

Construction of Association Networks from Communication in Teams working on Complex Projects

Laura Dabbish^{1*}, Ben Towne², Jana Diesner³ and James Herbsleb²

¹*The H. John Heinz III College and Human-Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA*

²*Institute for Software Research, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA*

³*Center for Computational Analysis of Social and Organizational Systems, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA*

Received 29 December 2010; revised 5 August 2011; accepted 10 August 2011

DOI:10.1002/sam.10135

Published online in Wiley Online Library (wileyonlinelibrary.com).

Abstract: This paper introduces a semi-automated method for extracting the associations among components of a complex engineering project by mining transcripts of design discussions about the engineered artifact. We contribute a novel method for deriving network data from communication data. This method uses relational text analysis to construct networks representing associations among designed pieces of the artifact. Our analysis shows that association networks extracted with this method correlate highly with those extracted by humans, and those constructed based on expert interviews. This method can help scholars and practitioners understand the nature and evolution of the relationship between tasks and components, and the successful execution of complex projects. The association networks produced by our method can be used to reveal key properties of the project and may suggest the coordination required among project members. Collaboration technologies could potentially take advantage of these networks to infer gaps in actual coordination, discover changes in dependencies among team members as they emerge, and suggest appropriate courses of action to managers and people whose work is related. © 2011 Wiley Periodicals, Inc. *Statistical Analysis and Data Mining* 4: 547–563, 2011

Keywords: association networks; conversation analysis; natural language processing; project management; coordination and collaboration in teams; managing complexity; product development

1. INTRODUCTION

Prior research has introduced techniques for identifying relationships among resources and activities in collaborative projects [1–4]. These techniques have primarily utilized data in the form of recorded actions on electronic artifacts. In software engineering, for example, researchers have used logs of information about concurrently modified source code files to construct network representations of technical associations among software components [4]. Analyses of the resulting network data have shown utility for indicating coordination requirements among people. The extent to which detected associations between components match the communication network predicts project performance across several different settings [4]. These associations can indicate places where project members' work may

influence one another, and may indicate what the structure of communication on a project should look like in order to manage conflicts that can occur as a function of task associations. However, explicit traces of socio-technical actions and relations are not available in the vast majority of collaborative projects. In contrast, conversation data provides a more readily available alternative source of data for detecting associations among resources and tasks in a project, e.g. in the form of voice recordings of meetings. Conversation data is a natural byproduct of offline interactions such as face-to-face discussions and on-site group meetings. These interactions are still commonly used in many collocated projects to coordinate collaborative work. Several network-centric methods exist for constructing association networks from text data [5]. However, these methods typically trade off a deep understanding of the content of the text data for computational efficiency and scalability [6,7].

Correspondence to: Laura Dabbish (dabbish@cmu.edu)

In this paper, we introduce a semi-automated method for extracting association networks from text data. The method allows for fast processing of vast numbers of documents while supporting a deep understanding of associations between components of technical artifacts through a domain-specific text coding approach. This method can be used when electronic activity trace data is not available, taking as input communication data from individuals or groups as an input, and producing as output network data representing associations between key concepts in the communication data. These key concepts can be, for instance, resources such as technical components, task that have to be solved, or information that is exchanged between team members. In this paper, we focus on technical components of complex, engineered artifacts that are designed by large teams whose members are not necessarily collocated. Relevant data sources for this method include transcripts from meetings, email conversations, and chat logs. The resulting network data have a variety of research and practical uses: for example, members of project teams and their managers can discover and act upon technical relationships among components of a project, e.g. by initiating communication between people who work on components for which a new interdependency has emerged, or by routing information updates about changes to interconnected components to only the appropriate members of the project. This method is particularly valuable when explicit and less noisy data sources, such as design artifact versioning systems, are not available. Moreover, the method does not require the team members or other experts' active attention to discover, draw, and continually update the network data. The support provided by this method is essential when the interdependencies of components of artifacts are too complex to be tracked manually or to be agreed upon by multiple parties.

In this paper, we describe the proposed method in detail and scrutinize its underlying assumptions. We then apply this method to extract association networks from transcripts of meetings about a lunar robotics engineering project. We compare the association networks generated by our method with networks extracted using other procedures. Our results suggest that the networks generated by the proposed method are accurate in that they correlate highly with those generated by human coders and fairly well with networks derived from field notes of expert interviews, although they do not seem to correlate with networks derived from a design structure matrix (DSM)-based on expert's perceptions.

Finally, we use the association networks generated from our sample dataset to illustrate some of the insights that can be gained from applying network analysis to the extracted data. In examining network evolution over time in our sample dataset, we find that the examined

association networks converge to represent the full set of observed project dependencies after a few rounds of observations. We discuss the potential utility of the observed networks for project management. For example, node degree when compared with simple mentions of a component may imply the level of complexity in coordination issues associated with that component; e.g. some associations, although they involve many different components, appear to be resolved quickly with little discussion, while other component relationships require extensive discussion over many meetings although they involve fewer components.

The presented method is interdisciplinary in that it combines natural language processing (NLP) with network analysis. Moreover, this method expands the scope of alternative methods for constructing collaboration and knowledge networks by not only considering the explicit accounts of human interaction but also by utilizing the content of communication data these interactions produce. The proposed method adds to the set of available methods for conducting document analysis from a network analytical perspective. Collaboration technologies can exploit networks constructed this way to discover changes in associations among components and respective tasks as they emerge, infer differences between actual and project-required coordination, and suggest appropriate courses of action to individuals or subgroups affected by project dynamism. Beyond the application for project management purposes, association networks such as those constructed with this method may help scholars and practitioners analyze the nature and time course of the relationship between resources, tasks, people, and success in executing complex team projects.

This paper is structured as follows: the Related Work section puts our method into the context of alternative techniques for distilling networks from text data sources, and highlights advantages and limitations with our method. In the Method section, we describe our network construction technique, the assumptions with this method, and an illustrative application of the method to a real-world dataset. This dataset consists of meeting transcripts from a complex lunar robotics engineering project. Next, we present the results comparing the networks generated using our method with three comparison networks: (i) networks generated by human coders on the same meeting transcript data, (ii) a network of associations in the same project generated using different data (analysis of expert interviews), and (iii) a network of associations generated based on expert observer perceptions (akin to DSM analysis). Finally, we discuss potential applications of the networks constructed using our method, and consider how characteristics of constructed association networks could impact coordination in complex projects.

2. RELATED WORK

2.1. Association Network Construction

In order to construct association networks, we build on previous research in software engineering that successfully computed highly volatile coordination requirements from the interaction of humans through technical systems [4,8]. As noted above, the source for that work was data automatically generated by software development tools such as code versioning systems. The activities explicitly logged by these tools were converted into a network representing task associations inferred from component interdependencies in the following way: for all of the files checked in by developers in one single session, a logical task dependency among these files was assumed and drawn in the task dependency network. The task assignment network was constructed by mapping software developers to the files (tasks) they had been uniquely assigned to. Once both task dependency and task assignment networks were constructed, coordination requirements were computed by multiplying the task assignment network by the task dependency network by the transpose of the task assignment network. This matrix algebra operation produced a network representing the coordination requirements between developers (where links among individuals in the network indicated association between those individuals' tasks). Next, a coordination congruence metric was computed by comparing the coordination requirement network with the network of actual communication among project developers. Analysis of this data revealed that task dependencies change rapidly over time, and that a networked representation of congruence was useful for predicting project performance [4,8]. Projects with higher congruence between the coordination requirement network and the actual communication network had higher project performance [4,8]. This stream of previous work introduces the concept of 'logical associations,'¹ referring to relationships between technical artifacts that imply a need for certain human actions and interactions. For most real-world projects, however, such explicit data about a project's development path are not readily available in electronic form, particularly for work other than the outcome of online production systems. Thus, we consider alternative, more common sources of this data and techniques for extracting network data from these sources.

