SOCS: Socially intelligent computing for coding of qualitative data

This proposal will combine human and computational resources—in the form of human researchers and Natural Language Processing (NLP) and Machine Learning (ML) tools—to solve problems that currently challenge either working alone. The proposal has the goal of developing and evaluating an innovative NLP and ML-based research tool to support qualitative social science research, specifically content analysis. Content analysis is a qualitative research technique for finding evidence of concepts of theoretical interest using text as raw data rather than numbers (Myers, 1997). The process of identifying and labelling significant features in text is referred to as "coding" and the result of such an analysis is a text annotated with codes for the concepts exhibited (Miles and Huberman, 1994). This technique has become increasingly popular and more applicable as the volume of available "born-digital" text has exploded. However, the reliance on manual analysis of the text limits the scale and scope of content analysis research.

In this proposal, the problem of coding qualitative data is conceptualized as an Information Extraction (IE) problem amenable to automation using NLP. However, rather than seeking to automate the process, the technologies will be used in a supporting role, creating a human-computer partnership. ML will be used to induce NLP rules from examples of coded text, avoiding the need to develop rules manually. To reduce the amount of training data needed from the human participants, an active learning process will be employed, in which a few hand-coded examples are used to create an initial model that can be further evolved through interaction with the user. These approaches will be combined in a prototype tool to support qualitative content analysis. As a demonstration and test of the tool, it will be applied to current and novel studies of cyber-infrastructure-supported distributed groups, specifically free/libre open source software development teams, and then to a broad range of social science research problems. This broad usage will also provide a test of the generalizability of a socio-computational approach to this problem.

Expected intellectual merit. The intellectual merit of the proposed research is four-fold. First, the proposal seeks to develop a novel *socio-computational system* that supports a human-computer partnership through the integration of information extraction and active learning. Second, a validation study will apply the tool to a diverse set of codes, providing *evidence of the generality and limits of a socio-computational approach*. Third, the demonstration studies using the tool will contribute to research on distributed groups. Finally, the project addresses a fundamental *methodological problem in the broad domain of qualitative research*, namely dealing with large quantities of unstructured qualitative data, by applying innovative computer-support. By avoiding the need for hand-writen rules and reducing the required amount of hand-annotated training data, this partnership will make practical the use of a system for coding large-quantities of qualitative data in various domains.

Expected broader impacts. The project has numerous broader impacts. In addition to the expected intellectual contributions described above, the proposed research will benefit society by providing useful infrastructure for research, first, in the form of a content analysis tool for social science (and other) research and second, in for the form of corpora of annotated data for use in future Natural Language Processing research. Third, the demonstration studies will provide *generalizable knowledge* to improve the effectiveness of distributed groups, an increasingly important mode of organization. Finally, the project contributes to the *education and training* (of women and minority group members in particular) through the participation of a Ph.D. student and undergraduate McNair program scholars in the research.

Key Words: qualitative content analysis; group maintenance; free/libre open source software development teams; information extraction; active learning

SOCS: Socially intelligent computing for coding of qualitative data

We propose to combine human and computational resources—in the form of human researchers and Natural Language Processing (NLP) and Machine Learning (ML) tools—to solve problems that currently challenge either working alone.

> The goal of the proposed research is to develop and test an innovative NLP and ML-based research tool that supports a computer-human partnership for qualitative social science.

The innovative contribution of this proposal is the integration of human processing with computational information extraction and active learning in a tool to support a commonly applied qualitative data analysis approach: content analysis, the extraction of structured research data from unstructured sources.

The proposed project has three phases, each lasting approximately 12 months. In the first phase (year 1), we will develop a working prototype research support system that uses NLP and active learning algorithms to partially automate qualitative data analysis. Development will draw on our current studies, examining how concepts of interest to social science researchers are linguistically realized in text to determine feasible candidates for identification using NLP and ML techniques. In the second phase (year 2), we will use the system, with previously-coded data from other projects and on a small number of existing and new research projects. The goal of this phase will be to refine the system to the point where it can be used more broadly and to provide some initial demonstration of its potential utility. We also anticipate that the research carried out during this phase will provide insights into the dynamics of distributed groups, a further intellectual contribution. In the third phase (year 3), we will make our system available to other researchers and provide support for the coding of data sets by other research groups. We will identify potential participants during phase two. We plan to select three to four groups working on different social science topics. We will be particularly interested in identifying groups that will be able and willing to share their data and the results of their coding with the wider research community in order to encourage further development of the tool.

An important result of phase three will be a study of the relationship between the effectiveness of automatic coding and characteristics of the codes themselves. We know that some codes are easier to recognize than others and know some factors that are related to coding effectiveness (e.g., ambiguity, training set size) but this will be the first comprehensive analysis based on multiple code sets developed for different applications by different research groups. The experience of using our newly developed system on a large body of data and with a diverse set of codes will provide: 1) validation of the utility of the system and the approach; 2) information about which kinds of concepts are more or less amenable to the proposed approach and thus the generalizability of our socio-computational approach; and 3) suggested enhancements to the system.

The intellectual merit of the proposed research is four-fold. First, the proposal seeks to develop a novel *socio-computational system* that supports a human-computer partnership through the integration of information extraction and active learning. Second, a validation study will apply the tool to a diverse set of codes, providing *evidence of the generality of and limits to this a socio-computational approach*. Third, the demonstration studies using the tool will contribute to research on distributed groups. Finally, the project addresses a fundamental *methodological problem in the broad domain of qualitative research*, namely dealing with large quantities of unstructured qualitative data, by applying innovative computer-support. By avoiding the need for handwritten rules and reducing the required amount of hand-annotated training data, this partnership will make practical the use of a system for coding large-quantities of qualitative data in various domains.

The remainder of this proposal is organized into four sections. In section 1, we discuss the process of qualitative data analysis and the increasingly pressing problems faced by qualitative researchers. We then discuss the promise offered by NLP and ML, and how tools might be used to support qualitative data analysis. In section 3, we present our approach to integrating human and computational elements to address this problem. In section 4, we present the study design, with details of system implementation, demonstration projects and a broader evaluation. We also include a project management plan. We conclude in section 5 by sketching the intellectual merits and expected broader impacts of our study and by reviewing results of prior NSF support.

1 The problem of qualitative research

Social science researchers often study texts such as transcripts of interpersonal communication in order to understand the practices of the populations in which they are interested[1] . However, analyzing textual data is very labour-intensive, as the text must be read and understood by a human to be analyzed. To put it bluntly, qualitative analysis does not scale—rather, it is limited by the capabilities of individual researchers.16

As a result, important research questions in the qualitative social sciences may rely on insufficient sample sizes because of the demand for intensive human effort, or worse yet, they may fail to be addressed at all. Even in the best case, dissemination of findings is delayed due to the time and effort required to analyze data. The challenge of qualitative data analysis is multiplying as organizations utilizing technology-supported collaboration (Handel and Herbsleb, 2002; Herbsleb and Mockus, 2003) generate increasing amounts of digital data, such as e-mail archives, instant messaging logs and blogs. The massive scale, diversity, and complexity of the available data sources offer enormous potential to augment traditional data sources such as interview transcripts and participant observation notes, but their volume poses significant challenges. If fully exploited, this digital data trove could make a rich contribution to the qualitative social science research, addressing behaviour in technology-supported groups, and by extension, group and organizational behaviour more generally.

Our research proposal is based on the belief that Natural Language Processing (NLP) and Machine Learning (ML) techniques can provide advanced analytic capabilities to assist qualitative social science researchers in their analyses. Such innovative tools offer the promise of extending the depth, breadth, and efficiency of qualitative data analysis and optimally utilizing researchers' intuitive and analytical skills by leveraging the large-scale processing capabilities of computers to deal with vast repositories in consistent, reproducible ways. Of course, it is unrealistic to expect such tools to automate analysis—instead, tools must be developed that work in partnership with researchers to make large volumes of data more tractable. If successful, these sophisticated NLP tools will advance the work of qualitative social science researchers by enabling researchers to explore massive amounts of data in more complex ways. In the remainder of this section, we first review the process of content analysis, which we illustrate with an extended example drawn from our prior NSF-supported research. We then discuss how NLP tools might be applied in this domain, again providing an illustration of some pilot results in our current research.

