Comparison of explicit and implicit social networks constructed from communication data

Jana Diesner
University of Illinois
Urbana Champaign
The iSchool and Department of Computer Science
jdiesner@illinois.edu

Amirhossein Aleyasen
University of Illinois
Urbana Champaign
Department of Computer Science
aleyase2@illinois.edu

Shubhanshu Mishra
University of Illinois
Urbana Champaign
The iSchool
smishra8@illinois.edu

Aaron Schecter
Northwestern University
Science of Networks in Communities (SONIC)
aaronschecter2016@u.northwestern.edu

Noshir Contractor
Northwestern University
Science of Networks in Communities (SONIC)
nosh@northwestern.edu

ABSTRACT

Social networks can be constructed from explicit information about who is talking to whom, and/ or inferred from the content of communication. How do the resulting network structures compare? We provided an answer to this question by constructing explicit social networks from chat logs and comparing them to implicit social networks built from text data generated by these agents. We apply different conceptualizations of similarity to the text data. This work helps to understand if explicit social networks (what people typically work with) can serve as a proxy for the true structure of communication networks.

Our findings suggest that the more simplistic approach on the lexical level outperforms the more complex, topic based approach. This means that reconstructing social networks based on lexical features is the best option tested, while detecting alternative and additional latent structures of people sharing the same topical knowledge requires looking for thematic clusters of word use.

Author Keywords

Communication networks, socio-semantic networks, text mining, validation

ACM Classification Keywords

E.1., G.2.3, H.3.1, I.2.7

General Terms

Human Factors; Design; Measurement.

1. INTRODUCTION

Communication networks are meant to represent network participants and the information flow between them. Typically, communication networks are constructed by observing or inquiring information about who is talking to whom. In a more general sense, these networks represent social networks where people interact with each other through a specific type of behavior, i.e. natural language. Building such networks by denoting social agents as nodes and the information flow between them as edges is straightforward, acknowledges the exchange of information, and can be highly efficient, e.g. when network data are automatically extracted from chat logs, email conversations or social media data. However, this approach also reduces the content of communication to the fact, frequency or likelihood of the flow of information between nodes. This can be problematic since prior research has shown that without considering the substance of communication data, our ability to model and understand the effects of language use in networks becomes limited. This includes the transformative role that language use can play in networks as well as the interplay and co-evolution of communication and networks [1-5].

To address this limitation, a variety of methods for building social network data from information explicitly or implicitly contained in unstructured, natural language text data has been developed [for an overview see 6]. These methods are explained in more details in the background section. We herein refer to network data collected by observing or asking network participants about their ties as *explicit social networks*, and networks inferred from the content of text data as *implicit social networks*. The body of prior work on these networks leaves three critical questions unanswered:

1. How do communication networks extracted from information contained in text data (implicit social networks) compare to networks constructed by collecting data from the networks participants directly, e.g. by questionnaires or observations (explicit social networks)?
2. Given a variety of available methods for extracting implicit social networks from text data, which method(s) best resemble(s) explicit social networks?
3. Are there any best practices for combining text-based methods (with or without explicit network data construction methods) for gaining a more comprehensive view of a network?

WebSci'14, Jun 2014, Bloomington, IN.
Copyright 2014 ACM 978-1-4503-1889-1....$10.00.

In this paper, we address questions one and two. The outcome of this work patches some holes at the intersection of network analysis and text mining, and lays the foundations for answering questions three. Why does this work matter? First, without knowing how closely implicit social networks resemble explicit social networks, we cannot assume that explicit social networks are a good proxy for the true structure of communication networks. Second, if explicit and implicit social network data align, then one type can serve as a substitute for the other type, e.g. in cases where data collection for a certain type is hard to infeasible. In order to address the given research questions, we have implemented and modified a variety of techniques for extracting social networks from text data (methods section), applied them to a corpus of empirical chat log data from which we also construct explicit social network structure (data section), and compared the results against each other (results section).