2.2. Extracting Networks from Communication Data

Communication data may provide a suitable alternative to electronic activity trace data and expert inputs. Collaborators talk to each other online and offline, and

their conversations contain explicit and implicit information about dependencies [9]. During such conversations, project members may discuss alternative solutions to particular design problems, benefits, and tradeoffs of pursuing these solutions, and decide among the alternatives. These discussions can reveal important associations among the components within complex projects beyond the limited information available from teams' online production systems.

Previous research suggests that association networks can be constructed from communication data: meeting transcripts have served as a source for constructing social network data [10] and studying the process of knowledge construction in design teams [11]. Several methods for extracting network data from natural language text exist (for an overview see ref. 5). These methods have emerged from different disciplines, mainly anthropology and social science [12,13], communication science [14,15], political science [7,16], and artificial intelligence and NLP [17–19]. These methods differ with respect to their theoretical grounding and assumptions, degree of automation, and generalizability across domains and genres [5]. The rest of this section compares these methods to the one we propose, outlining the advantages of our method over these previous approaches.

When using qualitative text coding methods [12,20] and grounded theory methodology [21], humans identify and document the relevant concepts, also referred to as codes, and the connections between those codes. There are several software tools that facilitate the coding process and the management and visualization of the data. These methods mainly support data exploration, in-depth analysis of small amounts of data, and hypothesis development. They are expensive in terms of manual efforts and require individual interpretations of terms and concepts for each new word or phrase [7]. The method we present herein differs from these methods in that the stage of identifying relevant concepts and their instances in the text data is followed by a systematic and automated assignment of codes to terms and short phrases.

Also, the connections between codes are not manually identified on a case-by-case basis, but the same mechanism is applied to all data at once. This procedure makes the coding process faster, more systematic, and easier to reproduce and modify. More specifically, we use a proximity-based approach that connects any pair of nodes co-occurring within a user-defined text unit [15]. We elaborate on our implementation of this approach in the methods section. Alternative approaches to link formation include semantic relations between words which often are specified in ontologies [6], logical relations, such as 'is a', 'is like' [22], which typically only cover a small set of relations [23], and syntactic relations, which are predefined in grammars [24,25]. Our proximity-based approach ignores lexical, syntactic,

¹ Note that this notion of logic differs from the meaning of logical links in artificial intelligence and information extraction [9].

and semantic details of the involved concepts and their context. This minimizes the amount of effort required for domain adaptation and rule expansion to cover deviations, but may increase the possibility of false positives.

In contrast to qualitative methods, other methods have automated the identification of nodes and edges by using predefined positive filters such as lists and thesauri [16], regular expressions that denote abstract patterns of the occurrence of symbols, and rules tailored to specific domains and genres [18]. In general, the deterministic nature of predefined lexical entries, patterns and rules can result in false positives, i.e. unintended matches, and false negatives, i.e. missing relevant entries, which can occur due to typos and deviations from assumed regularities. These limitations can be addressed by using NLP methods such as stemming [26,27], i.e. reducing terms to their root, thus eliminating lexical and syntactic variations (which our method employs), and the automated identification of key terms via (weighted) frequency metrics [28]. The method we propose also uses thesauri, but involves a systematic, computer-assisted process to ensure a high degree of fit and completeness of the thesauri for the data. Our thesaurus creation process employs various NLP methods and is documented in the methods section. While this process involves human-guided domain adaptation, it ensures a high degree of concept coverage in the given domain; compromising cross-domain generalizability for high concept recall rates.

Furthermore, there are a few fully automated methods that facilitate the extraction of network data from text data without relying on machine learning techniques. An example is centering resonance analysis (CRA) [6]. In CRA, noun phrases are first identified via parts of speech tagging, i.e. a grammatical function is assigned to each word, and then linked into networks based on their sequence in sentences. Next, 'influence' measures are computed over the retrieved noun phrases which indicate the prominence (influence) and uniqueness (resonance) of individual phrases. Finally, circular visualization of the most influential nodes and links are generated, where higher influence is represented as proximity to the center of the circle. Unlike our method, links in CRA are strictly based on sequences, not proximity. Moreover, in CRA, there is no thesaurus that allows for converting meaningful multiword units, such as 'lunar lander,' into a single concept. Finally, although influence and resonance might be interesting measures in the context of building artifacts and managing projects, they do not capture the proximity-based associations that we designed our method to detect. Moreover, our method is not confined to certain parts of speech, thus allowing consideration of modifiers and action words if needed.

Finally, the flexibility and accuracy of locating nodes and edges can be further increased using machine learning

methods [29,30]. Building a model that identifies instances of relevant entity and node classes involves significant engineering effort and time for training, evaluating and refining a model. These up-front investments may be more time consuming than the method we propose herein.

Other, non text-based methods for inferring associations have limitations which may make them less desirable in comparison. For example, *a priori* system architecture diagrams can quickly become obsolete. Also, having project participants create a dependency network manually is extremely time consuming and likely subject to bias as a result of their varying familiarity with the different components in the system. In contrast, our method provides an overview of the associations among components across the entire system, a view which may be difficult for any single project member to understand. Our method should be high in accuracy because it is emergent and based on interactions around the design itself.

3. METHOD

We formalize the representation of network data extracted with the presented method from communication data as follows: a single network represents the technical components and their associations from one project meeting. In these networks, each node represents a unique subsystem or aspect of the project, e.g. physical parts like a motor or a battery. A link represents a discussion about two involved components, e.g. mentioning the fact that the power from the selected battery must be sufficient for the motor.

Figure 1 outlines the main steps that are part of our method. It also shows additional steps that are necessary precursors, but not a core part of the method.

3.1. Input Acquisition and Preparation

Our method takes as input a set of communication data from a collaborative project. This data needs to be in text format. The method assumes that transcription has already been applied to any audio recordings of communication data. This means meeting observation and transcription (via human transcribers or speech-to-text converters) must be done prior to applying our method to meeting data. While this step is not a part of our method, we note that our method will be more useful when speech recognition and automated transcription technology reaches a high level of accuracy, making this input available at lower cost.

3.2. Node Definition

As a second input, our method requires definition of relevant entities that are converted into nodes for networks.

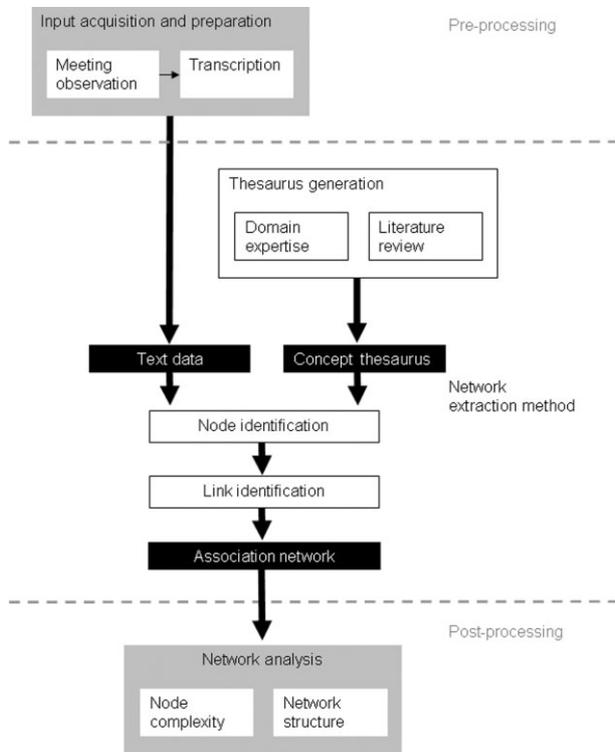


Fig. 1 Method workflow diagram.