1.1 Content analysis as a qualitative analysis approach

In our proposed research, we will harness NLP and ML techniques to support the process of qualitative research, specifically, to support the process of content analysis. Content analysis is a commonly-used technique for finding evidence of concepts of interest using text as raw data (Myers, 1997). The technique is used in many fields and is increasing in popularity: for example, Neuendorf (2002) describes the approach as the fastest-growing technique in the field of mass communications. Content analysis of computer-mediated communication in particular has been an active area of research (Beißwenger, 2003; Herring, 1996).

It is commonly assumed that qualitative analysis must be interpretivist (i.e., concerned with describing individuals' understandings of their social worlds), but in fact qualitative research can adopt any research perspective: positivist, interpretivist or critical (Myers, 1997). For this study, we assume

[1] Content analysis techniques can be useful to anyone looking to extract meaning from large bodies of text, e.g., for marketers looking for insight from emailed comments on products, but for this proposal, we limit our focus on social science research, still a very broad range of applications.

that the nature of the social processes of interest (e.g., group social processes) are accurately reflected in the texts produced, (e.g., logs of email conversations), making our approach essentially positivist. This approach has advantages: it does not require the active participation of the individuals being studied, which can be difficult to obtain, nor does it rely on participants' recollections or impressions of the process. On the other hand, the understanding we develop by analyzing the process from an external (or "etic") perspective may not be the same as the understanding participants have themselves (an "emic" perspective).

The process of identifying and labelling significant features in text is referred to as "coding" and the result of such an analysis is a text annotated with codes for the concepts exhibited (Miles and Huberman, 1994). A coded text can then be subject to further analysis, such as examination of the relationship between codes or quantitative analysis of their co-occurrence. A codebook documents the coding process by describing the characteristics of the text that count as evidence for each concept of interest (as simple as particular words or phrases or as complex as types of arguments or statements). The unit of coding might be fixed in advance (e.g., a sentence), though in manual coding it is common to code "semantic units", sections of text that contain the concept of interest, from a single word to an entire utterance (Baxter, 1993, p. 244). A codebook might also include definitions or references for the concepts represented and positive and negative examples of text that is evidence for a code, although it has to be admitted that much of the knowledge that guides coding is held tacitly by the coders.

Coding can be deductive or inductive or most often, a mix. A deductive approach is based on a theoretical framework that identifies concepts of interest for the codebook. Such an approach would be appropriate when the goal of the analysis is to test a theory. An inductive approach starts with a research problem and data and induces relevant concepts from them, setting aside any preexisting concepts (Glaser and Strauss, 1967). Such an approach is appropriate when the goal is developing novel theory for some unexplored setting. A mixed mode analysis, probably the most common approach, starts with relevant concepts from theory, but allows the set of concepts and indicators of concepts to evolve through the analysis based on experiences with data. Codes are created, deleted, split or merged depending on the evolving understanding of the theory.

A key concern in developing a codebook for positivist research is its reliability, i.e., the degree to which different coders working with the same text identify the same set of codes, as measured by the degree of inter-rater agreement. If coders do not agree, then it is typical to have them discuss the coding until they reach a higher level of shared understanding of the code and to update the codebook accordingly. or to drop a code altogether if it cannot be reliably coded. As Baxter (1993) notes, the use of semantic units for coding can confound disagreements on units of coding with disagreements about content, so unitizing disagreements also must be identified and resolved.

While coding would seem to be admirably suited for parallelization through crowd-sourcing, the need for reliability limits these possibilities. First, if the codebook is being evolved during the analysis, then having a large number of coders greatly complicates the process of achieving consensus on the code definitions. Second, for all but the most intuitive codes, coders must be trained. Researchers have experimented with distributed coding using systems such as Amazon's Mechanical Turk (Alonso and Mizzaro, 2009; Conley and Tosti-Kharas, 2010; Kittur et al., 2008), but the limited degree of training possible places limits on the broad applicability of this approach.

While the process has been described above as manual, researchers have been applying computers to the problems of text analysis for decades (Krippendorff (2004) reports applications as early as 1958). Qualitative researchers often use computer tools known collectively as Computer Assisted Qualitative Data Analysis Software (CAQDAS) (Booth, 1993). Such tools keep track of the codes manually applied to particular sections of text and provide various summary reports. Existing tools do offer a certain level of automation, e.g., the ability of search for and mark particular phrases or even regular expressions. Other tools are dictionary-based, e.g., counting the frequency of use of words summarized into a variety of categories (e.g., nouns for animals, verbs for finishing or positive vs. negative emotions in the case of the General Inquirer (Gen, 2002; Stone et al., 1966)), though these general categories may or may not match the concepts of interest to the researcher. However, while useful, CAQDAS tools overall have not reached the level of sophistication and automation of quantitative research tools (Carey et al., 1998; Morison and Moir, 1998; Welsh, 2002).

2 Results from prior funding: Content analysis of group maintenance behaviours in FLOSS teams

To make the discussion of qualitative content analysis coding more concrete, we present an example drawn from a study by the PIs, (supported by NSF Grant HSD 05–27457, *Investigating the Dynamics of Free/Libre Open Source Software (FLOSS) Development Teams*, with R. Heckman and E. Liddy). The overall goal of the project was to examine the evolution of effective work practices in a particular kind of distributed team, namely teams of free/libre open source software developers (Crowston et al., 2005b). The example study was an examination of group maintenance behaviours in online groups, that is, behaviours that serve to keep the group together and functioning rather than directly contributing to the task output Ridley (1996). Given the focus of the proposal, we present only the methodological issues in this study and will not discuss the substantive research question further.

The qualitative data we used for this research was typical of research on computer-supported groups, namely the email and discussion forum conversations among free/libre open source software (FLOSS) developers. For this study, we randomly selected 1,469 messages from the developer discussion forums for two FLOSS projects. A random sample of messages was used because the available human coder time was not sufficient to code the entire archive, the problem we hoped to address by using NLP. Two PhD students trained to code according to a coding scheme derived from the literature. Table 1 shows the particular constructs explored. An iterative process of coding, inspection, discussion and revision was carried out to inductively learn how the indicators of the relevant concepts evidenced themselves in the data, until the coders reached a solid coding scheme. Training continued until the coders reached an inter-rater reliability of 0.80, a typical level expected for human coding. The human coded data were used as the “gold standard” to train and to assess the performance of the NLP coding. These data consisted of the selected messages with short phrases identified and coded that expressed the various theoretical constructs; examples of texts and codes are shown in Table 2.

Because the process of coding involves careful reading of texts to find instances of the phenomena of interest, the analysis process was extremely labour-intensive. Coding for our study required nearly 1 person year of effort (2 coders working half-time for a year), with additional support from other researchers for conceptual development. Part of this time was spent developing and refining the codebook, but much of it was spent reading and rereading messages, coding the phenomena of interest

Table 1: Code book for group maintenance behaviours.

Indicator	Definition
Emoticons	Emphasis using emoticons
Capitalization	Emphasis using capitalization
Punctuation	Emphasis using punctuation
Slang	Use of colloquialisms or slang beyond group-specific jargon
Inclusive pronouns	Incorporating writer and recipient(s)
Complimenting	Complimenting others or message content
Agreement	Showing agreement
Apologies	Apologizing for one’s mistakes
Encouraging participation	Encouraging members of the group to participate
Show appreciation	Showing appreciation for another person’s actions
Hedges/Hesitation	Tactics to diminish force of act; hesitation in disagreement

Table 2: Examples of coded textual data.

Example text	Code
Hmmm.... the “real” one should be at /fire/*.lproj /MailControllerWindow.ni	Hedges/Hesitation
when ur typin in an im u always spell things howevere is da shortest way.	Slang
u guys are great	Complimenting

based on the codebook. The resulting dataset is suggestive, but the small number of projects (only 2) does not support firm conclusions about the study hypotheses. On the other hand, to increase the number of projects to 60 (a sufficient sample size for statistical analysis to have the power needed to draw conclusions) would require a prohibitive amount of labour, even with the developed codebook. This situation—a high input of labour for a small payoff of data, and thus limits on the kind of research question that can be addressed—is the problem we address in this proposal.