2. BACKGROUND

One general approach for constructing implicit social network data from text data is to conduct entity detection paired with relation extraction [7]. This means to identify all instances of references to social agents in the text data, and linking them based on criteria such as distance (the most common approach) [2], shallow and deep syntax [1], and statistical features [7-9]. This approach is reasonable when social agents as well as indicators for the relations between them are referred to in the text data. Overall, the underlying goal here is to correctly extract social structure from text data [10]. This is particularly useful when no other sources for network data might be available, e.g. in the cases of historic and covert networks.

Alternatively, authors of pieces of text data can be considered as agent nodes. This could be the authors of documents, posts, tweets, etc.. Information on social agents can also be entailed in log files that record discussions between people [11]. In either case, these nodes then get linked based on a certain amount of similarity between the agent's language use, sentiment etc. In the simplest case, this is realized by extracting salient terms or (one step up) themes that emerge from the text data, representing these (sequences of) tokens as vectors, and computing the congruence between these vectors by choosing from a variety of similarity metrics [12, 13]. Overall, this approach is useful when network participants have provided some content, which can then serve to construct social network data or supplement explicit social network data.

In this paper, we focus on the second approach. Why? First, because it represents the more general case, and relation extraction might still be conducted in addition to the similarity-based node linkage. Second, this approach eliminates uncertainties or error rates for the entity detection part such that we can focus on the core of our research questions without diluting the results with additional intervening variables. Accuracy of entity detection currently ranges, depending on the type of entity, between the upper 80ies to 90% percent; with this intervening factor being removed from our experimental design for this study.

3. DATA

The data for this project were captured from a computer-based simulation game designed specifically to identify the defining aspects of multiple teams working interdependently toward hierarchically arranged goals. The goal of this multi-team system simulation is to guide a convoy of humanitarian aid through enemy territory. To accomplish this goal, individuals must collect intelligence, neutralize threats, and move the convoy to reach as many destinations in their region as possible. There are four component teams – Atlantica, Baltica, Caspia, and Pacifica – in each simulation session. Each component team consisted of five individuals: a leader, a reconnaissance officer and a field specialist who work on counter-insurgency, and a reconnaissance officer and a field specialist who work on ordinance disposal. Roles were appointed randomly. The leaders were charged with moving the convoy. The leaders had to agree on where and when to advance the convoy. The four non-leader team members were responsible for identifying and neutralizing threats. Each team had a counter-insurgency and an ordinance disposal unit; each comprised of a reconnaissance officer and a field specialist. The reconnaissance officer is responsible for identifying potential threats and must communicate this information to the field specialist, who will then act on engaging and eliminating the threat.

All individuals were given their own laptop and headset. Communication was carried out primarily via Skype, both through chat and voice. For each simulation session, a full transcript of time-stamped messages was logged. For this paper, we are considering the chat logs of each simulation session. In addition to virtual communication channels, the players were grouped into six physical workstations, each of which had two to four individuals. Three workstations allowed face to face communication; in the other three, the players were forced to use only their headsets. Additionally, the rooms were shuffled so that players were not necessarily collocated with their own team. The room assignments and communication rules were constant across all sessions.

In total, thirty-three experimental sessions were carried out. Overall, there were 660 unique individuals in the experiment. However, the twenty game-specific roles were held constant across each run; in other words, "leader.caspia" was played by thirty-three people, but sat in the same seat in the same room on the same laptop, and was responsible for the same quadrant of the game map.

4. METHOD

From the abovementioned chat log data, we have constructed one explicit social network for each of the 33 simulation games. In order to compare these networks to implicit social networks that are built based on shared knowledge entailed in the content of chat logs, we apply different techniques as outlined below. We have implemented these techniques as described in this section and made them available as routines in ConText (http://context.lis.illinois.edu/), a publicly available toolkit.

Usually, communication logs consist of a messages set. Elements of each message can have attributes such as sender, message body, timestamp, priority, etc. We formalize the set of possible types of communication representable by chatlogs as person-to-person networks and broadcasting networks. In person-to-person networks, each message has sender and receiver attributes. In broadcasting networks, each message only has a sender; all nodes in the network are receivers in this case. The assumed data structure for communication data or chat logs that we are processing are csv files that contain at a minimum:

- A column that specifies senders
- A column that specifies receivers
- Communication content

We generate social networks from the textual evidence of communication activity as well as from text content by detecting concepts and themes that are shared between people. For the first approach, we basically parse the senders (and receivers) from the log file. For the second approach, we use token based and topic based text mining methods, which we describe next.

