the other. *Wilma* will be associated with the poster's email `wilma@bedrock.us`. Finally, *Dustin* and *Sam* will be considered to be addressees. However, as we noticed earlier, there is no information of addressees' emails in the posting headers. To work around this, we assumed that addressee(s) are likely to be somebody who posted in the thread previously; therefore, there is a good chance that their emails are likely to be in the *reference chain*. But because we do not know to which one, we associate the name with all emails in the *reference chain* using different weights. We distribute weights based on an email's position in the *reference chain*. From our observations, the earlier email appears in the *reference chain*, the less likely its owner is referenced in the current email; thus, it should gets the least weight. We found that a rectangular hyperbola function is a good candidate for weight assessment. In the current version of the algorithm, we used the following variation of the rectangular hyperbola function $w = 1 - \frac{1}{p+1}$, where ***p*** is the email's position in the *reference chain*. Following the formula above, when the value of ***p*** is increasing, indicating that we are moving from the first person in the chain to the most recent one, the weight ***w*** will be also increasing from 0.5 to close to 1. In the example above, Association A, between *Dustin* and `tank123@gl.edu` (thread starter), will get a weight of $1 - \frac{1}{1+1} = 0.5$ and between *Dustin* and `hle@gl.edu` will get a weight of $1 - \frac{1}{2+1} = 0.67$.

After processing all postings, the result is a list of *name-email* pairs and corresponding confidence levels. Because each message is unique, the confidence levels calculated based on different postings will be different even for the same *name-email* pair. To combine evidence from different postings, we calculated the overall value of the confidence level based on the confidence values of all occurrences of each unique *name-email* pair. Below is a formula that we have devised to accomplish this:

$$[\text{OVERALL CONFIDENCE LEVEL}]_{\text{for each unique } name\text{-}email \text{ pair}} = N_P \bullet M_P \bullet Par + N_A \bullet M_A$$

M_P, M_A are the median confidence level values for associations type P and A correspondently for each unique *name-email* pair. Note: The reason we are using *median* and not *average* function is to reduce the effect of possible outliers that may appear due to the variations in the posting formatting.

N_P, N_A represent the number of occurrences of each unique *name-email* pair for associations type P and A correspondently. Note: The reason we multiple medians M_P and M_A by N_P and N_A is to reflect the fact that the overall confidence level should grow proportionally to the number of the observed postings with that *name-email* pair.

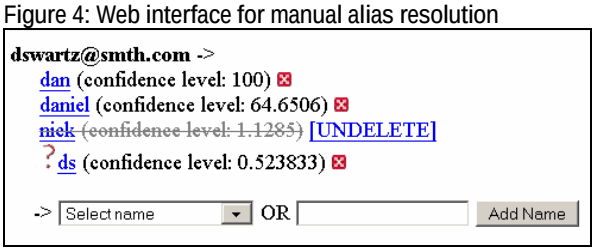
Par is an experimentally defined parameter (in the current version, it is set to *2*). *Par* is used to give more weight to the M_P-component of the formula. This is because there is less uncertainty in identifying Associations type P than Associations type A.

To exclude "weak" associations that might have appeared due to an error or those associations that do not have enough supportive evidence, we remove all associations where the value is less than *0.001* (defined experimentally). We also discovered that this is an effective way to remove all names of people who have never posted to the bulletin board.

Finally, to achieve a 100% level of accuracy on this task, we followed the same reasoning used when developing our Local name extractor and adopted a semi-automated approach. More specifically, we developed a web interface to allow a manual correction of the extracted associations (see Figure 4). For each email address that had at least one name associated with it, the system displays a list of choices for possible aliases sorted by their confidence levels. Using this interface, a researcher can easily remove and/or add a new *name-email* association by selecting a name from a list of all names found in the dataset from a drop down menu.

Using the formula described above, the end result of Step 3 is that for each distinct name found in the dataset, we will now have a list of email-candidates with their corresponding overall confidence level values. This list of email-candidates is then used during the final step of our alias resolution algorithm.

Figure 4: Web interface for manual alias resolution



Note: In the example above, the system assigned four names *Dan*, *Daniel*, *Nick*, and *DS* to dswartz@smth.com. However, after our manual examination of the results, we deleted *Nick* who was incorrectly associated with this email due to his frequent collocations in postings with *Dan*. Also a question mark next to *DS* indicates a small confidence level (less then 1).