2.1 Automated supported for qualitative content analysis coding

To support qualitative analysis and address the problem identified above, we plan to apply Natural Language Process (NLP) and Machine Learning (ML) technologies. In this section, we report on some initial experiments with this approach that illustrate its promise and potential problems, deferring detailed discussion of our planned work to the next section. (NB. These reported experiments represent initial trials rather than final decisions about the approach we plan to take.)

We conceptualize the problem of coding qualitative data against a researcher's codebook as comparable to an Information Extraction (IE) problem. IE is a subfield of NLP whose function is to extract or annotate desired parts of unstructured text. For example, Cole et al. developed a system to extract important terms from email messages in order to organize the emails for easy future retrieval (Cole and Eklund, 1999; Cole and Stumme, 2000). In our application, we use IE to extract and label text representing theoretical concepts of interest. This approach should be suitable for coding any concept that is reflected in regularities in the use of language, as in the qualitative analysis of communications archives for many research questions.

Information Extraction systems are of two types, rule-based and statistical-learning-based. A rule-based Information Extraction system relies on an expert to write topic and domain-dependent rules to capture an intended schema (i.e., a codebook). As part of our work on the HSD grant, we have experimented with rule-based IE for qualitative data coding. Rules for identifying examples of group maintenance behaviours in email messages were developed iteratively by a trained NLP analyst working with the human coders. Some rules, such as for Capitalization, were primarily based on regular expressions to detect upper case. Other rules, such as for Apology, focused on specific lexical items—‘sorry’, ‘apologies’—or a lexicon of items. But others, such as the rule for Agreement, required the use of the full range of features such as part of speech, token string and syntax. It is worth noting that email messages often exhibit numerous features that challenge traditional NLP algorithms, such as grammar and spelling mistakes, slang, embedded source code fragments and emoticon usage.

Rule-writing was interspersed with testing to assess performance on the training data during the development process. To evaluate performance, the system's output was compared to the human-applied codes, which were assumed to be correct. A portion of the human-coded data (155 messages, or about 10%) was reserved for testing of the completed ruleset. The remainder was used to assess the performance of the ruleset as it was being built. The first two columns of Table 3 reports performance of the final human-developed NLP rule set on the testing subset of messages. We report the proportion of sentences coded by the human coder and extracted by the system, the traditional information extraction metrics of recall and precision, and where recall measures the proportion of manually coded statements that were correctly extracted by the system and precision measures the proportion of extracted sentences that matched those manually coded.

As noted above, rather than seeking to automate the coding process, we planned to employ the NLP technologies in a partnership role. Specifically, we planned to have a human coder review and correct candidate codes extracted by the system before the data are used for further analysis. The rule development therefore had the goal of achieving high recall (i.e., extracting as many examples of the concepts in the data as possible) at the expense of precision (i.e., ensuring that the extracted example are all correct), under the assumption that coders can more easily discard incorrectly coded segments than they can search the entire corpus of email to find group maintenance behaviours not identified by the system. The results in Table 3 show that the human-developed NLP rules were able to achieve reasonable levels of recall for most codes and acceptable levels of precision for some (the surprisingly low performance for Capitalization and Punctuation was due to the inclusion of source code snippets in some messages that were coded by the NLP system but ignored by the human coders).

To summarize, our rule-based NLP coding experiment suggests the promise of the general approach to be explored further in this proposal. Even with the currently-achieved level of precision, identifying the 1% of sentences in the entire corpus likely to contain evidence of the phenomenon of interest (i.e., a code) could increase the human coders' productivity by two orders of magnitude, enabling us to code hundreds of projects, as desired. However, it also reveals the major drawback, namely, the necessary skill and effort to develop rules. Adopting this approach would replace the current qualitative analysis bottleneck with an even more serious NLP-analyst bottleneck. We anticipate using rule-based coding for some fixed aspects of messages (e.g., coding message time, subject, sender, receiver) but coding more varied theoretical constructs would require significant (and usually unavailable) development effort, making it infeasible in general.

2.2 Machine learning experiment

To address the bottleneck of needing trained analysts to write NLP rules, we briefly explored the use of machine-learning (ML) algorithms to automatically learn the complex patterns underlying the extraction decisions based on the statistical and semantic features identified in the textual data (Crowston et al., 2010). Using ML to infer rules can be more cost-effective than the rule-based approached as it derives rules from human-coded data and does not require the time of an NLP expert to write the rules (which is not to say that expertise is not required at all).

Table 3: Experimental results—NLP approaches compared to human coded data.

Code	**Hand-written rule performance**		**Machine learning rule performance**		**Training set size**
	Precision	Recall	Precision	Recall	
Apologies	0.67	0.67	0.50	1.00	5
Capitalization	0.19	0.60	0	0	20
Complimenting	0.40	0.67	0	0	36
Appreciation	0.45	0.64	0.50	0.60	60
Agreement	0.60	0.80	0.73	0.31	104
Salutations	0.86	0.86	0.87	0.52	105
Emoticon	0.81	0.91	0.38	0.53	144
Inclusive Pronouns	0.58	0.98	0.92	0.90	240
Punctuation	0.22	0.71	0.63	0.45	268
Slang	0.69	0.67	0.50	0.07	384
Hedges/ Hesitation	0.69	0.74	0.58	0.48	1276

Precision = proportion of sentences coded by the rules that match the human coded data.
Recall = proposition of sentences coded by the human coders that the rules coded.

We have carried out a very preliminary experiment to test the potential of an ML approach (again, this experiment represents just a proof of concept rather than final implementation decisions). As with the previous experiments, a portion (75%) of the human-coded data was used for training and the remainder for testing. The training data were used to train a classifier using a ML algorithm that inferred rules for extraction using features of the messages. The ML algorithm used in this experiment was Winnow (Littlestone, 1988), a linear classifier that works by updating the weights assigned to the different features. (In the proposed study, we plan to explore the use of other algorithms.) We chose Winnow for this experiment because it had been successfully used for information extraction problems (e.g., by Zhang et al. (2002)), and is known to be an effective algorithm in the presence of irrelevant attributes (Dhagat and Hellerstein, 1994; Littlestone, 1988), which we expected given the nature of message data.

The performance of the ML depends on correct selection of the features in the text that should be used for the rules to be learned. However, grammar, spelling and capitalization mistakes and frequent use of domain-specific proper nouns make it hard to create a usable feature space for email messages. In these initial experiments, we explored only a few simple sets of features: the words themselves, the location of a word, for example, one word or two words before or after the target coding result, part-of-speech and capitalization. A [-3, 3] text window (all six tokens) around the target coding result was used to define the feature space. (Again, for the proposed study, we plan to explore a much broader range of features.)

Results of our preliminary experiments are shown in the second two columns of Table 3. Even with the very simple feature space, the performance of the ML rules is surprisingly good for some codes, suggesting that this approach may have potential. However, these results also demonstrate that performance of the ML approach is highly dependent on having a large number of training examples from which to learn. Although coding data by hand is easier than developing specific rules for an IE system, a considerable amount of coded data is required before learning can start (Finn and Kushmerick, 2006): obtaining a large corpus of coded data for each domain is the main bottleneck of using learning-based IE systems. Unfortunately, we have only a few examples for many of the codes in this study. As a result, ML alone would thus seem to provide only a partial solution to the problem of coding social science data.

3 The proposed approach: Active learning for NLP coding

In the proposed study, we will explore a third alternative to manual and completely automated NLP rule development, which is to apply a socio-computational approach. Specifically, we will couple limited human coding of data with a machine-learning approach to develop NLP rules. Figure 1 shows the overall architecture of our proposed approach. Specifically, we will employ *sequence tagging* and *classification* algorithms with *active learning*, an approach that has been successfully used for different IE applications (Bunescua et al., 2005; Chidlovskii et al., 2006; Shen et al., 2004). In this section, we present an overview of the proposed approach, deferring discussion of the specific planned implementation to the following section.