4.1 Token Similarity Based Networks

A communication network consists of agent sets $A = \{a_1, \cdots, a_n\}$. For simplicity, we consider each message as a pair (b, s) where b is message body and s is message sender. The messages set is $M = \{(b_1, s_1), (b_2, s_2), \cdots, (b_m, s_m)\}$ where $b_i \in A$ and $s_i$are arbitrary strings. We define the similarity of two agents using their messages similarity in M,

$$d(i, j) = \frac{\sum_{(b_i, s_k), (b_j, s_l) \in M} sim(s_k, s_l)}{|M_{a_i}| + |M_{a_j}|}$$

Where $|M_{a_i}|$ is the number of message in M that a_i sent.

For messages similarity, we use different conceptualizations of the string similarity of any pair of messages. In general, there are two groups of string similarity methods: *edit-distance like functions* and *token-based distance functions* [12]. In edit-distance like functions, the distance $d(s_i, s_j)$ is calculated as the costs of the operations needed to convert s_i to s_j. Typical edit operations include character insertion, deletion and substitution. Each operation has predefined costs. Levenstein, Jaro and Jaro-Winkler are three most common edit-distance like methods.

In token-based distance functions, $s_i$and s_j are considered as multisets of tokens (we define words as space separated tokens). Jaccard similarity, cosine similarity and Jensen-Shannon are the most common token-based functions.

We herein use the following similarity methods: (1) Jaccard similarity, and (2) SoftTFIDF Jaro-Winkler. The Jaro–Winkler distance metric is designed and best suited for short strings such as the names of people, organizations and locations. The score is normalized such that 0 equates to no similarity and 1 indicates an exact match. We provide the "soft" version of TFIDF in Jaro-Winkler, in which similar tokens are considered as well as exact match tokens in Jaro-Winkler. It has been empirically shown that the best-performing method for string distance metrics in terms of accuracy and speed is SoftTFIDF Jaro-Winkler [12].

4.2 Topic Based Networks

In order to identify agents who are connected based on similar topics we conduct topic modelling [14]. Topic modeling is an unsupervised summarization technique that represents the main themes occurring in a body of text data in terms of topics, where topics are unlabeled ordered sets of text-based tokens that most strongly represent that topic. We have adapted LDA based topic modelling as provided in Mallet [15], which we have integrated into ConText. The input data for our version of topic modeling are the messages sent by each user. Given *n* users in a chat log file, we construct *n* documents, one per user. Each document contains all the messages from that one user.

In order to also capture users connected through topics that might be less prevalent overall but highly descriptive for individual users, we recommend generating a large number of topics, e.g. 50. This strategy was also used for this project. From the outcome of topic modeling we collect the probability scores that indicate the fitness of each topic per document; i.e. user. From here on, we provide four ways of generating social networks from this data, namely:

4.2.1 Cosine Similarity Based Networks

These networks are generated by first creating a topic probability vector for each document, then calculating the cosine similarity for each pair of actors, and linking each pair of actors for who the similarity value is equal to or higher than a user-defined threshold value. We provide a default threshold value of 0.5. In the resulting networks, link weights represent the similarity value, which ranges from the threshold value to 1. The networks are undirected.

$$N = \text{Number of Topics}$$
$$p_i^k = \text{Probability of topic i for sender k}$$
$$s_k = < p_1^k, p_2^k, \ldots, p_N^k > \text{ Vector of topic probabilities for sender } s_k$$

$$\text{Edge Weight } w_{i,j} = \frac{s_i \,.\, s_j}{||s_i|| * ||s_j||}$$

4.2.2 K - Top Similar Topic Cosine Similarity Based Network

This is a variant of the method describe above, with the difference being that the topic probability vector for each document only includes the K common topics shared between any pair of documents. Conceptually, this represents a convergence towards the largest common denominator or consensus among a group of people; penalizing marginalized opinions – which the prior approach does capture.