Step 4. Disambiguating Personal Names: After learning all possible *name-email* associations and their overall confidence levels, the algorithm will go through all postings once again to replace those names mentioned in the body of the postings that have been associated with at least one email. If a name has more than one email-candidate, then the algorithm will use an email with the highest level of confidence. However, in some cases selecting an email with the highest level of confidence may produce an incorrect result. For example, in our dataset, there were two *Wilmas*: `wilma@bedrock.us` with the confidence level set to *27.45* and `wm2@iso.edu` with the confidence level set to *18.83*. If we were to select an email with the highest confidence level, then all mentions of *Wilma* in all postings would be attributed to only `wilma@bedrock.us`. But, of course, this would be wrong since in some instances it might be `wm2@iso.edu`. To ensure that we identify the right *Wilma*, we implemented the following fail-safe measure. If there are more than one email-candidate, we then rely on an additional source of evidence - *reference chain*. First, we identify an overlap between email-candidates for a name (from Step 3) and emails from the *reference chain*. If the overlap is empty, then we proceed as usual and use an email with the highest confidence level (further referred to as the *strongest candidate*). When the overlap is not empty, it means that one or more email-candidates have previously posted to the thread. Our assumption (based on the manual analysis of the dataset) is that the name mentioned in the posting is more likely to belong to an email-candidate that is also in the *reference chain* than to an email-candidate that is not. Following this assumption, if there are two possible email-candidates, as in case with *Wilma*, and the *strongest candidate* (`wilma@bedrock.us`) is not present in the *reference chain*, but the other candidate (`wm2@iso.edu`) is, then we use the one that is also in the *reference chain*. In cases, when both email-candidates have previously posted to the thread, we will take the candidate who has posted the most

recent posting to the thread. Although this approach does not guarantee 100% of accuracy, it performed much better in tests than a simple "the strongest wins" approach.

In the future, we are planning to improve the performance of our alias resolution algorithm by implementing other techniques found in the literature on authorship, citation analysis, spam detection, author disambiguation in digital libraries and other. In general, these approaches try to learn a unique 'signature' that can be associated with each person. They often rely on either unique linguistic characteristics of a person's writing (e.g. common writing styles, punctuation marks, an average length of sentences, expertise keywords, etc) (e.g., Bollegala et al., 2006; Fleischman & Hovy, 2004; Hsiung, 2004; Hsiung et al., 2005; Mann & Yarowsky, 2003; Pedersen et al., 2006) or network-based patterns of interactions (e.g., common senders and recipients) (e.g., Hölzer et al., 2005; Malin et al., 2005; Phan et al., 2006). There is also a number of simple but effective methods that rely on matching variations of names and/or email addresses to be considered (e.g., Bird et al., 2006; Feitelson, 2004; Patman & Thompson, 2003).

3.3.3. Estimating Tie Strengths via Information Exchange

After identifying ties based on the method described above, the next step is to estimate tie strengths. According to Mori et al. (2005), "[t]ie strength itself is a complex construct of several characteristics of social relations" and that "[n]o consensus for defining and measuring them exists" (p.83). Despite this observation, a simple approach to measure tie strength is to count the number of messages exchanged between individuals. However, as mentioned previously, this approach would take into account all messages including those that are not necessarily indicators of the kind of tie that is to be examined. Ideally tie strength should only reflect the type of relations that a researcher is trying to study. Since different messages may expose different types of *relations*, a researcher needs to be able to single out which messages are to be included in the analysis. If all messages are counted, later on it may be difficult to make assertions about the quality of tie strengths or to interpret their values. For example, if a *relation* being studied is *agreement*, then a researcher might want to decide to ignore all messages that are neutral in nature and keep only those that suggest *agreement* or *disagreement*.

In the current paper, we will single out one particular *relation* that is especially important in e-learning and assessment - *Information Exchange* (IE). IE is considered by many researchers to be an essential social interaction in a networked organization (Haythornthwaite, 1996; Haythornthwaite & Wellman, 1998), a key factor in the operation and success of virtual communities (Burnett, 2000), and specifically learning communities (Spada et al., 2005). Under Communication Theory, IE is seen as a mechanism to reach a mutual understanding (Rogers, 2003) and reduce uncertainty in Kincaid's Convergence model (Kincaid, 1979) and Uncertainty Reduction theory (Berger, 1975). Furthermore, Rafaeli has attempted to draw connection between IE and his emerging theory of interaction (Rafaeli & Ariel, 2007; Rafaeli & Ravid, 1997; Rafaeli & Sudweeks, 1997).