In *sequence tagging*, the information extraction task is performed by assigning a semantic label to each token that identifies whether it is a valid start of an instance, the end of an instance or a continuation within an instance. The classifier also labels the instance with the appropriate code. To do the *classification*, we will use statistical learning techniques rather than rule learning because of their robustness over different document structures and classification tasks. Conditional Random Fields and Maximum Entropy are two well known algorithms; Support Vector Machines (SVM) are also being used (Zelenko et al., 2003). Part of the proposed research is to evaluate the performance of these algorithms for this problem.

Coding in qualitative research will likely encompass complicated structures that may not be possible to capture with regular expressions. We therefore propose to apply classification over an *NLP-augmented feature set*. The open-ended nature of the structure of the concept phrases and the unpre-

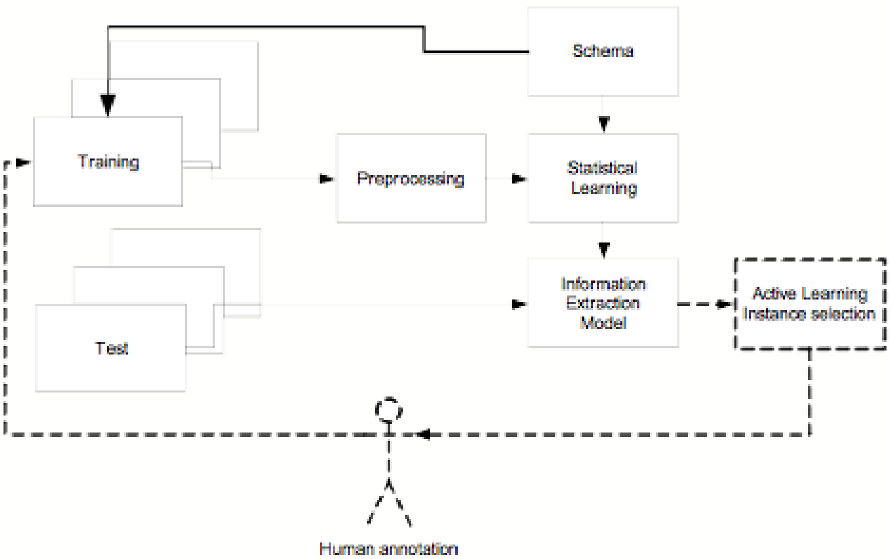

Figure 1: Generic active learning model for information extraction.

dictability of language in email leads us to start with simple sequential clues, such as capitalization of the token and whether surrounding words are in a particular lexical class. However, from earlier studies on classification we know that using additional semantic clues such as part-of-speech tags or categories of named entities help in the categorization task. An important part of the development of the proposed system will be experiments with different feature sets.

To reduce the required size of the training set, we will apply *active learning*, which is an expansion of the supervised learning process (Cohn et al., 1994; Finn and Kushmerick, 2006; Scheffer, 2002; Thompson et al., 1999). In the active learning process, a few hand-coded examples are used to create a model and the model is then run over the test documents. During that process, the system may ask the user to annotate more data, e.g., the instances that it is least certain about. The user can also choose to correct other annotations or create new ones. Newly annotated data are fed back into the training set and a new model is created. As a result of this focused coding, the system performance improves quickly with fewer training examples. Active learning continues until the user is satisfied with the output or a certain predefined performance measure is reached. Wu and Pottenger (2005) similarly used active learning to learn rules for reduced regular expressions to extract patterns of information from unstructured text (i.e., police reports). The result of the process is a trained classifier that can be used to code large corpora of texts for further analysis.

4 Study design

The basic concept proposed above is straightforward, but there are many questions to be answered to create a useable socio-computational system for this problem, including:

1. What text features are useful in recognizing social science concepts in text?
2. What is the best way to learn the patterns of features that represent concepts (i.e., codes)?
3. What kind of feedback can users provide that will improve system performance?
4. What is the best interface for obtaining such feedback?
5. What are the tradeoffs in seeking feedback from many naive users vs. a few trained ones?
6. What kinds of concepts exhibit sufficient regularity to be automatically recognized?
7. Can a partnership between human and system perform better for qualitative data analysis than either could alone?

In this section, we discuss the design of the overall project, including algorithm and software design, the design of the initial demonstration studies (basic research strategy, concepts to be examined, sample populations and proposed data collection and analysis techniques), and our plans for validating the tool set in other research settings.

4.1 Phase one: System implementation

In the first phase of the project, we will address questions 1–5 above, as we work on refining algorithms for information extraction, developing software for the active learning of rules and creation of a prototype coding system. We anticipate completing an initial system prototype during year 1, but implementation of and experimentation with some of the features discussed might overlap the following phases.

Feature extraction. The first step will involve delineating the predictable linguistic features on which algorithms to detect the research-relevant concepts can be based. We will provide these additional clues from the NLP processing by applying Syracuse University's TextTagger. Note that other NLP systems could be used for many of the processing tasks (e.g., GATE (Cunningham et al., 2002)), but we are currently planning to use TextTagger because we are familiar with it and the technology is mature; TextTagger has been used in more than 35 projects internally and with other users. TextTagger will be used to identify features including sentence boundaries; part-of-speech tags, including person and tense; stems and lemmatized words; various types of phrases; categorization for named entities and most common nouns; event detection; and co-references. The categorization capability of TextTagger provides an easy way to incorporate semantic classes of words and phrases and lexical domain knowledge that can be provided by a variety of sources. We plan to explore the use of subject-specific thesauri, subject-specific ontologies and general world knowledge and pragmatics. Finally, there are some potentially useful textual features that could easily be collected in TextTagger if needed to improve the performance of the classification, including counts of adjectives, adverbs, modals, intensifiers, etc.; sentence length and complexity; co-occurrence and collocation of terms; and other discourse level clues. We will analyze the text of codes found by the human coders to identify classes of features useful for the system and will run experiments with different feature sets. The sequential and NLP clues from the processed text will be represented as vectors in the feature space, and developers will be able to view and edit the feature space before the system moves into the learning phase.

Machine learning. For the learning phase, we will utilize algorithms implemented in the open source projects Mallet[2] and MinorThird[3]. These projects provide open source Java implementations for CRF and Maximum Entropy algorithms. As an overall framework to run our experiments we are currently planning to use MLToolkit, a machine learning and experimentation framework developed by Dr. Yilmazel in his doctoral thesis (Yilmazel, 2006, 2008). (Dr. Yilmazel has agreed to serve as a consultant for our project.) MLToolkit currently includes Support Vector Machines, Decision Trees and Naïve Bayes learning algorithms, as well as several statistical feature selection algorithms. MLToolkit will need to be extended for this project to add Maximum Entropy and Conditional Random Fields learning algorithms. The experiment management framework in MLToolkit implements various supervised learning experimental designs, such as multi-label categorization, n-fold cross validations and hierarchical categorization. The advantage of using the MLToolkit framework and expanding its capabilities as needed is that it already incorporates NLP processing for the creation of features for text.

Active learning strategy. Active learning can use many different techniques to select specific documents or instances for manual annotation; for example, Finn and Kushmerick (2003); Jones et al. (2003); Muslea et al. (2006) discuss uses of active learning in Information Extraction. While the active learning in the latter two works use contention points from multiple views for selection, the data available for our project is not known to have the independent sets of attributes required by this ap-

[2] http://mallet.cs.umass.edu

[3] http://minorthird.sourceforge.net

proach. Therefore, we focus on Finn and Kushmerick (2003), whose study proposed six document selection strategies, of which three seem applicable to our information extraction problems: *Compare*, *ExtractCompare* and *Bagging*. *Compare* compares the given document to the training set and chooses the document that is least similar to the training set. *ExtractCompare* applies the learned extractions over the document and compares the results of the extraction to the training set and selects the document whose extractions are least similar to the training set. *Bagging* divides the training set into different parts and builds models using these parts. Information extraction is done over documents by using these distinct models and documents with the least agreement from different extraction systems are selected for further annotation. To implement these active learning strategies in our system, we will extend the application programming interface (API) of MLToolkit to include the document selection strategies.