$$N = \text{Number of Topics}$$
$$K = \text{Number of top topics to compare}$$
$$p_i^k = \text{Probability of topic i for sender k}$$
$$s_{i,K} = < p_1^i, p_2^i, \ldots, p_K^i > \text{ Vector of top K topic probabilities for sender } s_i$$

$$s_{j,K} = < p_1^j, p_2^j, \ldots, p_K^j > \text{ Vector of top K topic probabilities for sender } s_j$$

$$\text{Edge Weight } w_{i,j,K} = \frac{s_{i,K} \,.\, s_{j,K}}{||s_{i,K}|| * ||s_{j,K}||}$$

4.2.3 Max Topic Based Network

In these networks, people are only connected if their highest scoring topics match. The shared top topic is stored as a property of the edge. This enables content-based edge labeling.

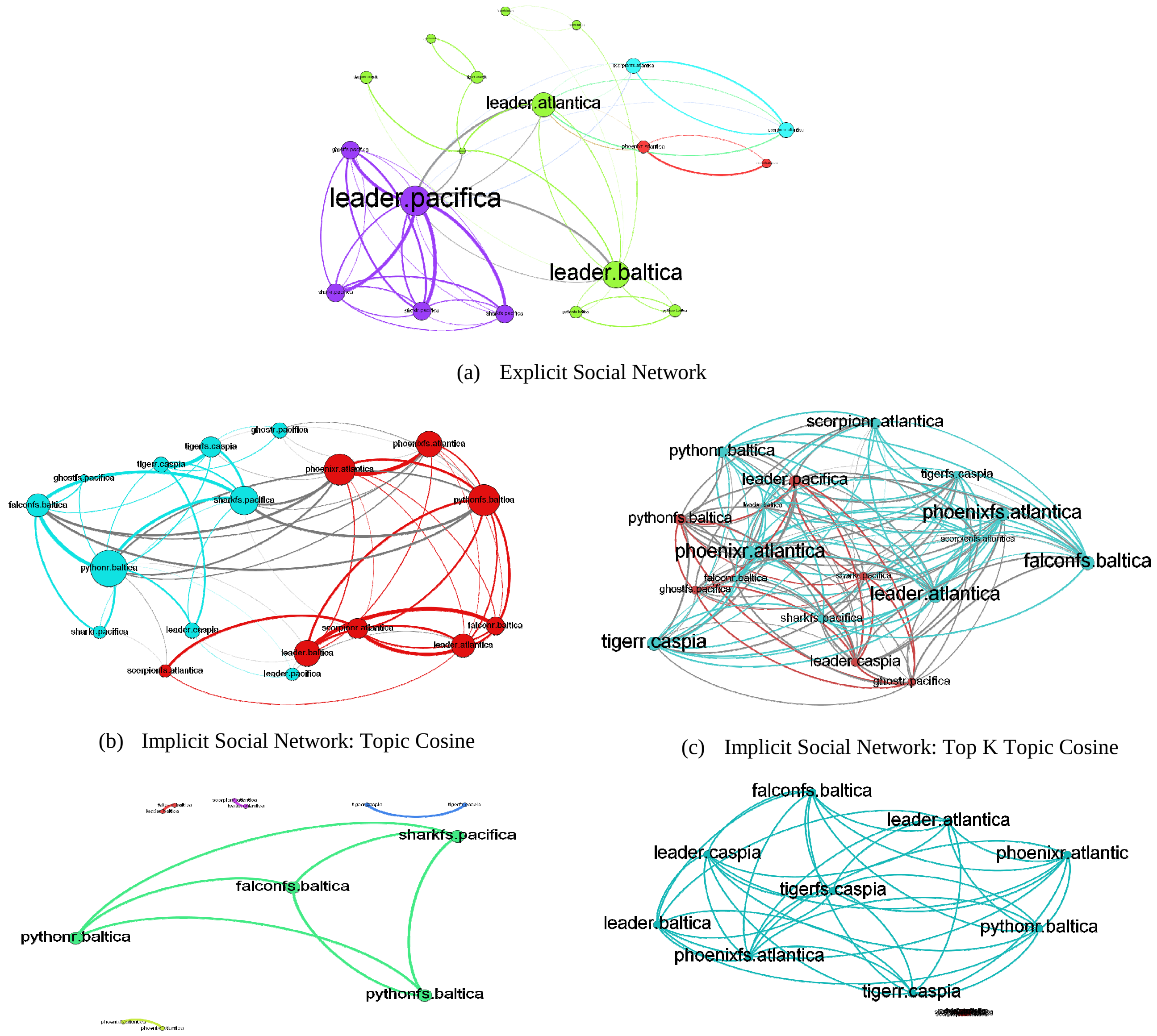
4.2.4 Min Topic Based Network

This is the counterpart to the method provide above. In these networks, people are only connected if their lowest scoring topics match. This network is useful for identifying the most distant people in terms of shared knowledge. The respective topic is stored as an edge property. Note that min topic serves as a pseudo control case or sanity check here – we hypothesize that these networks resemble the explicit social network least, and worse than any other type.

5. Results

To illustrate the types of network produced and compared herein, Figure 1 shows an example for each type for one randomly picked simulation game. Colors represent groups (based on modularity), node sizes are scaled by degree centrality, and node label sizes reflect betweeness centrality. The visualizations were produced in Gephi (https://gephi.org/). Graph a) represents the explicit social network. Graphs b-g show the implicit social networks (two token based networks (Soft-TfIdf, Jaccard) and the four topic based networks (#Topics: 15, Threshold: 0.2, #CommonTopics: 5)).