Despite a general agreement on the importance of this concept, there is little consensus on how to measure IE in electronic communication automatically. Researchers usually take one or two common routes. One is based on the patterns of communication (network-based) and the other is based on the content of communication (content-based). The network-based methods focus on the *information exchange* routes and usually rely on measures borrowed from Social Network Analysis (SNA) such as *centrality*, *cohesion* and *density* of social networks (e.g., Nurmela et al., 1999; Reyes & Tchounikine, 2005; Stephenson & Zelen, 1989). The content-based measures are grounded in communication and linguistic theories and often rely on methods from *content* and *discourse analysis* (e.g., Hsieh & Shannon, 2005; Wilson, 1993). In our work we will be relying on the content-based approach. This is because the alternative network-based approach is not applicable to our work; it requires an existing network to operate. In our case, the network is not built yet. And actually, our intention is to use IE data itself to build a network.

One of the first major decisions that we had to make was to decide between the two approaches to content-based measure: *binary* or *weighted*. The *binary* approach determines whether a message demonstrates a social interaction that can be categorized as IE or not. An example of a work that uses a *binary* definition of IE is Burnett's topology of IE (Burnett & Buerkle, 2004; Burnett, 2000). In his research, the author identified a list of top-level information-oriented behaviors in online environments that can be considered as IE such as *announcements* and various *information requests*. However, more detailed, lower-level categories would need to be devised for the topology to be used in an automated method such as ours. After some consideration, the *weighted* method was our preferred method because it can make use of all messages. With the weighted method, ties with more informative messages will receive higher weights than those with less informative ones. To measure the amount of information transferred through each posting, we simply count the number of descriptive concepts in each posting. (The discussion about what words are considered descriptive follows below.) Our supposition is that the higher number of descriptive concepts exchanged between two people will generally signify a higher level of IE and, thus, a stronger tie. We also believe that by using important concepts in their messages, participants, particularly in a learning setting, demonstrate their understanding of concepts that are being discussed. Since longer messages are likely to contain a higher number of descriptive concepts, we divide the total number of descriptive concepts by the number of all concepts in the message before including it into the final results.

In our work, we consider nouns and noun phrases as candidates for descriptive concepts. This is because nouns and noun phrases are believed by many researchers to be the most informative elements of the text (Carley, 1997; Corman et al., 2002; Boguraev & Kennedy, 1999; Haythornthwaite & Gruzd, 2007). To extract them from text, we used a free web service from Yahoo! called the *Term Extractor*[2]. Among the main reasons we chose to use this service is that it is fast and works well on texts across many different domains and genres. Unfortunately, due to its proprietary nature, we did not have a lot of information about the extractor's inner workings (although, some techniques behind the Yahoo! Term Extractor can be found in Kraft et al., 2005). Overall, we were satisfied with Term Extractor's performance. However, in some cases, it would miss some descriptive concepts and/or include some concepts that are not very descriptive. In our future work, we would like to have some more control over the concept extraction process and be able to compare various methods. In the near future we are planning to implement and evaluate some popular linguistic and statistical methods for terminology extraction ourselves.

The Yahoo! Term Extractor works as a black box. First a user submits a piece of text, and then the extractor returns a list of important concepts. For example, a sample message below will return three concepts: "*google*", "*search technology*" and "*library*". Thus, the amount of information it transmits can be estimated as $\frac{3}{49} = 0.06$; where 49 is a total number of words in the message.

> Keep in mind that google and other search technology are still evolving and getting better. I certainly don't believe that they will be as effective as a library in 2-5 years, but if they improve significantly, it will continue to be difficult for the public to perceive the difference.

By using this approach, we were able to detect and ignore non-informative messages[3] that do not contain important/descriptive concepts. Below is an example of how our IE-based weighting procedure influenced tie strengths in an ego network for a student A. Due to the absence of important/descriptive concepts in communication between A and F, a link between them was deleted.