Human feedback. Application of active learning requires access to human coders who can provide feedback to the system. As in our earlier work, we will initially rely on coders (e.g., graduate students) who have been trained in the use of the codebook to produce text annotations for training, active learning feedback and evaluation. The quality of trained coders' decisions is in general very high (high inter-rater reliability) but they represent an expensive and scarce resource. To reduce the cost of obtaining human-generated coding decisions, we also plan to explore the use of crowd-sourcing techniques. Crowd-sourcing has been used as a low cost method for gathering human judgments, although the quality of judgments is not consistent (Snow et al., 2008). However, in other work at Syracuse, we developed software to use Amazon Turk to gather relevance judgments to support traditional information retrieval experiments, with good results. For example, a student working with the co-PI used Amazon Turk to gather judgments about time-varying topical interests to support query reranking (Liu, 2010). The coding decisions required for the proposed project are more complex than the simple document/query relevance judgments in these experiments additional training and use of multiple coders might produce acceptable quality at reduced cost. A small amount of funding is requested to support these experiments. We will also evaluate the use of crowd-sourcing games to obtain coding decisions.

User interface. Finally, we will integrate the information extraction and active learning algorithms with a user interface in a working prototype system. We plan to implement the system as a Web-based system for ease of use, using AJAX (Asynchronous JavaScript and XML) to make the user interface for annotation as smooth as possible. Funding is requested for professional programmer support to enable us to create a functioning tool that can be used both for our own work (e.g., on the planned study) and by other researchers.

The system will provide facilities for basic preprocessing of input data, such as conversion from common file formats to text. As well, as a demonstration of the way such a system could be incorporated in scientific cyber-infrastructure, we will provide the capability to retrieve interaction data directly from existing FLOSS data repositories, such as the FLOSSMole project[4], developed by PI Crowston and others as part of prior funded research and currently being extended with support from NSF CNS Grant 07-08437 (with M. Conklin). We will also explore the possibility of importing data from other repositories (e.g., email messages from Gmane[5]). The system will display imported documents in their original format as well as the annotated version. The initial set of codes can be applied interactively using the system or by importing annotated data from a CAQDAS tool such as Atlas-ti (via its XML export feature) or the open source system TAMS Analyzer[6]. Importing coded data will also enable calculation of inter-rater reliability, comparing codes from two human coders or between a human and the machine coding.

After the initial codes are applied, the system will use the coded data to infer an initial model and use the model to code additional documents. The system will ask the user for feedback on the accuracy

[4] http://flossmole.org/

[5] http://www.gmane.org/

[6] http://tamsys.sourceforge.net/

of the coding during the process and use the newly coded data to refine the model. Once a satisfactory model is obtained, remaining data will be coded in bulk. The resulting coded data can then be further edited for accuracy via the Web interface or in a CAQDAS tool by a human coder and finally exported for analysis, e.g., as an input to an analysis workflow or to be stored back into a data repository for further use. The integration of the coding system with other pieces of scientific cyber-infrastructure will facilitate the use of mixed data analysis, incorporating both quantitative and qualitative data.

4.2 Phase two: System use and refinement

In phase II of the proposed study, we will use the prototype systems on two demonstration studies. In this section we will present the design of these studies. The goal of these studies is to provide a testbed for experimentation with and evolution and evaluation of the proposed tool. First, we plan to continue the research described above on group maintenance behaviours in distributed teams. We will use the already-developed codebooks and human-coded data as a basis for system development starting in phase I and will use the newly developed system to complete the study in phase II. Second, during phase II, we will carry out a new study to further test and extend the usefulness of the tool. Our current plan to study leadership behaviours in distributed groups, drawing again on our prior work (Heckman et al., 2007b). A recent review of leadership theory (Avolio et al., 2009) suggested research on virtual leadership as a future “growth area” for leadership research, as the nature of leadership in virtual teams seems likely to differ from conventional teams, making it an interesting topic for further study. In parallel with our use of the tool for these demonstration projects, we anticipate continued tool refinement and experimentation with different algorithms. (Conversely, we may begin work on these projects earlier, depending on the progress on system implementation.) In the remainder of this section, we will describe the example studies, covering in turn sample selection, data collection and cleaning and data analysis.

Sample. We will start each phase by identifying promising distributed groups for study. Because of our prior experience in the area, we plan to focus our analysis, at least initially, on FLOSS software development groups (we also have access to interaction data from other kinds of cyber-infrastructure supported collaborations, which can be analyzed time permitting). There are thousands of FLOSS projects, spanning a wide range of applications. Due to their size, success and influence, the Linux operating system and the Apache Web Server and related projects are the most well known, but hundreds of others are in widespread use, including projects on Internet infrastructure (e.g., sendmail, bind), user applications (e.g., Mozilla, OpenOffice) and programming languages (e.g., Perl, Python, gcc) and even enterprise systems (e.g., eGroupware, Compiere, openCRX).

During the first stage of each study, we will analyze a small number of projects (on the order of six). In the second stage of each study, the size of the sample will be limited by the available data and processing power (computer and human). In choosing these groups, we will apply the previously developed effectiveness assessments (described above) as a theoretical sampling filter to ensure that we have groups of different types with varying degrees of effectiveness. We will also take into consideration pragmatic considerations, such as only selecting projects where we have access to the needed data. Finally, we will choose projects that produce comparable systems in order to control for the nature of the program, thus allowing a more direct comparison of the groups’ effectiveness, but in a range of categories, thus permitting theoretical comparisons across categories. For example, in the HSD grant described above, we compared Internet Messaging (IM) client projects and enterprise software projects (Heckman et al., 2006).

Data collection and cleaning. To explore a range of concepts, we will collect and analyze a range of data. The most voluminous source of data will be collected from archives of computer-mediated communications (CMC) tools used to support the groups’ interactions (Herbsleb and Moitra, 2001; Lee and Cole, 2003). These data are useful because they are unobtrusive measures of the groups’ behaviours (Webb and Weick, 1979). In particular, mailing list archives will be a primary source of interaction data, as email is one of the primary tools used to support group communication (Lanzara and Morner, 2004). In the FLOSS setting, such archives are the primary mode of communication and so contain a huge amount of data (e.g., the Linux kernel list receives 5000–7000 messages per month; the Apache httpd list receives an average of 40 messages a day).

While the raw data are already available, significant effort is needed to extract scientifically useful information from them. The initial processing to prepare the data for analysis will be to download the data from the message archives, clean the data (e.g., by removing unnecessary attachments or quotations), provide descriptive metadata on each archive and extract the elements such as the date, sender and any individual recipient names, the sender of the original message, in the case of a response, and text of each message. In this preparatory stage, we will record available demographic data such as gender, region, organization and role within the group.

Data analysis. While voluminous, the raw data described above are at a low level of abstraction. The processed data will be analyzed using the proposed new tool to raise the level of conceptualization to fit our theoretical perspective and thus answer our research questions. For the group maintenance study, we will start with the already coded data. For the new study, we will initially use CAQDAS tools for content analysis to develop an initial training dataset, followed by interaction with the system to code additional data. Data will be content analyzed following the process suggested by Miles and Huberman (1994), iterating between data collection, data reduction (coding), data display and drawing and verifying conclusions. Part of this work will establish how best to use the tool during the development of a codebook. From the data coded by the system, we plan to examine the relationship between different aspects of the group process that are exhibited in their texts. We will also develop hypotheses about the relationship between group behaviour and group performance across various settings, based on a developing understanding of the group processes. For example, Scialdone et al. (2009) noted a possible relationship between the use of inclusive pronouns by peripheral group members and project success. Though this simple relation could be tested with current CAQDAS tools (with a lot of manual work), the proposed system would allow analysis of a large number of projects to test this and other such hypotheses.

4.3 Phase three: External use and validation

The research up to this point will have been conducted by the PIs at Syracuse University. The aim of the final project phase is to test the effectiveness of the system when used by other research groups with much different interests, data sets and coding needs. There are two main objectives for this phase. First, we are interested in evaluating how well or poorly the coding system works when used by groups other than the developers. This evaluation will allow us to gauge whether the overall NLP and ML approach allows significant reductions in coding costs or significant improvements in the volume of material that can be coded. This evaluation should also suggest areas in which the coding system could be improved.