These entities represent key components. This step involves two key decisions:

First, one can select from using automated and/or manual approaches for node identification. Automatic techniques identify a list of potential key concepts on their own by using term weighting techniques [31]. Another family of automated techniques are unsupervised machine learning methods such as topic modeling and clustering that result in term clusters which represent the most descriptive dimensions of a corpus [32]. Manual techniques are close to the aforementioned qualitative coding techniques, where humans read all or portions of the text data in order to identify key concepts. The individuals creating the node definitions might also consult with subject matter experts or study accounts of subject matter expertise on this task. The non-standardized, idiosyncratic nature of engineering and product development processes means that project-specific expertise might be necessary for creating data that allows for meaningful analysis. We suggest a combination of manual and automated steps for creating a node definition codebook, using our sample application of the method as a model: We first consulted project members and field-specific publications to develop a comprehensive list of key components of the lunar robotics project used for our sample dataset. Then we combined all of our meeting transcripts with field-specific publications and manually examined the most frequently occurring concepts to ensure

that we were not inadvertently missing key concepts from the text. We did this by using the AutoMap software to create lists of most frequent terms both on the unigram level (single words) and the bigram level (two word pairs that represent a single concept, such as ‘solar panel’) [33] and then checking those lists for representativeness.

Second, one needs to select an appropriate level of granularity of concepts, in our case components, considered for analysis. By tailoring the nodes exactly to the project under study, the network may perfectly represent the given artifact at that point in time. However, such a specific representation may overfit to the data or quickly become obsolete as the project evolves across time and teams. Therefore, in our sample application we desired a component list specific enough to represent the particular artifact being developed, but also general enough to account for changes in the details of the artifact. Thus, we recommend considering components commonly present in the general class of artifact under consideration.

3.3. Construction of Codebook and Thesaurus

The associations between specific terms with more general components are documented in a codebook. A codebook specifies the guidelines for node and edge identification and contains enough details to ensure reproducibility of the method and results. It also helps the analysts document their coding choices and provides examples for their codes. Finally, the codebook serves as the basis for constructing a thesaurus, which is a second input to our method. A thesaurus is a multi-column data structure that associates text terms with higher level concepts. This mapping from text terms to concepts supports analysis on different levels of granularity as appropriate for the given research question; from highly detailed to highly aggregated. One or more levels of association are possible. The thesaurus created for our sample application of the method has two columns, mapping text terms representing specific pieces of a particular project (left-hand side entries), e.g. ‘camera,’ to more general component categories (right-hand side entry), e.g. ‘sensors.’ Each component (left side of Table 1) describes one particular area of functionality.

Building a codebook and thesaurus is an iterative manual process that occupies tens of hours of researcher time and involves human decisions that impact the resulting network structure. In comparison to fully automated node identification methods, our method has limitations in terms of costs and time. However, it does enable the consideration of subject matter expertise, which can be essential for distilling meaningful network data. Compared to manual text coding approaches, it takes less time to apply a thesaurus in a computer-supported fashion than to train humans in this task. Moreover, the one-time up-front effort in codebook

Table 1. Codebook listing and describing key components.

Component	Brief description
Communications	Communications external to the robot, for control or mission, including operator interface
External relations	Acquiring external resources (including funding, parts, and purchasing), publicity, investor and media relations
Internal relations	Project/program management, HR, task assignments, training, collaboration tools, clarifying norms, and expectations
Mobility effectors/actuators	Effectors and actuators that propel the entire robot: e.g. tracks, wheels, shocks, and motors with associated firmware
Mission specific effectors/actuators	All other motors, gears, and moving parts that do not move the robot as a whole, e.g. camera mast rotation motor
Perception software/computing	Software and any dedicated hardware, for terrain mapping, environmental modeling, and/or object detection. Camera/lens zoom, shutter, and focus control software
Planning software/computing	Mission task planning, including the overall mission plan and computing resources for semi-autonomous execution
Power	Includes batteries, solar cells, switches, power cables, and controls
Sensors	Camera; thermal, ultrasonic, tactile, radar/sonar range sensors; inertial measurement unit, GPS, and any wiring or processing going from sensors to controls
Shared/general computing	Includes general purpose processors/onboard computers (e.g. avionics box). Abbreviated 'gpp.'
Structure	Chassis, fasteners (e.g. Frangibolt, weld joints), radiator, payload, paints, reflectors

and thesaurus construction pays off quickly when the thesaurus is applied repeatedly, or modifications to the coding scheme are introduced and recoding becomes necessary.

Our codebook was created by first conducting multiple interviews with the manager of the lunar robotics project, who is an expert on the topic. He delineated a component list that was very specific to the particular project. For example, his component list had a specific keyword for the 'camera/mast assembly,' which allowed the lunar robot to look around on the moon. While the components identified through these interviews were essential for the given project, they are unlikely to generalize to other technical product development projects, or even to a significant redesign of the given project. In order to overcome this limitation, we built a thesaurus that can also be used for studying significant redesigns of the given project as well as other field robotics projects. This coverage extension of the codebook and thesaurus means our method can assist in addressing research questions that involve comparing and contrasting multiple projects in the field robotics domain. Based on this rationale, we built a codebook and thesaurus that map specific components of our sample project to general functional areas of a robot. To learn what components and component categories are common across robotics domains, we acquired information about three distinct subfields: Search & rescue robotics, interplanetary robotics (of which our sample project was one), and autonomous urban vehicles (e.g. the DARPA Grand Challenge). We reviewed articles written by subject matter experts in these subfields to identify the common components and component categories that they perceive as distinct and important parts of a robot and its design [34–38]. The parts we identified are shown in Table 1. Finally, we added this level of generality to our

codebook and thesaurus: We revisited the component list created in consultation with the project manager and associated each of their components with a general robotics component category. The resulting full thesaurus (available on request) can be used, extended and modified by others interested in analyzing complex field robotics projects.

Once the relevant components and higher level categories are identified, their instances in the text data have to be specified in the thesaurus. The thesaurus maps text terms and their synonyms, abbreviations and spelling variations to predefined relevant components. This mapping can be done in a computer-assisted, manual fashion where humans read through portions of the data to find terms that refer to each component or concept in the codebook. Our method uses the initial thesaurus constructed as described above as a baseline. We constructed our initial thesaurus in consultation with project managers and based on scientific literature from the application domain.

To further refine and extend the thesaurus, we performed text mining on the source data: First, we applied stemming to the text data to convert words into their morphemes; increasing the coverage per term. We used the Krovetz stemmer [27] as provided in the AutoMap software [33]. Second, we used AutoMap to generate a list of cumulative frequencies for all unigrams and bigrams. We added the highly frequent unigrams (≥ 20 cumulative frequency), e.g. 'wheel', and bigrams, e.g. 'solar cell' that were relevant for our project to the thesaurus and mapped the new entries to a corresponding higher level component. Common misspellings (e.g. 'chasis'/'chassis'), additional synonyms (e.g. 'buck'/'dollar'), and word variants that the stemmer had missed (e.g. 'antenna'/'antennae') were also revealed and corrected during this stage.

3.4. Automated Node Identification

Our method automatically identifies the predefined nodes in the text data by applying the thesaurus. During this process, text terms identical to a left-hand side entry in the thesaurus are replaced with the corresponding right-hand side entry (see Table 1 for a complete list of components and a brief description of each). The original distances (number of words in the transcript) between terms are maintained in this process, which is crucial when the proximity of terms is used for link creation. We used the AutoMap software for thesaurus application.

3.5. Automated Link Identification and Network Data Extraction

To identify links between nodes, our method uses a proximity-based approach [15]. This approach is based on the assumption that components mentioned together in the same window of a user-defined number of consecutive words are related to each other. The link weight represents the number of times that two components co-occurred in the user-defined window.

The key issue with this approach is finding an appropriate window size. There are no standards for determining an appropriate window size [6]. Moreover, the window size impacts the properties of the retrieved networks. Finally, the right window size may vary from corpus to corpus. In order to find an appropriate window size, we propose running the analysis with various window sizes and comparing the results against human-coded networks in order to find an automated setting that most closely resembles human coding. This approach assumes human coding to be a gold standard, which can be argued against, especially for a task as complex and cognitively demanding as coding linear structures into network data [39]. However, it does establish a clear point of evaluation which has been previously used by others [6].

We tested various window sizes ranging from 5 to 400 and used QAP to calculate the correlation between the human-coded and machine-coded networks. QAP is a nonparametric test that measures similarity between two matrices of the same size, in our case the human coded and machine coded networks [40,41]. For computing QAP, we used the ORA software [42]. We found the correlation to be highest for window sizes around 15, which we subsequently chose for our analysis.