From	To	Original weight	With IE
A	B	1	0.5
A	C	2	1.6
A	D	2	2.1
A	E	3	2.5
A	F	1	0

4. Preliminary Evaluation

As a way to evaluate our *name network* method and compare it against the *chain network* method, we used two sample datasets. Sample A consists of 853 postings in 12 bulletin boards, and Sample B consists of 534 postings in 5 bulletin boards. In our evaluation, we focused on questions regarding (1) different options for building chain networks, (2) the impact of the *Information Exchanged*-based weighting procedure, and (3) the accuracy of each of the two types of networks..

Question 1. The first question relates to whether there is any significant difference between the different options for building *chain networks*? The four options that were included in our analysis are:

- **Option 1.** Connecting a poster to the last person in the post chain only
- **Option 2**. Connecting a poster to the last and first (=thread starter) person in the chain, and assigning equal weight values of 1 to both ties
- **Option 3.** Same as option 2, but the tie between a poster and the first person is assigned only half the weight (0.5)
- **Option 4.** Connecting a poster to all people in the reference chain with decreasing weights

To compare these options, we used a QAP correlation. If two options produce similar networks, then their QAP correlation will be approaching 1. First, using different options, we built four different *chain networks*. We will further refer to these networks as R1, R2, R3, and R4 (the numbers for each network correspond to the four different options described above). Next, we calculated pairwise QAP correlations between these four networks. The results are in Table 7 below. As we expected, networks R2 and R3 turned out to be very similar (r=0.98). Since either one will serve our purpose, we do not need to use both R2 and R3 in future evaluations. The correlations also confirmed

[2]Yahoo! Term Extractor http://developer.yahoo.com/search/content/V1/termExtraction.html

[3]This applies to the IE relation. For other relations, 'non-informative' IE messages may indeed be informative. For example, a message that only says "I agree" may signal important social hierarchies of agreement.

the fact that option R4 was the most different option from all of the others. Finally, although networks R1 and R2 have a moderate correlation, they are different enough to be considered as two separate options in future evaluations. As a result of this exercise, we were able to reduce the number of chain networks that we would want to use in evaluations from 4 to 3 and to ensure that the remaining three options do produce different types of networks.

Table 7: QAP Correlations between Chain Networks (R) with different options for connecting a poster with people in the reference chain

	Sample A				Sample B			
	R1	R2	R3	R4	R1	R2	R3	R4
R1	1	0.358	0.377	-0.033	1	0.289	0.302	0.121
R2		1	0.978	-0.086		1	0.980	0.122
R3			1	-0.082			1	0.121
R4				1				1

Question 2. Next, we wanted to find out if there is a difference between networks built with and without our *Information Exchange* (IE)-based weighting measure. The resulting QAP correlations (see Table 8) show that (1) the correlation between networks without IE-based weighting and networks with IE is stronger for the *name network*, than for the *chain network*, and that (2) the *name network* has consistently higher values for both samples. These two facts together suggest that the use of IE-based weights do not change the structure of the *name network* as much as it does in the *chain network*. This is likely due to the mechanism of building *name networks*. Our hunch is that the use of only postings with direct references to others in a group may already filter out postings with low information even without the use of IE-based weights. This requires further testing.

Another observation is that when we used the IE-based weighting procedure on the R1, R2 and R3 networks in Sample B, it appears to have the strongest impact on their structures as demonstrated by consistently yielding lower correlation values. What is interesting here is that the same kinds of networks R1, R2, R3 in Sample A have yielded much higher correlation values, than those in Sample B. Among possible factors attributed to this difference may be the size of the samples and/or the nature of the selected bulletin boards/postings. A more detailed analysis of resulting networks is required to investigate these possibilities.

Table 8: QAP Correlations between networks with and without taking into account Information Exchange for Chain Networks (R) and Name Networks (N).