Second, we will study the relationship between the codes themselves and coding effectiveness. Earlier work has shown that there are considerable differences in the effectiveness of NLP for different codes. Our preliminary results in Table 3, for example, show that codes with a wide range of surface expression (e.g., politeness) are more difficult to recognize than codes with a limited range of surface expression (e.g., use of emoticons). Similarly, codes for which there are relatively large training sets generally produce better recognizers and higher effectiveness. Our work to date, though, has been based on a limited number of code sets produced by one research group for use with e-mail. The coding results produced in phase three will allow us to undertake a large scale comparison of effectiveness across multiple code sets and social/computational applications.

During phase two we will identify other research groups that have an interest in coding large data sets and who could benefit from automated coding support. For phase three, we will select three to four groups and provide the support needed for them to install and use the tool set for their application. To minimize support costs we will schedule training time with the selected groups at professional conferences that we already attend (funding is requested for travel to conferences for dissemination and outreach). We are particularly interested in groups that represent diverse applications, code sets, and data sets. We are also particularly interested in groups that are able to make their data and code sets available to other researchers. These test corpora will allow us to undertake the large scale comparison of coding effectiveness, will allow direct comparisons between different coding sets on the same data, and will provide a baseline that can be used to evaluate improved automatic coding tools.

4.4 System evaluation

A key aspect of the project will be evaluation of the performance of the NLP and ML algorithms for the task. We anticipate carrying out evaluations at multiple stages in the projects, gradually increasing their scope. The performance of the NLP and ML algorithms will be evaluated initially by comparing their output to human coding in order to determine precision and recall, as in the previous example. The initial training data for this purpose will be the data coded as part of the initial studies reviewed above, augmented with additional data coded in year 1. Because the group maintenance codebook includes a wide range of types of code, these tests will provide a good initial test of the generality and limits of the proposed approach; such testing will be extended to a second demonstration study in year 2 and further extended in year 3. Year 3 will also include a systematic evaluation of the performance of the proposed approach across a range of codes. Finally, the key evaluation will be how well the system as a whole does at supporting human coders and thus speeding the process of qualitative data analysis. This evaluation will be carried out at the end of the project by examining the coding done using the tool and assessing measures such as the speed and volume of coding, the precision of the coding and thus the amount of rework needed and the general capability to support the domain of research.

4.5 Management plan

Just as the proposed system is a partnership between a human coder and a system, the project will be carried out by a partnership between social and computer scientists. Based on a preliminary assessment of the effort required, we are requesting funding for an interdisciplinary team comprising two PIs, one in information science and one in the domain of information systems and organizational behaviour, a consultant in the area of machine learning, one Ph.D. student, and a professional programmer. In addition, we plan to recruit two undergraduate students from the Ronald E. McNair Post-Baccalaureate Achievement Program[7]. As well, our School has a program that employs masters students to work on research projects; these students could be used in addition (or in place, if our recruitment efforts do not pan out).

The PI, Kevin Crowston, will work during the summer on project management and research design, and devote effort during the academic year to project management and oversight. The co-PI, Nancy McCracken, has an academic year research appointment and will therefore be funded for work during the academic year on the project. Both PIs will share in project selection, overall project design and report writing. Each PI will be responsible for designing specific aspects of the project and overseeing those aspects:

- Dr. Crowston will direct the project and be responsible for project oversight and reporting and will lead the substantive research on the FLOSS groups.
- Dr. McCracken will lead the computer/information science research team in NLP tool development and integration.

To assist with the development of the NLP and ML algorithms and software, we are requesting funding for a consultant, Dr. Ozgur Yilmazel, who developed the ML-toolkit that we are planning to use. To oversee the integration of these pieces into a functional data-coding tool we are requesting

[7] The McNair program is a federally funded TRIO program, designed to prepare students for graduate education leading to doctoral studies. Undergraduate students eligible for the program must either be a potential first generation graduate student or be a member of a group underrepresented in graduate education. Students in the program attend a summer seminar in Research Methodology and Academic Literacy and are provided assistance in preparing for the GREs. During the academic year, students work on research projects with a faculty mentor.

funding for a professional programmer, at a higher level in years 1 and 2 where we anticipate doing the bulk of the development, and at a lower level in the final year. The Ph.D. student will support the principal investigators in using the tool throughout the project: working on sample selection, definition of constructs and variables, and data collection and analysis, under the oversight of the PIs. Finally, the PIs will act as faculty research mentors for two McNair undergraduate students. The exact tasks to be assigned to these students will depend on their interests and capabilities, but we anticipate having them work on data collection and analysis (e.g., coding of data). While student stipends are provided by the McNair program, travel funds are requested for these students to attend one conference per year, with the goal of presenting a research poster.

We will employ two main *project management techniques*. First, we will have regular meetings of the project members to share findings and to plan the work. Initially, these will be every other week, but the frequency of meetings will be adjusted depending on our experience and the pace of the work being carried out at the time. These formal meetings of all project participants will augment the regular interaction of the teams of PIs and students working on the data analysis and expected frequent interactions of the students as they analyze data from the same projects. The NLP development team will meet semi-weekly during the design phases and then weekly during implementation. The experience of this team on the existing toolset bodes well for an accelerated process of iterative requirements, implementation, usage and new requirements. A more formal review meeting will be held at least quarterly to assess progress and to make plans for the next quarter. Second, an initial project activity will be the development of a detailed timeline against which progress will be measured. The budget includes support during summer and academic year to support these activities.

5 Conclusion

In this proposal, we discussed the challenges of qualitative data analysis and the possibility of using innovative NLP and ML techniques in partnership with human coders to address it. These techniques will be deployed in a prototype data analysis tool that uses active learning to support a partnership between human and computer tools. As a testbed and to demonstrate the utility of the proposed tool, we plan to use it to investigate group functions within distributed groups. The proposed project will have significant intellectual merits and broader impacts.

5.1 Expected intellectual merits

The intellectual merit of the proposed research is four-fold. First, the proposal seeks to develop a novel *socio-computational system* that supports a human-computer partnership through the integration of information extraction and active learning. Second, a validation study will apply the tool to a diverse set of codes, providing *evidence of the generality and limits of a socio-computational approach*. Third, the demonstration studies using the tool will contribute to research on distributed groups. Finally, the project addresses a fundamental *methodological problem in the broad domain of qualitative research*, namely dealing with large quantities of unstructured qualitative data, by applying innovative computer-support. By avoiding the need for hand-written rules and reducing the required amount of hand-annotated training data, this partnership will make practical the use of a system for coding large-quantities of qualitative data in various domains.

5.2 Expected broader impacts

The proposed project will have numerous broader impacts in addition to the expected intellectual contributions described above. The proposed research will benefit society by providing a useful tool for qualitative science research, namely a socio-computational content analysis system, thus contributing to the *infrastructure of science*. While aimed at researchers, the tool could potentially be broadly useful to anyone seeking to mine large corpora of unstructured text (e.g., emailed comments on a product or service). In addition to the system itself, we plan to make available several test corpora for use by the research community. Each corpus will consist of a set of raw data, the coded data, the code definitions and the coding rules generated by the system. In other aspects of NLP text processing, where annotated data sets have been made available, important strides in NLP techniques have been achieved, and thus we feel that providing open access to such data sets will be a contribution to the research community.

A side-benefit of the planned demonstrations studies will be to provide *generalizable knowledge* to improve the effectiveness of distributed groups, a further benefit to society. Such groups are an increasingly important approach to needs such as software development, scientific research and policy development making it important to better understand their functioning.

To ensure that our study has a significant impact, we plan to *broadly disseminate* results through journal publications, conferences, workshops and on our Web pages. We also plan to disseminate the two contributions to the *shared infrastructure for research and education*, namely the software tool and the annotated data. Both the software tool and the annotated data will be made available on web sites maintained by the Center for Natural Language Processing at Syracuse University, through an institutional repository or through disciplinary repositories such as the UCI Machine Learning Repository[8], thus providing a long-term and maintainable platform for dissemination.