Finally, a sliding window is run over the text data after the aforementioned processes, such as thesaurus application, have already been performed. A link between any two concepts is recorded any time the concepts co-occur within the same window. Using this procedure, we constructed one association network per meeting in our

sample dataset. Again, we used the AutoMap tool for this process. In AutoMap, the user decides not only the window length, but also whether the window should stop at or span across the boundary of sentences or paragraphs where applicable. We chose not to reset the window at the end of sentences and paragraphs, because the associations we wished to capture were often represented between speakers, and each speaker was given a separate paragraph in the transcription.

3.6. Assumptions with the Proposed Method

The proposed method involves several assumptions: First, our method assumes that component relationships are explicitly addressed in project meetings and other conversations among project members. If this assumption holds, natural language text data from the meetings can be used as a source for revealing these associations among components.

Next, the extraction of association networks involves the possibility of false positives and false negatives. False positives can occur, for instance, when a person who is responsible for managing two independent parts mentions both parts in a discussion about budgeting. It can be argued that this may imply an interdependency between components in some sense, e.g. for project management. False negatives can be due to the fact that obvious and unproblematic associations are not mentioned, for example. Both examples for errors are a function of the nature of the data and will not be corrected with the proposed method. The method itself can also lead to false positives, e.g. when a relevant term has a second meaning that does not relate to the domain of interest, and also to false negatives, e.g. due to the deterministic nature of thesauri.

The networks generated with our method are symmetric and undirected. We made this choice to account for the fact that the ordering in which concepts are mentioned allows for variations in the linguistic input that the speaker had chosen to present. This is because once a speaker has followed the syntactic rules for generating a natural language utterance in a given language she is still left with a plethora of choices for formulating a sentence. For example, in an enumeration of technical parts, the order in which the parts are mentioned can vary depending on the speaker's memory and personal preferences for ordering.

Network extraction from text data based on term or concept co-occurrence has been previously criticized for a lack of theoretical grounding [6]. However, we provide connection to theory that helps to justify the proposed procedure: Prior research has identified design decision discussions as one type of technically focused communication that has important implications for product development [43]. This observation is theoretically supported by Conway's law [44], which postulates that organizations design and

produce systems that mirror the organization's communication structure. Conway's law provides theoretical grounding for the assumption that the content of communication data, as a network of who is speaking to whom, can be used to build association networks, which can then be used to infer task associations and respective coordination requirements. Linking prior work from software engineering on 'logical associations' to Conway's law, we can also assume that if components are never or seldom co-mentioned in a discussion, they are likely to be independent from each other. Analogously, if two components are co-mentioned often, Conway's law provides a theoretical base for assuming an association or link between these components.

The method we present is a baseline that can be further extended: The method as described results in a straightforward and simple representation of network data: All nodes and all links are of the same type, i.e. the method facilitates the extraction of one-mode, uniplex networks. The method and technology used for applying the method lend themselves to the extraction of more complex network representations: the nodes and/ or links can be further classified into multiple categories [29]. Table 2 summarizes the steps associated with our method, the decisions required in each step, and the choices we made in our sample application of the method.

4. DATA

The data used to illustrate our method are weekly status meetings on a lunar robotics project. The project was an engineering venture assembled to compete in Google's Lunar X-PRIZE challenge to send a mobile robot to the moon (for more information on the competition refer to:

<http://googlelunarxprize.org>). The meetings we considered were the venue for all project members to discuss particularly important design changes and decisions they were making, and to air issues that hadn't been dealt with informally outside of the meetings. An additional goal of the meetings was to deal with unhandled exceptions and cross-component dependencies. We recommend choosing this type of project-wide communication for obtaining explicit or implicit information about associations between concepts. Such data can be useful for addressing research questions about the identification and management of project-wide associations as well as particularly difficult component linkages.

We conducted an in-depth field observation of this project over a period of ten months to inform our understanding of task associations in research engineering and development work. The project was a complex effort, with very tight coupling among tasks performed by four organizations. Work during the period of observation occurred primarily at one site, and involved a mostly collocated group of about 30 engineers from various disciplines including mechanical, electrical, structural, software, and systems engineering. Teams were generally organized around these disciplines and each team had a technical leader. The work style was highly collaborative, with fluid responsibilities and constant lateral communication. Engineers at other sites served primarily as technical resources, advising the team and reviewing their progress. Several team members worked in a large 'project room' which was an open space with desks, chairs, computers, and telephones. A few team members were located in the 'high bay,' a much larger space nearby used to assemble and test robots. The other team members had individual offices in the same building.

Table 2. Summary of method steps.

Step	Decisions required	Choices we made
1. Input acquisition and preparation	Data sources	Conversation data where implications of project level decisions are discussed and entire project membership is represented
2. Node definition	Use of automated versus manual node identification Concept granularity Identification of project components higher level categories	Combination of manual and automated steps Generality over project specificity Iteratively consult with subject matter experts and literature in project domain
3. Thesaurus construction	Process for identifying and mapping text terms and their synonym and spelling variants Frequency cutoffs for integration in the thesaurus	Convert words into their morphemes, use text mining software to generate term frequency list, add highly frequent unigrams and bigrams to thesaurus
4. Automated node identification	Level of thesaurus granularity applied to text data	Highest level denoted in thesaurus
5. Automated link identification	Window size selection for proximity analysis Whether window should span across sentence boundaries	Comparison of sliding window of various window sizes against human-coded networks

Team members attended a weekly ‘all-hands’ meeting, typically 1–2 hours in length (for the meetings we consider). At these meetings, the technical leader for each area of the project, (e.g. mechanical, avionics, etc.), would present a summary of the week’s activities, the decisions that were made, and decisions that needed to be made. Other meeting participants would pose questions, discuss the consequences of the decisions, and suggest alternatives for problems that arose. At these meetings everyone was collocated except for one systems engineer at a remote location who participated via telephone. Information from outside experts was frequently presented at the meeting by the engineers who had consulted with the experts, and additional questions for the experts were often formulated at the meetings. These meetings were unique from other in-person technical communication because they were the primary venue for raising questions and concerns that bridged across team or component boundaries. Most technical work during the rest of the week was focused inward on each team’s component, resolving intra-team or intra-component issues.

We focused our analysis on the interactions that occurred during the all-hands meetings, because these meetings were the group’s primary project-wide forum. Issues that had not been recognized or resolved during the week often triggered extensive discussions at these meetings. We therefore saw these interactions as the place where the more far-reaching consequences and the most complex decisions tended to receive the greatest attention. By focusing on these meetings we believe we were able to observe most of the difficult technical coordination issues, particularly those which involved multiple robot components and revealed important intercomponent associations. Technical work and discussion also took place elsewhere, but were not captured in our sample data if these events were not brought up in the all-hands meetings.

During our observation period, we video and audio recorded the all-hands meetings. These meetings were then manually transcribed. This resulted in a total of 25 meeting transcripts. Five meetings were disregarded for the analysis presented in this paper because of poor audio quality. The average transcript contained over 10,000 words.

5. METHOD EVALUATION

We applied the proposed method to the meeting transcripts. In this section, we report on comparison of the distilled association networks with networks generated using three alternative techniques.

An important question this research raises is: what is the appropriate way to validate an association network? In order to gauge the validity of our method, we compare the

extracted network data against data constructed using three alternative network construction techniques:

1. Hand-coded networks based on the same data.
2. An association network drawn from expert interviews.
3. An association network based on a DSM style analysis by project observers.

We do not assume or expect the latter two types of networks to resemble the machine and hand-coded networks based on the meeting transcripts, but to provide additional angles from which the data can be analyzed. We argue that these multiple networks can provide a more comprehensive view of the project than a single method for network construction. This is because we assume these different methods capture different views on relations that cannot be directly observed, allowing us to contrast these views. The correlations between these types of networks as reported in this section give a sense of whether we are observing the same information or whether we are measuring different parts of the association network.