The point to be made with these images – and this generalizes across the simulation games – is that none of the implicit social networks closely resembles the explicit one. But how far off are they? To answer this question, we produced each of these seven networks for all 33 simulation games. We then compared the implicit networks to the explicit one for each simulation game numerically by computing hamming distances and generated a matrix of respective scores for comparing any two networks (Tables 1.2). Hamming distances basically express the agreement in edge identity between any given pair of graphs.



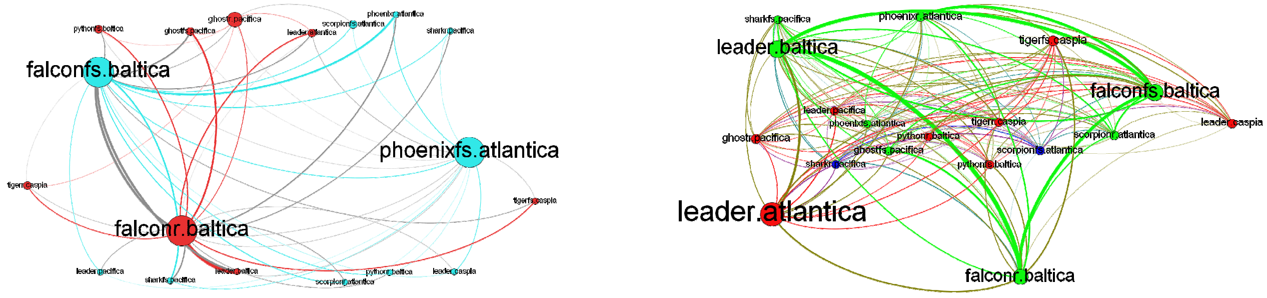
(a) Explicit Social Network

(b) Implicit Social Network: Topic Cosine

(c) Implicit Social Network: Top K Topic Cosine

(d) Implicit Social Network: Max Topic

(e) Implicit Social Network: Min Topic



(f) Implicit Social Network: Jaccard

(g) Implicit Social Network: Soft-TFIDF

Figure 1. Explicit (a) and implicit (b-g) social networks for one simulation game.

To drill deeper into illustrating the topic based networks, Figure 2 shows the connection of individual authors through their top K (=5) topics selected from a pool of 20 topics. Authors (agent nodes) are shown on the left hand side, topics on the right hand side. This is the underlying information used for generating networks based on topic similarity for authors.