Finally, the project will *promote teaching, training, and learning* by students involved in the research project, providing them the opportunity to develop skills in model development, theory application, data collection and analysis. We expect that the supported Ph.D. student will be a student in the Syracuse University School for Information Studies, which has made significant progress in achieving gender and racial balance in the numbers of students. The current group of 52 Ph.D. students is 50% US-citizens and 50% foreign; 46% female and 54% male; and 44% white, 23% Asian, 17% black or African and 6% Hispanic. Further educational impact will be achieved in this project through the encouragement of under-represented groups at the undergraduate level through collaboration with the McNair program.

5.3 Results from prior research

The PI for this grant, Crowston, has been funded by several NSF awards within the past five years. The awards most relevant to the current proposal is HSD 05–27457 ($684,882, 2005–2008, with R. Heckman, E. Liddy and N. McCracken) and CNS Grant 07–08437 ($200,000, 2007–2010, with M. Conklin, Elon University), for *Collaborative Research: CRI: CRD: Data and analysis archive for research on Free and Open Source Software and its development*. The first award (discussed earlier) supported a study of the evolution of effective work practices for distributed groups, specifically, for free/libre open source software (FLOSS) projects. The project included a component applying NLP techniques to analyze large corpora of email (as noted above) and provided the PIs with significant experience working in an interdisciplinary team. Findings from previous work included a taxonomy of success measures for FLOSS projects, evidence about the structure of projects and descriptions of key practices, e.g., for decision making. The final grant, still ongoing, supports the development of cyber-infrastructure to support the FLOSS research community more broadly, including a repository of FLOSS-related data (FLOSSmole[9]) and of working papers and other research output[10], as well as development of example workflows replicating key FLOSS papers Howison et al. (2008); Wiggins et al. (2008). We will leverage this investment in supporting the proposed work. Overall, NSF-supported research has resulted in nine journal papers (including *ACM Computing Surveys*, *IEEE Software*, *IEEE Transactions on Professional Communications*, *Information Technology Journal* and *IEE Proceedings Software*) (Crowston, 2005; Crowston and Howison, 2005, 2006a,b; Crowston and Scozzi, 2008; Crowston et al., 2006a, 2007, In press; Howison et al., 2006), with others still under review, a book chapter (Crowston, 2008) and multiple conference papers (Crowston et al., 2003, 2005a,b, 2006b, 2008; Heckman et al., 2006, 2007a,b; Howison et al., 2008; Scozzi et al., 2008; Wiggins et al., 2008). These grants have supported a total of six PhD students; several others have been involved in specific aspects of the projects.

[8] http://archive.ics.uci.edu/ml/

[9] http://flossmole.org/

[10] http://flosshub.org/

References

General inquirer home page, 12 September 2002. URL http://www.wjh.harvard.edu/~inquirer/.

Omar Alonso and Stefano Mizzaro. Relevance criteria for e-commerce: A crowdsourcing-based experimental analysis. In *Proceedings of the ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '09, pages 760–761, New York, NY, USA, 2009. ACM. doi:10.1145/1571941.1572115.

Bruce J. Avolio, Fred O. Walumbwa, and Todd J. Weber. Leadership: Current theories, research, and future directions. *Annual Review of Psychology*, 60:421–449, 2009. doi:10.1146/annurev.psych.60.110707.163621.

Leslie A. Baxter. Content analysis. In Barbara M. Montgomery and Steve Duck, editors, *Studying Interpersonal Interaction*, pages 239–254. Guilford Press, 1993.

Michael Beißwenger. Bibliography of chat communications, 2003. URL http://www.chat-bibliography.de/.

Shirley Booth. Computer-assisted analysis in qualitative research. *Computers in Human Behavior*, 9: 203–211, 1993. doi:10.1016/0747-5632(93)90007-F.

Razvan Bunescua, Ruifang Gea, Rohit J. Katea, Edward M. Marcotteb, Raymond J. Mooneya, Arun K. Ramanib, and Yuk Wah Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155, 2005. doi:10.1016/j.artmed.2004.07.016.

James W. Carey, Patrick H. Wenzel, Cindy Reilly, John Sheridan, and Jill M. Steinberg. CDC EZ-Text: Software for management and analysis of semi-structured qualitative data sets. *Cultural Anthropology Methods*, 10:14–20, 1998. URL http://www.cdc.gov/hiv/topics/surveillance/resources/software/ez-text/pdf/eztext.pdf.

Boris Chidlovskii, Jérôme Fuselier, and Loïc Lecerf. ALDAI: Active learning documents annotation interface. In *Proceedings of the ACM symposium on Document engineering (DocEng)*, pages 184–185, New York, NY, USA, 2006. ACM. doi:10.1145/1166160.1166208.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994. doi:10.1007/BF00993277.

Richard Cole and Peter Eklund. Analyzing an email collection using formal concept analysis. In *Principles Of Data Mining And Knowledge Discovery: Proceedings of the European Conference*, pages 309–315, Prague, Czech Republic, 15–18 September 1999. Springer Verlag. doi:10.1007/978-3-540-48247-5_35.

Richard Cole and Gerd Stumme. CEM: A conceptual email manager. In Bernhard Ganter and Guy W. Mineau, editors, *Conceptual Structures: Logical, Linguistic, and Computational Issues: Proceedings of the International Conference on Conceptual Structures*, volume 1867 of *Lecture Notes In Computer Science*, pages 438–452, Darmstadt, Germany, 14–18 August 2000. Springer. doi:10.1007/10722280_30.

Caryn A. Conley and Jennifer Tosti-Kharas. Crowdsourcing content analysis for behavioral research: Insights from Mechanical Turk. Paper presented at the *Academy of Management Conference*, 2010.

Kevin Crowston. Future research on FLOSS development. *First Monday*, 2005. URL http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1465/1380.

Kevin Crowston. The bug fixing process in proprietary and free/libre open source software: A coordination theory analysis. In Varun Grover and M. Lynne Markus, editors, *Business Process Transformation*. M. E. Sharpe, Armonk, NY, 2008. URL http://crowston.syr.edu/node/87.

Kevin Crowston and James Howison. The social structure of free and open source software development. *First Monday*, 10(2), 2005. URL http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1207/1127.

Kevin Crowston and James Howison. Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, 18(4):65–85, 2006a. doi:10.1007/s12130-006-1004-8.

Kevin Crowston and James Howison. Assessing the health of open source communities. *IEEE Computer*, 39(5):89–91, May 2006b. doi:10.1109/MC.2006.152.

Kevin Crowston and Barbara Scozzi. Bug fixing practices within free/libre open source software development teams. *Journal of Database Management*, 19(2):1, 2008. doi:10.4018/jdm.2008040101.

Kevin Crowston, Hala Annabi, and James Howison. Defining open source software project success. In *Proceedings of the 24th International Conference on Information Systems*, Seattle, WA, 2003. URL http://crowston.syr.edu/content/defining-open-source-software-project-success.

Kevin Crowston, Hala Annabi, James Howison, and Chengetai Masango. Effective work practices for FLOSS development: A model and propositions. In *Proceedings of the 38th Hawai'i International Conference on System Sciences*, Big Island, Hawai'i, 2005a. doi:10.1109/HICSS.2005.222.

Kevin Crowston, Kangning Wei, Qing Li, U. Yeliz Eseryel, and James Howison. Coordination of free/libre open source software development. In *Proceedings of the International Conference on Information Systems*, Las Vegas, NV, USA, 2005b. URL http://crowston.syr.edu/content/coordination-freelibre-open-source-software-development.

Kevin Crowston, James Howison, and Hala Annabi. Information systems success in free and open source software development: Theory and measures. *Software Process–Improvement and Practice*, 11(2):123–148, 2006a. doi:10.1002/spip.259.

Kevin Crowston, Kangning Wei, Qing Li, and James Howison. Core and periphery in free/libre and open source software team communications. In *Proceedings of the 39th Hawai'i International Conference on System System*, Kaua'i, Hawai'i, 2006b. doi:10.1109/HICSS.2006.101. URL http://csdl2.computer.org/comp/proceedings/hicss/2006/2507/06/250760118a.pdf.