5.1. Comparison with Human Coding

In order to assess the accuracy of the automatically extracted networks, we compared them to networks built by hand. Two authors of the paper served as human coders. One human first segmented each transcript into chunks that represented distinct relevant pieces of conversation as described below, bringing intelligence to the analysis (detecting shifts in conversations) that a machine could not. These semantic units were useful for the human coding process and allowed the humans to consider the deep semantics of the text when identifying components that formed links. Next, both coders independently applied the thesaurus to the segments. Finally, components were linked if they co-occurred within the same segment. Three meetings were hand coded in this way.

After each of the three meetings was coded, the coders discussed differences in how codes were assigned and iteratively refined the codebook and thesaurus accordingly. One of the coders had personally observed the actual meetings, and had additional knowledge based on interviews with almost every member of the project. The second coder was new to the research team, and helped to develop the coding material, but had not attended any of the meetings or interviews.

In qualitative text coding, coding units, which we call segments, can span varying lengths of text data, ranging from a long sentence to multiple paragraphs [21]. In NLP, a coding unit typically entails a small number of sentences,

often one [16,45]. One of our human coders segmented the meeting transcripts into ‘decision discussions,’ i.e. chunks of text data in which a single decision regarding a technical component is discussed. Instead of identifying these segments as a particular fixed length, the coder identified them from the semantic and pragmatic context of the text data. The guidelines for segmentation were:

1. Identify design decision discussions: A design decision discussion is interpersonal conversation about one aspect of the design, a discussion of a need that implies a change in the design, or a discussion of alternatives for one particular design decision. A design decision may also involve a discussion of alternatives for the design of a system component, a testing plan related to the decision, the design of testing equipment related to the decision, or the results of tests that bear on the decision.
2. Coding asides or interruptions of design decision or test discussion: If the interruption itself is not something we code (not a new decision) and it is brief, and if the conversation following seems to be a continuation of the same discussion, then we include it in the design decision discussion.

Note that segmentation is not a core part of our method. It only applies to the hand-coding method that we used as a comparison point. We did not calculate inter- or intra-coder segmentation reliability because only one segmentation was done per transcript by a single coder. The method that we introduce in this paper avoids segmentation, and the manual choices associated with it, by using the sliding window technique which can be sensitivity-tested and tuned without as much manual effort.

We evaluated the agreement between both coders by determining the inter-coder reliability, as an initial test of the reliability of our coding scheme. In order to do that, we performed QAP on the manually generated networks. Note that QAP is a comparison on the network level, not on the entity or frequency vector level, which is typically the norm for content analysis [46]. Agreement on the network level is harder to achieve than on the entity level because not only matches and mismatches on the entity (node) level are counted, but also agreement on the edge formation level. In general, matrices are more suitable than frequency vectors for contrasting networks as they consider structure. We chose to compute agreement on the network level so we could compare the hand coding to the machine coding. For the three hand-coded meetings, the inter-coder agreement on the network level averaged 0.80. We interpreted the high level of agreement between the human coders, who had very different levels of experience with this robotics

project, as an indicator that the coding materials and procedure were reliable, robust and comprehensive. This is an important result: As with most a posteriori network studies of behavioral data, assessing the content validity of network data and respective results is hard because the systems of interest might not be accessible anymore and other sources for ground truth data might not exist [47]. Therefore, a rigorous assessment of the robustness of the coding material and procedure as presented herein is important for evaluating novel methods.

We next compared each of the hand-coded networks against the respective machine-generated networks for each meeting, meaning we calculated two QAP values for each meeting (first human to machine comparison, and second human to machine comparison). The average QAP values for the human to machine comparisons were 0.53, 0.36, and 0.77 for the first, second, and third meeting respectively; with an average of 0.55. These numbers would be unacceptably low for a comparison on the entity level, but are reasonable on the network level when compared to previous work. Agreement on manual coding of texts into networks has not often been reported in the literature. In fact, the only other study we are aware of [6] found a QAP correlation of $r = 0.25$ and $r = 0.22$ between human and machine coding for two different datasets, respectively. Both values were statistically non-significant. This approach is comparable to what we did, but our agreement is well above these values and significant (at $p < 0.05$). In the previous work, the networks built by each individual were aggregated into one cumulative hand-coded network per dataset. These two networks were again compared against the machine coded networks, resulting in QAP values of $r = 0.55$ and 0.51 , both of which were significant. In our own analysis, when we aggregated the three machine-coded meetings (by summing link weights) and compared them with the (six) aggregated human-coded networks from the same meetings, the QAP correlation of 0.74 ($p < 0.0001$) suggested a high level of agreement. Furthermore, since our average inter-coder reliability was 0.80, the agreement between machine coding and hand coding could not meaningfully exceed this value. Only one of the machine coded networks (0.77 agreement) approached this theoretical maximum.

Finally, we performed an error analysis on the agreement between human and machine coding: We ran the automated node identification on each of the 123 ‘decision discussion’ segments identified in the two of the hand-coded meetings. Given the eleven keywords in our codebook, there were ($123 \times 11 =$) 1353 possible codeword-segment pairings. 409 of these were coded by at least one of the two coders. 121 pairings were identified by the machine but not by either human (false positives). 49 pairings were identified by both humans but missed by the machine (false

negatives). For the node identification step, we argue it is preferable to have more false positives than false negatives (even though neither type of error is desirable) for two reasons: First, false positives are easier to correct for than false negatives. Second, links between false positives can be expected to have a low link weight, which may be filtered out by setting a threshold for link weight, as described below.

5.2. Comparison with Expert Interviews

We also extracted a network based on expert interviews in order to provide another point of comparison for the networks extracted using our method. In the fourth and fifth week of our project observation, we conducted 23 one-on-one interviews with 22 of the 30 project members about the nature of their work (one project member was interviewed twice). In the interviews, the participants were asked to describe what they were working on at the time, other aspects of the project that affected their work, and who else they coordinated with about that work. The interviews lasted for approximately 30 minutes and we obtained notes on the participants' responses. Nine of the interview notes were very detailed, approaching the level of transcription, and the other 14 were more abbreviated field notes. Using these 23 text files as input, for each interview we hand coded project components mentioned in the context of a dependency and used those codes to associate the two components with a link. We used our thesaurus to identify instances of terms, and then looked at the linguistic context around them to identify any components mentioned in association. We then aggregated these networks into a single symmetric network based on the interview data. In order to be able to compare this against an aggregated network of networks from the meeting transcripts, we also aggregated the networks per meeting into one network as follows:

We combined the meeting-level networks into one aggregate network by summing the link weights. Binarizing the resulting network (by converting all the non-zero link weights to one) resulted in a completely connected network. Based on the assumption that links that only occur in a few meetings are not as prevalent as links addressed in most or every meeting, we decided to control for false positives by applying a link weight threshold to the aggregated network. A low threshold may produce a fully connected network, while a high threshold may produce many isolated nodes. We argue that an intermediate threshold will retrieve the links that are most relevant given the data. In our sample dataset, we varied the threshold and measured the density of the resulting aggregate networks. This is one way to examine the frequency distribution of link weights. We chose a target density of 0.5 to eliminate spurious links and reveal the salient associations for this project. In social

networks as well as networks that have more entities than ours, a density of 0.5 would be considered quite high, but in our context it simply means that half of all technical components are related to each other. We obtained a density of 0.5 when a threshold of cumulative edge frequency was 33. In general, we recommend consulting the literature to determine reasonable and empirically based density values for association networks. After sorting links by their weight, project managers may also be able to identify a threshold value to separate the links they perceive as the most salient from those that they believe are spurious or unimportant. For our sample project and the purpose of validation (as described below), we had knowledgeable observers identify which links between components they perceived as strong (as opposed to weak or absent), building a perceived association network directly. This produced a network with a density of 0.55 (0.45 if self loops are omitted). This further supports our choice of 0.5 for a density threshold in the aggregated network.