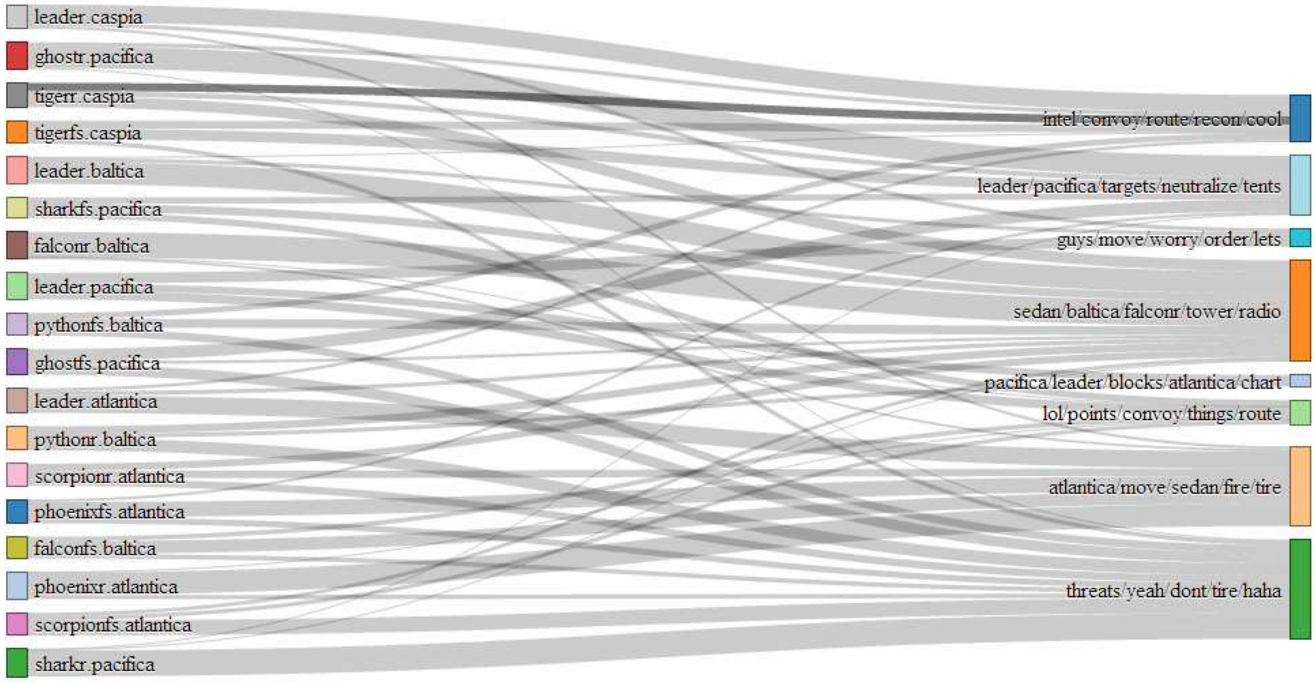


Figure 2. Top 5 topic linkages for authors for one simulation game. Left side: name of authors, right side: topics.

The percentage values in Table 1 quantify the difference between the implicit and explicit social networks. Table 2 aggregates these values into descriptive statistics. To ease the reading of the Tables, values are color coded on a green to red scale representing low to high disagreement. Our results suggest that the networks based on MaxTopic (a summarization/ topic based approach) and Jaccard (a lexical/ token level approach) are most similar to explicit social networks. Differences between the different simulation games exist, but the observed patterns are fairly robust across the games.

As expected, the min topic networks are worse reconstructions of the given social structure than most other types derived from text data content – except for TopKCosineSimilarity networks. One explanation for this discrepancy might be that TopKCosineSimilarity by its algorithmic nature produces a higher number of edges; including a large amount of false positives when it comes to the conducted comparison. However, these additional edges might suggest meaningful further connections between people who share a certain amount of knowledge. Since these latent agreements are not entailed in or visible from the explicitly given structure, text mining based network construction can help to reveal them. Further follow-ups with the participants in the experiments would be needed to verify this assumption.

Additionally, such deviations are a chance to complement or enhance our understanding of a given networks with suggestions for people who have some knowledge or information in common, but never actually talked to each other. These people could be introduced to each other, strategically distributed across work units, or serve as back-ups for their respective functional roles.