	Sample A	Sample B
R1	0.734	0.040
R2	0.778	0.289
R3	0.753	0.263
R4	0.837	0.857
N	0.941	0.940

Question 3. Finally, we wanted to evaluate the accuracy of the *name network* versus *chain network*. Again, we started by measuring pairwise QAP correlations. Some pairs of *name* & *chain networks* demonstrated weak correlations between 0.097 and 0.196 suggesting some overlap between the two types of networks. Although there is an overlap, there are also substantial differences in what is revealed by these two network derivations. To better understand these differences, we compared all connections that make up each tie from the *name network* with those of the *chain networks*. Of the 853 postings in sample A, only 319 postings contained explicit references to other people in the group. In analyzing these 319 postings, our program found 373 explicitly identified names of addressees that were then used to build the *name network*. We discovered that 102 (27.34%) of all addressees were not in the *reference chain*. This means that for these 319 messages, regardless of the method used for building it, a *chain network* misses about 27% of the potentially important connections and about the same amount will be incorrectly identified. Furthermore, if we used option 1 for building *chain networks*, the resulting network would incorrectly identify 157 (42.09%) connections for these 319 messages. Similar results are found for Sample B; 108 (38.57%) of potentially important connections in 219 messages were not found in the *reference chains*. And 152 (54.28%) of all addressees were not the most recent posters in the *reference chain* in these 219 messages. These preliminary results demonstrate that *name networks* address some of the shortcoming of *chain networks*. But more research is needed to test the generalizability of the *name network* method with regards to datasets from other domains or genres. In future research, we will also need to address some of the possible limitations of *name networks*. For example, there is a risk of missing some of the connections that exist within messages that do not have explicit references to a specific name.

5. Summary and Future Research

The *chain* and *name network* approaches described here provide one more step towards the task of understanding and extracting social interaction networks from online discussion board data. With the help of egocentric analysis, our future plans call for a more detailed comparison of the chain and name networks to each other, and to networks generated from participant judgments of their interactions.

When fully developed and tested, we believe that *name networks* will be a useful diagnostic tool for instructors to evaluate and improve lesson plans, and to identify students who might need additional help or students who may provide such help to others. This is possible because of two important features associated with the *name network* methodology. First, *name networks* take into account only those messages that contain personal references to others in a group. These messages tend to be more interactive and argumentative. The present of both of these elements makes these messages good indicators of collaborative learning. Second, by operationalizing and measuring information exchange, our mechanism of assigning tie strengths allows us to increase weights for messages that we believe to be better contributors to shared knowledge construction.

References

Allen, I.E. and Seaman, J. (2007). Online Nation: Five Years of Growth in Online Learning. Needham, MA, Sloan Consortium.

Berger, C.R. (1975). Some Explorations in Initial Interaction and Beyond: Toward a Developmental Theory of Interpersonal Communication. *Human Communication Research* 1(2): 99.

Bird, C., Gourley, A., Devanbu, P., Gertz, M. and Swaminathan, A. (2006). Mining Email Social Networks. *In the Proceedings of the 2006 international workshop on Mining software repositories*, Shanghai, China, ACM Press.

Boguraev, B. and Kennedy, C. (1999). Applications of Term Identification Technology: Domain Description and Content Characterisation. *Natural Language Engineering* 5(01): 17-44.

Bollegala, D., Matsuo, Y. and Ishizuka, M. (2006). Extracting Key Phrases to Disambiguate Personal Names on the Web. *Computational Linguistics and Intelligent Text Processing***:** 223-234.

Bruffee, K.A. (1993). Collaborative Learning: Higher Education, Interdependence, and the Authority of Knowledge. Baltimore, Johns Hopkins University Press.

Burnett, G. (2000). Information Exchange in Virtual Communities: A Typology. *Information Research* 5(4).

Burnett, G. and Buerkle, H. (2004). Information Exchange in Virtual Communities: A Comparative Study. *Journal of Computer-Mediated Communication (JCMC)* 9(2).

Campbell, C.S., Maglio, P.P., Cozzi, A. and Dom, B. (2003). Expertise Identification Using Email Communications. *In the Proceedings of the twelfth international conference on Information and knowledge management*, New Orleans, LA, USA, ACM.

Carley, K.M. (1997). Extracting Team Mental Models through Textual Analysis. *Journal of Organizational Behavior* 18(S1): 533-558.

Chen, Z., Wenyin, L. and Zhang, F. (2002). A New Statistical Approach to Personal Name Extraction. *In the Proceedings of the Nineteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc.

Chinchor, N. (1997). Muc-7 Named Entity Task Definition. *Proceedings of the 7th Message Understanding Conference.*

Chklovski, T. and Pantel, P. (2004). Verbocean: Mining the Web for Fine-Grained Semantic Verb Relations. *In the Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain.