Finally, we compared the aggregated network of the interviews with the one produced using our method (the aggregated network of the meetings). We obtained a significant ($p < 0.05$) QAP correlation of 0.42. This correlation suggests that the networks generated by machine from the 'all-hands' meeting data are significantly more similar to the (hand-coded) networks based on interview data when the interview network had its nodes labeled correctly than if the node labels were randomly rearranged. Interestingly although both data sources correspond well to the machine coded networks, the hand-coded meeting networks seem to have a stronger association to the machine coded networks than the interview networks. This may be a function of the different views these data sources represent on the underlying technical artifact structure.

5.3. Comparison with DSM-style Association Network

Three of the authors were immersed in the robotics project, regularly participating in and observing the weekly meetings, or extensively discussing the project and its dependencies. These three people manually created a description of project dependencies similar to a DSM [48,49]. This is a square matrix indicating the presence or absence of an association between key components. Because we were only measuring associations instead of dependencies, the matrix was symmetric. For each possible association, the three people indicated whether the connection was strong, weak, or absent; providing ordinal data with three values. We omitted self loops from consideration, because the fixed strong association between each component and itself would artificially inflate the correlation. We then returned to the machine coded aggregate network of meeting data, and selected two new thresholds

(replacing the previous threshold of 33). The new thresholds separated the links into categories of strong, moderate, and weak links, such that the distribution between these three categories most closely matched the distribution of strong/weak/absent ties from the manually constructed matrix. For our sample data, links with a weight less than 18 were considered weak (15/55; 27%), and links with a weight greater than 36 were considered strong (25/55; 45%). Because this was ordinal data with only a few groups relative to the number of links, we compared the networks by calculating Spearman's gamma. Spearman's gamma (correlation) between the manually constructed DSM-style network and the aggregated network extracted with our method was 0.25, and was not statistically significant ($z = 0.87$). This indicates that the DSM constructed by experts was measuring a different aspect of the association network than the networks generated from the all-hands meetings transcripts. This result suggests that a fruitful area of future research is study of the relationship between data source attributes and the nature of the corresponding association network. This research could work to identify what kinds of association-related questions different text and knowledge-based data sources are best able to answer as a function of the nature of dependencies discussed in the data.

6. APPLICATIONS OF METHOD

In this section, we consider potential applications of the output produced using our method. We first report on the evolution of association networks over time in our sample data and what this type of analysis might reveal. We next consider more generally the application of network analysis metrics to reveal properties of a project. For each set of analyses discussed, we consider research questions that these analyses might help to answer and highlight the insights that project managers, engineers and designers might gain from analyzing association networks in this way.

6.1. Dependencies and Network Structure over Time

Examining how association networks evolve over time may represent a way of characterizing the continuity of associations in an underlying technical artifact over the course of a project. This analysis is described simply as an illustration of an application of our methods output to characterize the stability of one's extracted networks, and is not intended to develop theory about network evolution. In our sample dataset, each meeting can be thought of as one particular view of the project. The association network generated by adding all networks revealed up to a certain point in time (or a particular meeting in our dataset) can reveal when new associations become important, while

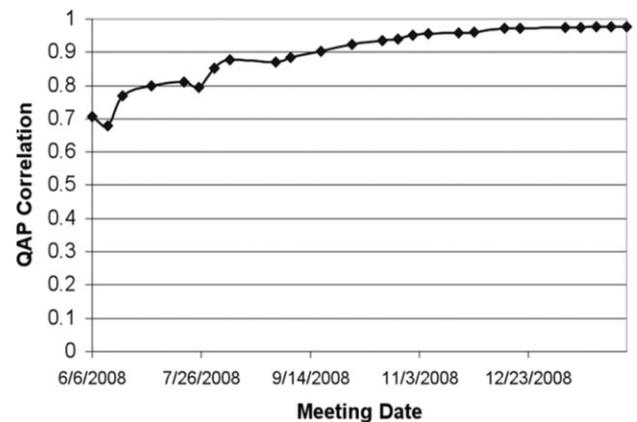


Fig. 2 Correlation between cumulative and aggregated network.

other associations drop out. If the aggregated network quickly converges to the full network that was observed over all meetings, it would suggest that most project associations were revealed early in the project's history. If, however, the networks take a long time to converge, this would suggest that the network is fairly volatile with associations changing frequently.

To make this assessment for our sample network, we examined the correlation between aggregated thresholded network across all weeks and an increasingly cumulative network of all meetings up to and including the comparison week. This comparison over time is presented in Figure 2.

Figure 2 shows that the partial network converges fairly quickly with the aggregated network. In the first meeting, the correlation is already 0.70. This value dips slightly after the second meeting, but reaches 0.80 by the fourth meeting. By the eighth meeting, 90% of all dependencies (above our threshold link weight) have been identified. Thus in our focal interdisciplinary engineering project which involved over 30 people, most technical relationships among components were brought up fairly quickly. Associations that are mentioned early can be attended to by project managers, especially when they are explicitly identified, e.g. through the method presented here.

6.2. Network Analysis of Association Networks

An association network may provide a richer way to represent relations among components in a project as compared to simpler typologies of interdependence. For example, modularity is a central idea in the product development literature (e.g. [50]). Functionality is partitioned among components (or modules) that interact in limited ways, allowing teams working on these components to function in a loosely coupled fashion. Yet modularity provides only a partial picture of the nature of dependencies across project components.

The association networks produced by the presented method can reveal details about nature of relations within projects. For example, the node-level properties of the association network may indicate the complexity of each component of the technical artifact. Examining how often a component is discussed (node weight), how many other components it is connected to (degree centrality), or how many other components may be connected through it (betweenness centrality) can have implications for how to effectively manage dependencies in a project (for a definition of these measures see ref. 51). Network measures may reveal the extent to which changes are likely to propagate broadly in a novel complex project. For example, a high clustering coefficient would suggest there is some level of task modularity, meaning changes would not be likely to propagate beyond self-contained modules. Similarly, the characteristic path length would indicate the extent to which a change in any design component can propagate quickly within the network, with shorter path lengths implying quicker propagation. The dynamics of node and network level measures can also be examined over the course of a project. Analysis of network measures over time can uncover associations that are consistently issues, as distinguished from those that may be problematic for a short time but then resolved. Future work is needed to examine and define these characteristics of association networks and their implications for project management and project performance outcomes.

6.3. Applications of Association Networks for Collaboration Support

The association networks produced with our method can also be used to enhance coordination support technologies that are employed for innovative product development and engineering projects:

6.3.1. Guide communications based on component and respective task associations

If updates on each component change were posted to all engineers in every group, it would result in a confusing flood of information. But if data on associations between components were available, they could be used to filter updates to only display those that impact a member directly or indirectly. These tailored updates might help to identify and resolve potential conflicts early on. Text-based communication tools already employed in engineering environments could be used to create special-purpose communication channels to connect engineers working on sets of strongly linked modules. Previous work has found such channels are effective for software development teams [4,52,53], and we expect association-based targeting

to enhance their utility substantially. If future work using this method shows that patterns of associations within a domain are relatively stable and consistent, even in the face of continuous design change, communication channels can be set up in advance to support collaboration where appropriate.

6.3.2. Enable dependency awareness in real time

Association networks could be applied to enhance awareness of interdependent project components, but this application is based on the assumption that component associations imply coordination requirements. If two components are mentioned in close proximity during meeting conversation data, their designs are assumed to be mutually influential. Using the congruence computation method from [4], the association network drawn from our method could be combined with a task assignment network indicating engineer responsibilities for each project component, in order to compute a network of coordination requirements. We assume that networks are useful representations of component associations. Links imply that components are involved in the same decision. We are inferring that this means the components affect each other in the context of the decision—revealing potential interdependence in the system itself.

We envision that the semi-automated method for constructing association networks could be used to intelligently drive and configure collaboration technology in real time. To realize this vision, future research needs to develop ways to integrate these association computations to drive system behavior. Collaboration systems would then be able to intelligently route information or adapt displays based on task associations detected in design discussions. Such a network could be combined with a corresponding task assignment network based on which artifacts the engineers are currently working on, to compute who needs to coordinate with whom in real time. This information could be used to automatically reconfigure communication and collaboration technology as described above, guiding communication between each person and the others whom they need to coordinate with most in order to complete their task. Relevant information could also be displayed peripherally and non-intrusively, allowing the engineers to browse it as desired or when convenient.