Network	Cosine	MaxTopic	MinTopic	TopKCosine	Jaccard	Soft-TFIDF
ChatData#4	8.50%	11.80%	17.30%	38.20%	3.90%	2.90%
ChatData#5	9.20%	6.70%	21.30%	40.80%	3.40%	4.70%
ChatData#6	12.40%	7.30%	24.60%	41.60%	4.50%	8.90%
ChatData#7	7.60%	12.70%	36.00%	40.00%	10.00%	8.90%
ChatData#8	6.30%	7.60%	23.20%	43.20%	1.60%	6.10%
ChatData#11	13.20%	10.60%	21.00%	42.10%	2.60%	9.20%
ChatData#13	12.60%	7.60%	15.00%	41.80%	5.40%	2.40%
ChatData#14	7.40%	3.80%	20.20%	42.60%	33.30%	9.20%
ChatData#15	10.30%	12.60%	29.50%	42.10%	4.90%	2.40%
ChatData#16	12.40%	6.20%	27.80%	42.90%	5.80%	6.80%
ChatData#17	10.50%	6.60%	26.00%	42.40%	10.00%	7.60%
ChatData#18	9.50%	21.20%	27.10%	42.90%	5.80%	5.50%
ChatData#19	11.60%	13.80%	20.50%	44.50%	3.90%	10.80%
ChatData#21	10.00%	8.30%	18.10%	37.60%	6.10%	8.90%
ChatData#23	9.40%	9.00%	22.10%	42.60%	6.80%	10.50%
ChatData#24	6.00%	7.50%	34.30%	38.80%	2.60%	10.70%
ChatData#25	9.20%	8.80%	36.30%	41.10%	5.30%	8.40%
ChatData#26	8.40%	7.60%	17.50%	43.20%	15.00%	1.60%
ChatData#27	11.40%	9.10%	35.60%	42.40%	5.60%	3.80%
ChatData#28	12.60%	12.90%	16.30%	41.30%	7.40%	10.00%
ChatData#29	10.00%	15.90%	26.60%	43.40%	4.70%	4.70%
ChatData#30	10.30%	10.30%	22.60%	40.80%	6.10%	5.00%
ChatData#32	8.20%	10.00%	36.30%	41.10%	3.90%	3.70%
ChatData#33	6.60%	4.50%	20.80%	41.60%	1.80%	2.10%
ChatData#34	8.90%	4.50%	25.10%	38.20%	3.20%	2.60%
ChatData#35	8.90%	8.30%	32.60%	38.20%	4.70%	8.90%
ChatData#37	5.30%	6.40%	28.80%	36.10%	7.10%	15.00%
ChatData#38	4.00%	6.80%	29.50%	41.90%	3.80%	3.10%
ChatData#39	7.10%	9.00%	35.40%	39.20%	8.40%	3.20%
ChatData#40	10.50%	10.00%	21.60%	45.00%	16.70%	4.50%
ChatData#41	10.50%	10.30%	22.10%	39.50%	8.50%	10.50%
ChatData#42	6.70%	9.10%	21.90%	41.50%	1.60%	1.50%
ChatData#43	9.20%	9.90%	48.40%	37.40%	4.20%	2.60%

Table 1. Comparison of percentage difference between underlying networks and each similarity based networks.

METRIC	Cosine	MaxTopic	MinTopic	TopKCosine	Jaccard	SoftTfIdf
Mean	34.42	14.61	89.88	155.27	14.82	23.76
Std. Deviat.	8.69	6.80	26.62	11.27	8.77	13.26
Variance	75.46	46.30	708.41	126.93	76.88	175.70
Max	50.00	33.00	148.00	176.00	32.00	57.00
Min	17.00	7.00	46.00	117.00	0.00	5.00
Median	35.00	13.00	85.00	158.00	15.00	21.00
Avrg. Deviat.	63.85	39.18	599.42	107.40	65.05	148.67

Table 2. Aggregated statistics over hamming distances.

6. DISCUSSION AND CONCLUSION

We provide new empirical insights into the relationship between social networks constructed from a) explicit data on network participants and the fact of communication exchange between them and b) based on the similarity of text data produced by these agents. Overall, the more simplistic approach on the lexical level (token based networks) outperforms more complex, topic based methods. This means that explicit social networks are best approximated by sticking to similarities on the word use level. More advanced representation of language use – in this case the summarization of an agent's utterances into emerging themes – lead to network structures that deviate from explicitly given structures more strongly. This means that reconstructing social network data based on lexical features is the best option tested, while detecting alternative latent structure of people who share the same topical knowledge requires looking for thematic clusters of word use.