Cook, S.D.N. and Brown, J.S. (1999). Bridging Epistemologies: The Generative Dance between Organizational Knowledge and Organizational Knowing. *Organization Science* 10(4): 381-400.

Corman, S.R., Kuhn, T., McPhee, R.D. and Dooley, K.J. (2002). Studying Complex Discursive Systems: Centering Resonance Analysis of Communication. *Human Communication Research* 28(2): 157-206.

Culotta, A., Bekkerman, R. and McCallum, A. (2004). Extracting Social Networks and Contact Information from Email and the Web. *In the Proceedings* of *CEAS*, Mountain View, CA.

Dutton, W.H. and Shepherd, A. (2006). Trust in the Internet as an Experience Technology. *Information, Communication & Society* 9(4): 433-451.

Feitelson, D.G. (2004). On Identifying Name Equivalences in Digital Libraries. *Information Research* 8(4): 192

Fleischman, M.B. and Hovy, E. (2004). Multi-Document Person Name Resolution. *Proceedings of ACL-42, Reference Resolution Workshop*.

Gregor, S. and Jones, K. (1999). Beef Producers Online: Diffusion Theory Applied. *Information Technology & People* 12(1): 71.

Harada, M., Sato, S. and Kazama, K. (2004). Finding Authoritative People from the Web. *In the Proceedings of Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference*.

Haythornthwaite, C. (1996). Social Network Analysis: An Approach and Technique for the Study of Information Exchange. *Library & Information Science Research* 18(4): 323-342.

Haythornthwaite, C. and Gruzd, A.A. (2007). A Noun Phrase Analysis Tool for Mining Online Community. *In the Proceedings of the 3rd International Conference on Communities and Technologies*, Michigan State University.

Haythornthwaite, C. and Wellman, B. (1998). Work, Friendship, and Media Use for Information Exchange in a Networked Organization. *Journal of the American Society for Information Science* 49(12): 1101-1114.

Hölzer, R., Malin, B. and Sweeney, L. (2005). Email Alias Detection Using Social Network Analysis. *In the Proceedings of the 3rd international workshop on Link discovery*, Chicago, Illinois, ACM Press.

Hsieh, H.-F. and Shannon, S.E. (2005). Three Approaches to Qualitative Content Analysis. *Qual Health Res* 15(9): 1277-1288.

Hsiung, P. (2004). Alias Detection in Link Data sets. *Robotics Institute*. Pittsburgh, PA, Carnegie Mellon University. Master.

Hsiung, P., Moore, A., Neill, D. and Schneider, J. (2005). Alias Detection in Link Data sets. *In the Proceedings of the International Conference on Intelligence Analysis*.

Jenkins, H. (2006). Confronting the Challenges of Participatory Culture. Chicago, IL, MacArthur Foundation.

Kautz, H., Selman, B. and Shah, M. (1997). Referral Web: Combining Social Networks and Collaborative Filtering. *Commun. ACM* 40(3): 63-65.

Kincaid, D.L. (1979). The Convergence Model of Communication (Paper No. 18). *Honolulu, HI: East West Center, Communication Institute*.

Klavans, J. and Kan, M.Y. (1998). Role of Verbs in Document Analysis. *In the Proceedings of the 17th International Conference on Computational Linguistics*.

Koschmann, T., Ed. (1996). *CSCL: Theory and Practice of an Emerging Paradigm*. Mahwah, NJ, Erlbaum.

Kozima, H. and Furugori, T. (1993). Similarity between Words Computed by Spreading Activation on an English Dictionary. *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics, April*: 21-23.

Kraft, R., Maghoul, F. and Chang, C.C. (2005). Y!Q: Contextual Search at the Point of Inspiration. *In the Proceedings of the 14th ACM international conference on Information and knowledge management*, Bremen, Germany, ACM.

Lave, J. and Wenger, E. (1991). Situated Learning: Legitimate Peripheral Participation. Cambridge, UK, Cambridge University Press.

Maguitman, A.G., Menczer, F., Roinestad, H. and Vespignani, A. (2005). Algorithmic Detection of Semantic Similarity. *In the Proceedings of 14th international conference on World Wide Web*.

Malin, B., Airoldi, E. and Carley, K.M. (2005). A Network Analysis Model for Disambiguation of Names in Lists. *Comput. Math. Organ. Theory* 11(2): 119-139.