This application assumes that the inferred association network suggests how to filter information about changes to design components. At the same time, further research is needed to establish the potential utility of broad awareness of project activity information versus targeted updates. Given that associations may change as the design changes, a broad awareness of the design state may prepare developers and engineers for unanticipated changes to components.

7. LIMITATIONS

This work has several limitations. We believe that each limitation represents an opportunity for future work. Communication data are one among multiple indicators of component associations. Alternative types of signals can be collected from the environment and exploited to construct task association networks (e.g. [54]). Some 'signals' of component association are more accurate than others, for example explicit electronic traces from online production systems and project email communication may be more reliable than hallway chatter. Future work is needed to compare the agreement between different types of signal data used for this purpose, and to identify differences that could complement each other. It is also important to note that our method is not designed to reveal the social structure of a project. An open question is how the social structure might influence the component associations detected using our method. The social relationships among the people on a project might influence who is already aware of each other's work, the kinds of issues that are resolved prior to conversations observed and analyzed with our method (e.g. all-hands meetings in our sample data). Social relationships could have some impact on the components that get mentioned during observed conversations or the order in which components are mentioned. This could be a potential limitation of the method, but also represents an interesting area for future research on the relationship between task association networks and social structures.

In evaluating the proposed method, we analyzed one project from one domain. Therefore, the general utility of the coding choices beyond the steps of the method (Table 2) need to be tested for each new text coding project. In addition, we have not yet linked the analysis of the association networks with project outcome measures. This connection will be important in establishing the practical usefulness of this technique and in understanding how properties of association networks influence project performance. Future work should examine the extent to which congruence between the extracted association networks and the social networks representing connections among team members relates to effective performance.

Our approach is further constrained by the reliance on data being available in textual format. Given the cost and effort required to obtain transcripts from phonetic input, this represents a limitation for using verbal communication data as input to our method. As speech to text recognition technology is greatly improving, this will be less of an issue in the future.

This raises the question of to what extent transcription accuracy influences the validity of our technique. We expect the agreement between a well-formed thesaurus and text data to decline as the transcription accuracy declines. This caveat also represents a potential limitation of our

thesaurus-based approach, as statistical methods can be expected to work better on less accurately transcribed methods than the deterministic thesaurus based method we use.

An additional shortcoming with our method is that its validation is partially based on a comparison with manually segmented transcripts. The quality of the manual segmentation may influence our results, and future work is needed to validate our method with other segmentation strategies in other contexts. It is important to note that manual segmentation is not part of the method. Moreover, the manual segmentation allowed us to go beyond the classic use of sentences or paragraphs as segments, which do not necessarily represent meaningful conversation units.

Another major limitation of the presented method is that manually creating a codebook and thesaurus requires many hours of labor and expert engagement, although this is also required as an input to hand-coding methods. A data driven approach to thesaurus construction would significantly reduce this time and could be done in a domain-independent way or in consultation with subject matter expert material for a domain-driven approach. Despite the time required for thesaurus construction, once the thesaurus is built, the machine coding of association networks is fast and perfectly reliable [7].

Finally, the method we have presented is just one method for obtaining a representation of component associations on a complex project. The relative benefit of our method should be evaluated in light of other methods for obtaining association data such as system architecture diagrams, manual construction, or activity trace data. Future work should examine the relative accuracy one could expect from other methods for representing component associations, as compared with the method we have proposed.

8. FUTURE DIRECTIONS

Our technique raises several questions which represent opportunities for future research. One open question is whether the same results could have been obtained without using external expertise to construct a thesaurus, but by only using a data driven approach to building the thesaurus. Our method combines the data-driven approach of looking at the most common unigrams and bigrams with subject matter expertise to identify the salient terms and mappings of those terms to higher level concepts. Alternatively, one can use a data-driven only, n-gram based approach to identify key concepts and generate thesauri within a domain.

Another open question raised by this work is the choice of an appropriate window size, and dependence of the network structure on the window size chosen. The appropriate window size is likely to vary across domains,

datasets and genres. More research is needed to identify window size selection guidelines. As more people use our method in different settings, their results will suggest appropriate window sizes for other contexts, and whether they vary across domains.

Our proposed method constructs a network from text data. Alternatively, one could use classic content analysis methods in order to identify salient concepts [46]. Future work could explore how well both techniques agree on the entity identification level. Such a procedure would also allow one to compare frequency vectors of terms and analyze whether the correlation between those vectors complement comparisons on the network level in a meaningful way. In addition, future work may want to assess the validity of our method using network correspondence approaches other than QAP, e.g. Hamming distance.

9. CONCLUSIONS

We have introduced a novel, interdisciplinary method for extracting component associations from meeting conversation data. In our analysis we have considered how the networks extracted using this method compare with networks generated using three alternative approaches. We have shown that the networks generated the proposed method are similar to those generated by human coders from the same data, as well as networks generated from interviews with project members, but compare less closely to a DSM constructed by project experts.

Armed with this new method for identifying associations, it becomes practical to mine text data sources to generate a better understanding of the consistency or variability of associations in various domains. We analyzed one case of a lunar rover engineering project, but since we based our coding scheme on a number of sources, including a selected subset of technical papers from the field robotics domain, we expect that the coding scheme will have some generality within the domain. The resulting thesaurus is available on request, for future use and modification. We have provided guidelines on how to generate a codebook and thesaurus for other domains.

We have described how the characteristics of the association networks constructed using our method can reveal important information about real-world engineering projects. We believe this method has value both for (i) understanding the nature of collaboration around technology in various engineering and innovation domains and (ii) helping to support collaboration and coordination in particular projects.

Network analyses on the extracted data could be used to reveal crucial information about how work is connected on

a complex project, with potentially important implications for coordination among people. In summary, the method we introduce represents a contribution to the field of applied network analysis because it provides a means of capturing associations among components of dynamic projects where electronic artifacts may not be available. We encourage others to take up and improve upon the method that we have presented, to explore association networks in other domains, and incorporate techniques for capturing and representing the nature of interdependencies so that we can achieve better knowledge of supporting and executing complex projects. Supporting the interaction of multiple parties and interdependent project components remains a critical challenge in the organization of complex projects. Our work can help address that challenge. In dynamic environments, it is important that we base our understanding of collaboration and complex project management on a detailed analysis of associations to ensure effective coordination while preventing information overload. The method we have introduced provides a generally useful way to model tasks and inform the activities needed to coordinate work in innovative engineering collaborations.

REFERENCES

- [1] I. Foster, C. Kesselman, and S. Tuecke, The anatomy of the grid: enabling scalable virtual organizations, *Int J High Perform Comput Applicat* 15 (2001), 200–222.
- [2] H. W. Chesbrough and D. J. Teece, Organizing for innovation: when is virtual virtuous? *Harvard Bus Rev* 80 (2002), 127–135.
- [3] T. W. Malone and K. Crowston, The interdisciplinary theory of coordination, *ACM Comput Surv* 26 (1994), 87–119.
- [4] M. Cataldo, P. A. Wagstrom, J. D. Herbsleb, and K. M. Carley, Identification of coordination requirements: implications for the design of collaboration and awareness tools, In *Computer-Supported Cooperative Work*, Banff, Alberta, Canada, ACM Press, 2006, 353–362.
- [5] J. Diesner and K. M. Carley, Words and networks, In *Encyclopedia of Social Networking*, G. Barnett and J. G. Golson, eds. Thousand Oaks, CA, Sage, 2011.
- [6] S. R. Corman, T. Kuhn, R. D. McPhee, and K. J. Dooley, Studying complex discursive systems: centering resonance analysis of communication, *Human Commun Res* 28 (2002), 157–206.
- [7] G. King and W. Lowe, An automated information extraction tool for international conflict data with performance as good as human coders: a rare events evaluation design, *Int Org* 57 (2003), 617–642.
- [8] M. Cataldo, J. D. Herbsleb, and K. M. Carley, Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity, In *International Symposium on Empirical Software Engineering and Measurement*, Kaiserslautern, Germany, 2008.
- [9] V. Krishnan and K. T. Ulrich, Product development decisions: a review of the literature, *Manag Sci* 47 (2001), 1–21.