Our findings are limited by the empirical data used and the techniques considered. While we did analyze 33 different communication sessions with different people playing the same roles, all data come from one particular domain; namely planning courses of action between collaborating individuals. We plan to address this limitation by also working with chat logs and communication data from other topic domains. We anticipate the outcome of this process to calibrate our findings presented herein. We also aim to experiment with additional methods for inferring social structure from text data, including the relation extraction approach explained in the background section. We also plan to enhance our findings by conducting deeper error analysis to understand the false positives and false negatives that the implicit networks contain.

Finally, we will study how the implicit networks compare to each other, and try to identify how explicit social networks can best be enhanced with implicit ones and vice versa to gain a more comprehensive understanding of socio-semantic networks.

7. ACKNOWLEDGMENTS

The authors wish to acknowledge the following funding sources that contributed to the development of materials presented in this article: Army Research Institute W5J9CQ-12-C-0017; Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053; and National Science Foundation Grants: BCS- 0940851, CNS-1010904, OCI-0904356. The views, opinions, and/or findings contained in this report are those of the authors, and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documents. We are also grateful to the Ford Foundation 0125-6162 for supporting the development of ConText.

8. REFERENCES

1. Corman, S.R., et al., *Studying Complex Discursive Systems: Centering Resonance Analysis of Communication.* Human Communication Research, 2002. **28**(2): p. 157-206.
2. Danowski, J.A., *Network Analysis of Message Content.* Progress in Communication Sciences, 1993. **12**: p. 198-221.
3. Alderson, D., *Catching the'network science'bug: Insight and opportunity for the operations researcher.* Operations Research, 2008. **56**(5): p. 1047-1065.
4. Milroy, J. and L. Milroy, *Linguistic change, social network and speaker innovation.* Journal of Linguistics, 1985. **21**: p. 339-384.
5. Roth, C. and J. Cointet, *Social and semantic coevolution in knowledge networks.* Social Networks, 2010. **32**(1): p. 16-29.
6. Diesner, J., *Uncovering and managing the impact of methodological choices for the computational construction of socio-technical networks from texts.* 2012, Carnegie Mellon University.
7. Roth, D. and W. Yih. *Probabilistic reasoning for entity and relation recognition*. in *International Conference on Computational Linguistics (COLING).* 2002. Taipei, Taiwan.
8. Zelenko, D., C. Aone, and A. Richardella, *Kernel methods for relation extraction.* The Journal of Machine Learning Research, 2003. **3**: p. 1083-1106.
9. Bunescu, R. and R. Mooney, *Subsequence kernels for relation extraction.* ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 2006. **18**: p. 171.
10. Diesner, J., *From Texts to Networks: Detecting and Managing the Impact of Methodological Choices for Extracting Network Data from Text Data.* Künstliche Intelligenz/ Artificial Intelligence, 2013. **27**(1): p. 75-78.
11. Gloor, P. and Y. Zhao. *Analyzing actors and their discussion topics by semantic social network analysis.* in *10th IEEE International Conference on Information Visualisation* 2006. London, UK: Citeseer.

12. Cohen, W.W., P. Ravikumar, and S. Fienberg. *A comparison of string metrics for matching names and records*. in *KDD Workshop on Data Cleaning and Object Consolidation*. 2003. Washington, DC.
13. McCallum, A., X. Wang, and N. Mohanty, *Joint Group and Topic Discovery from Relations and Text*, in *Statistical Network Analysis: Models, Issues, and New Directions. Lecture Notes in Computer Science 4503*. 2007. p. 28-44.
14. Blei, D., A. Ng, and M. Jordan, *Latent dirichlet allocation.* The Journal of Machine Learning Research, 2003. **3**: p. 993-1022.
15. McCallum, A.K. *MALLET: A Machine Learning for Language Toolkit*. 2002; Available from: http://mallet.cs.umass.edu.