Mann, G. and Yarowsky, D. (2003). Unsupervised Personal Name Disambiguation. *In the Proceedings of CoNLL*, Edmonton, Alberta, Canada.

Matsuo, Y., Hamasaki, M., Nakamura, Y., Nishimura, T., Hasida, K., Takeda, H., Mori, J., Bollegala, D. and Ishizuka, M. (2006). Spinning Multiple Social Networks for Semantic Web. *In the Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence*, Boston, Massachusetts, The AAAI Press, Menlo Park, California.

Matsuo, Y., Tomobe, H., Hasida, K. and Ishizuka, M. (2004). Finding Social Network for Trust Calculation. *In the Proceedings of the 16th European Conf. on Artificial Intelligence (ECAI2004)*.

Medlock, B. (2006). An Introduction to NLP-Based Textual Anonymisation. *In the Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Minkov, E., Wang, R.C. and Cohen, W.W. (2005). Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text. *In the Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada, Association for Computational Linguistics.

Miyake, N. (2007). Computer Supported Collaborative Learning. *Handbook of E-Learning Research*. R. A. C. Haythornthwaite. London, Sage**:** 263-280.

Mori, J., Sugiyama, T. and Matsuo, Y. (2005). Real-World Oriented Information Sharing Using Social Networks. *In the Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, Sanibel Island, Florida, USA, ACM.

Nurmela, K., Lehtinen, E. and Palonen, T. (1999). Evaluating CSCL Log Files by Social Network Analysis. *In the Proceedings of the 1999 conference on Computer support for collaborative learning*, Palo Alto, California, International Society of the Learning Sciences.

Patman, F. and Thompson, P. (2003). Names: A New Frontier in Text Mining. *Intelligence and Security Informatics: First Nsf/Nij Symposium, Isi 2003, Tucson, Az, USA, June 2-3, 2003. Proceedings***:** 960-960.

Pedersen, T., Kulkarni, A., Angheluta, R., Kozareva, Z. and Solorio, T. (2006). An Unsupervised Language Independent Method of Name Discrimination Using Second Order Co-Occurrence Features. *Computational Linguistics and Intelligent Text Processing***:** 208-222.

Phan, X.-H., Nguyen, L.-M. and Horiguchi, S. (2006). Personal Name Resolution Crossover Documents by a Semantics-Based Approach. *IEICE Trans Inf Syst* E89-D(2): 825-836.

Rafaeli, S. and Ariel, Y. (2007). Assessing Interactivity in Computer-Mediated Research. *The Oxford Handbook of Internet Psychology*. A. N. Joinson, K. Y. A. McKenna, T. Postmes and U. D. Rieps, Oxford University Press**:** 71-88.

Rafaeli, S. and Ravid, G. (1997). Online, Web-Based Learning Environment for an Information System Course: Access Logs, Linearity and Performance. *In the Proceedings of ISECON '97 (pp. 92-99)*.

Rafaeli, S. and Sudweeks, F. (1997). Networked Interactivity. *Journal of Computer Mediated Communication* 2(4).

Resnik, P. (1999). Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial IntelligenceResearch* 11: 95-130.

Reyes, P. and Tchounikine, P. (2005). Mining Learning Groups' Activities in Forum-Type Tools. *In the Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!*, Taipei, Taiwan, International Society of the Learning Sciences.

Rogers, E.M. (2003). Diffusion of Innovations, Simon and Schuster.

Spada, H., Meier, A., Rummel, N. and Hauser, S. (2005). A New Method to Assess the Quality of Collaborative Process in CSCL. *In the Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!*, Taipei, Taiwan, International Society of the Learning Sciences.

Stephenson, K. and Zelen, M. (1989). Rethinking Centrality: Methods and Examples. *Social Networks* 11(1): 1-37.

Sweeney, L. (2002). K-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems* 10 (5): 557-570.

Sweeney, L. (2004). Finding Lists of People on the Web. *ACM Computers and Society* 34(1).

Uzuner, O., Luo, Y. and Szolovits, P. (2007). Evaluating the State-of-the-Art in Automatic De-Identification. *J Am Med Inform Assoc* 14(5): 550-563.

Wilson, A. (1993). Towards an Integration of Content Analysis and Discourse Analysis: The Automatic Linkage of Key Relations in Text. Lancaster, UCREL.