- [10] D. Broniatowski, A method for generating social networks from meeting transcripts, In 21st International Joint Conference on Artificial Intelligence (IJCAI), Pasadena, CA, 2009.
- [11] A. Dong, The latent semantic approach to studying design team communication, *Design Studies* 26 (2005), 445–461.
- [12] H. Bernard and G. Ryan, Text analysis: qualitative and quantitative methods, In *Handbook of Methods in Cultural Anthropology*, H. Bernard, ed. Walnut Creek, Altamire press, 1998, 595–646.
- [13] K. M. Carley, Coding choices for textual analysis: a comparison of content analysis and map analysis, *Sociol Methodol* 23 (1993), 75–126.
- [14] M. Doerfel, What constitutes semantic network analysis? a comparison of research and methodologies, *Connections* 21 (1998), 16–26.
- [15] J. A. Danowski, Network analysis of message content, *Prog Commun Sci* 12 (1993), 198–221.
- [16] D. Gerner, P. Schrod, R. Francisco, and J. Weddle, Machine coding of event data using regional and international sources, *Int Studies Quart* 38 (1994), 91–119.
- [17] J. Janas and C. Schwind, Extensional semantic networks, In *Associative Networks. Representation and Use of Knowledge by Computers*, N. V. Findler, ed. New York, San Francisco, London, Academic Press, 1979, 267–302.
- [18] A. McCallum, Information extraction: distilling structured data from unstructured text, *ACM Queue* 3 (2005), 48–57.
- [19] J. Sowa, Semantic networks, In *Encyclopedia of Artificial Intelligence*, (2nd ed.), S. C. Shapiro, ed. New York, NY, USA, Wiley and Sons, 1992, 1493–1511.
- [20] A. Lewins and C. Silver, *Using Software in Qualitative Research: A Step-by-step Guide*, London, UK, Sage, 2007.
- [21] B. Glaser and A. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*, New York, NY, Aldine, 1967.
- [22] W. Woods, What’s in a link: foundations for semantic networks, In *Representation and Understanding: Studies in Cognitive Science*, D. Bobrow and A. Collins, eds. New York, NY, Academic Press, 1975, 35–82.
- [23] I. Hendrickx, S. Kim, Z. Kozareva, and P. Nakov, Semeval-2010 task 8: multi-way classification of semantic relations between pairs of nominals, In *NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, Boulder, CO, 2009, 94–99.
- [24] C. J. Fillmore, The case for case, In *Universals in Linguistic Theory*, E. Bach and R. T. Harms, eds. New York, NY, USA, Holt, Rinehart, and Winston, 1968, 1–88.
- [25] C. W. Roberts, A generic semantic grammar for quantitative text analysis: applications to east and west berlin radio news content from 1979, *Sociol Methodol* 27 (1997), 89–129.
- [26] M. F. Porter, An algorithm for suffix stripping, *Program* 14 (1980), 130–137.
- [27] B. Krovetz, *Word Sense Disambiguation for Large Text Databases*, Department of Computer Science, Ph.D. Thesis; Amherst, University of Massachusetts, 1995.
- [28] Y. Yang and J. Pedersen, A comparative study on feature selection in text categorization, In *ICML-97, 14th International Conference on Machine Learning*, J. D. H. Fisher, ed. Nashville, TN, Morgan Kaufman Publishers, Inc., 1997, 412–420.
- [29] J. Diesner and K. M. Carley, Conditional random fields for entity extraction and ontological text coding, *J Comput Math Organ Theory* 14 (2008), 248–262.
- [30] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel, A novel use of statistical parsing to extract information from text, In 1st Conference of North American chapter of the Association for Computational Linguistics, Seattle, WA, 2000, 226–233.
- [31] G. Salton and C. Buckley, Term-weighting approaches in automatic text retrieval, *Inform Process Manag* 24 (1988), 513–523.
- [32] A. McCallum, X. Wang, and A. Corrada-Emmanuel, Topic and role discovery in social networks with experiments on Enron and academic email, *J Artif Intell Res* 30 (2007), 249–272.
- [33] K. M. Carley, D. Columbus, M. Bigrigg, and F. Kunkel, *AutoMap user’s guide*, Technical Report CMU-ISR-09-114, Carnegie Mellon, 2010.
- [34] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, A perception-driven autonomous urban vehicle, *J Field Robot* 25 (2008), 727–774.
- [35] J. Carlson and R. Murphy, How UGVs physically fail in the field, *IEEE Trans Robot* 21 (2005), 423–437.
- [36] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebel, F. von Hundelshausen, O. Pink, C. Frese, and C. Stiller, Team ANNIE WAY’s autonomous system for the 2007 DARPA urban challenge, *J Field Robot* 25 (2008), 615–639.
- [37] R. R. Murphy and S. Stover, Rescue robots for mudslides: a descriptive study of the 2005 la conchita mudslide response, *J Field Robot* 25 (2008), 3–16.
- [38] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. R. Whittaker, Z. Wolkowicki, and J. Ziegler, Autonomous driving in urban environments: boss and the urban challenge, *J Field Robot* 25 (2008), 425–466.
- [39] P. A. Schrod, Ö. Yilmaz, D. J. Gerner, and D. Hermick, Coding substate actors using the cameo (conflict and mediation event observations) actor coding framework, In *Annual Meeting of the International Studies Association*, San Francisco, CA, 2008.
- [40] D. Krackardt, QAP partialling as a test of spuriousness, *Social Networks* 9 (1987), 171–186.
- [41] L. Hubert and J. Schultz, Quadratic assignment as a general data analysis strategy, *Br J Math Stat Psychol* 29 (1976), 190–240.
- [42] K. M. Carley, J. Reminga, J. Storricks, and D. Columbus, *ORA user’s guide*, Technical Report CMU-ISR-09-115, Carnegie Mellon, 2010.
- [43] G. M. Olson, J. S. Olson, M. R. Carter, and M. Storosten, Small group design meetings: an analysis of collaboration, *Human-Computer Interact* 7 (1992), 347–374.
- [44] M. Conway, How do committees invent, *Datamation* 14 (1968), 28–31.
- [45] J. Diesner and K. M. Carley, Revealing social structure from texts: meta-matrix text analysis as a novel method for network text analysis, In *Causal Mapping for Information Systems and Technology Research: Approaches, Advances,*

- and Illustrations, V. K. Narayanan and D. J. Armstrong, eds. Harrisburg, PA, Idea Group Publishing, 2005, 81–108.
- [46] K. Krippendorff, *Content Analysis: An Introduction to its Methodology*, Thousand Oaks, CA, Sage, 2004.
- [47] R. Burt and N. Lin, Network time series from archival records, In *Sociological Methodology*, D. R. Heise, ed. San Francisco, CA, Jossey-Bass, 1977, 224–254.
- [48] T. M. Browning, Applying the design structure matrix to system decomposition and integration problems: a review and new directions, *IEEE Trans Eng Manag* 48 (2001), 292–306.
- [49] D. V. Steward, The design structure system: a method for managing the design of complex systems, *IEEE Trans Eng Manag* 28 (1981), 71–74.
- [50] C. Y. Baldwin and K. B. Clark, *Design Rules: The Power of Modularity*, 1, Cambridge, MA, The MIT Press, 2000.
- [51] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.
- [52] M. Handel and J. D. Herbsleb, What is chat doing in the workplace? In *Conference on Computer-Supported Cooperative Work*, New Orleans, LA, 2002, 1–10.
- [53] J. D. Herbsleb, D. L. Atkins, D. G. Boyer, M. Handel, and T. A. Finholt, Introducing instant messaging and chat into the workplace, In *ACM Conference on Computer-Human Interaction*, Minneapolis, MN, 2002, 171–178.
- [54] A. Pentland, *Honest Signals: How they Shape Our World*, Cambridge, MA, The MIT Press, 2008.