Kevin Crowston, Kangning Wei, Qing Li, U. Yeliz Eseryel, and James Howison. Self-organization of teams in free/libre open source software development. *Information and Software Technology Journal, Special issue on Understanding the Social Side of Software Engineering, Qualitative Software Engineering Research*, 49:564–575, 2007. doi:10.1016/j.infsof.2007.02.004.

Kevin Crowston, James Howison, and Andrea Wiggins. Opportunities for eScience research on free/libre open source software. In *Proceedings of the Oxford e-Research Conference*, Oxford, England, 11–13 September 2008. URL http://crowston.syr.edu/content/opportunities-escience-research-freelibre-open-source-software.

Kevin Crowston, Xiaozhong Liu, and Eileen E. Allen. Machine learning and rule-based automated coding of qualitative data (poster). Paper presented at the *American Society for Information Science and Technology Annual Conference*, Pittsburgh, PA, 23–27 October 2010. URL http://crowston.syr.edu/content/machine-learning-and-rule-based-automated-coding-qualitative-data.

Kevin Crowston, Kangning Wei, James Howison, and Andrea Wiggins. Free/libre open source software: What we know and what we do not know. *ACM Computing Surveys*, In press. URL http://crowston.syr.edu/content/freelibre-open-source-software-development-what-we-know-and-what-we-do-not-know.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA*, 2002. doi:10.3115/1073083.1073112.

Aditi Dhagat and Lisa Hellerstein. PAC learning with irrelevant attributes. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, Santa Fe, NM, USA, 20–22 November 1994. IEEE Comput. Soc. Press. doi:10.1109/SFCS.1994.365704.

A. Finn and N. Kushmerick. Active learning selection strategies for information extraction. In *Proceedings of the International Workshop on Adaptive Text Extraction and Mining*, 2003. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.287.

Aidan Finn and Nicholas Kushmerick. Learning to classify documents according to genre. *Journal of the American Society for Information Science and Technology*, 57(11):1506–1518, 2006. doi:10.1002/asi.20427.

Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing, Chicago, 1967.

Mark Handel and James D. Herbsleb. What is chat doing in the workplace? In *ACM Conference on Computer-Supported Cooperative Work (CSCW)*, New Orleans, LA, 2002. doi:10.1145/587078.587080.

Robert Heckman, Kevin Crowston, Qing Li, Eileen E. Allen, Yeliz Eseryel, James Howison, and Kangning Wei. Emergent decision-making practices in technology-supported self-organizing distributed teams. In *Proceedings of the International Conference on Information Systems (ICIS)*, Milwaukee, WI, 10–13 Dec 2006. URL http://crowston.syr.edu/node/56.

Robert Heckman, Kevin Crowston, U. Yeliz Eseryel, James Howison, Eileen Allen, and Qing Li. Emergent decision-making practices in free/libre open source software (FLOSS) development teams. In *Proceedings of the 3rd International Conference on Open Source Software*, Limerick, Ireland, 2007a. doi:10.1007/978-0-387-72486-7_6.

Robert Heckman, Kevin Crowston, and Nora Misiolek. A structurational perspective on leadership in virtual teams. In Kevin Crowston and Sandra Seiber, editors, *Proceedings of the IFIP Working Group 8.2/9.5 Working Conference on Virtuality and Virtualization*, pages 151–168, Portland, OR, 2007b. Springer. doi:10.1007/978-0-387-73025-7_12.

James D. Herbsleb and Deependra Moitra. Global software development. *IEEE Software*, 18(2):16–20, March/April 2001. doi:10.1109/52.914732.

J.D. Herbsleb and A. Mockus. An empirical study of speed and communication in globally-distributed software development. *IEEE Transactions on Software Engineering*, 29(3):1–14, 2003. doi:10.1109/TSE.2003.1205177.

Susan C. Herring. Two variants of an electronic message schema. In Susan C. Herring, editor, *Computer-Mediated Communication: Linguistic, Social and Cross-Cultural Perspectives*. John Benjamins, Amsterdam, 1996.

James Howison, Megan Conklin, and Kevin Crowston. FLOSSmole: A collaborative repository for FLOSS research data and analyses. *International Journal of Information Technology and Web Engineering*, 1(3):17–26, 2006. doi:10.4018/jitwe.2006070102.

James Howison, Andrea Wiggins, and Kevin Crowston. eResearch workflows for studying free and open source software development. In *Proceedings of the Fourth International Conference on Open Source Software (IFIP 2.13)*, Milan, Italy, 7-10 September 2008. doi:10.1007/978-0-387-09684-1_39.

Rosie Jones, Rayid Ghani, Tom Mitchell, and Ellen Riloff. Active learning for information extraction with multiple view feature sets. Paper presented at the *Proceedings of the ECML Workshop on Adaptive Text Extraction and Mining (ATEM)*, Washington, DC, 21–24 Aug 2003. URL http://www-2.cs.cmu.edu/~rosie/papers/activelearning-ecml-atem2003.ps.gz.

Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceeding of the SIGCHI conference on Human factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM. doi:10.1145/1357054.1357127.

Klaus Krippendorff. *Content analysis: An introduction to its methodology*. Sage, Newbury Park, CA, 2004.

Giovan Francesco Lanzara and Michèle Morner. Making and sharing knowledge at electronic crossroads: the evolutionary ecology of open source. Paper presented at the *5th European Conference on Organizational Knowledge, Learning and Capabilities*, Innsbruck, Austria, 2004.

Gwendolyn K. Lee and Robert E. Cole. From a firm-based to a community-based model of knowledge creation: The case of Linux kernel development. *Organization Science*, 14(6):633–649, 2003. doi:10.1287/orsc.14.6.633.24866.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Journal of Machine Learning*, 2(4):285–318, 1988. doi:10.1007/BF00116827.

Xiaozhong Liu. *Community Interest as an Indicator for Ranking*. PhD thesis, Syracuse University, 2010.

Matthew B. Miles and A. M. Huberman. *Qualitative Data Analysis: An Expanded Sourcebook*. Sage Publications, Thousand Oaks, 2nd edition, 1994.

Moya Morison and Jim Moir. The role of computer software in the analysis of qualitative data: Efficient clerk, research assistant or Trojan horse? *Journal of Advanced Nursing*, 28(1):106–116, 1998. doi:10.1046/j.1365-2648.1998.00768.x.

Ion Muslea, Steven Minton, and Craig A. Knoblock. Active learning with multiple views. *Journal of Artificial Intelligence Research*, 27:203–233, October 2006. URL http://portal.acm.org/citation.cfm?id=1622572.1622579.

Michael D. Myers. Qualitative research in information systems. *MIS Quarterly*, 21(2):241–242. MISQ Discovery, archival version, June 1997, http://www.misq.org/discovery/MISQD_isworld/. MISQ Discovery, updated version, last modified: November 5, 2008, accessed 27 April 2009, 1997.

Kimberly A. Neuendorf. *The content analysis guidebook*. Sage, Thousand Oaks, CA, 2002.

Matt Ridley. *The Origins of Virtue: Human Instincts and the Evolution of Cooperation*. Viking, New York, 1996.

Judi Scheffer. Data mining in the survey setting: Why do children go off the rails? *Research Letters in the Information and Mathematical Sciences*, 3:161–189, 2002. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.6675.

Michael J. Scialdone, Robert Heckman, and Kevin Crowston. Group maintenance behaviours of core and peripheral members of free/libre open source software teams. Skövde, Sweden, 3–6 June 2009. Springer. doi:10.1007/978-3-642-02032-2_26.

Barbara Scozzi, Kevin Crowston, U. Yeliz Eseryel, and Qing Li. Shared mental models among open source software developers. In *Proceedings of the 41st Hawai'i International Conference on System Sciences*, Big Island, Hawai'i, 2008. doi:10.1109/HICSS.2008.391.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-lim Tan. Multi-criteria-based active learning for named entity recognition. 2004. doi:10.3115/1218955.1219030.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 254–263, Morristown, NJ, USA, 2008. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1613715.1613751.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, Daniel M. Ogilvie, and associates. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA, 1966.

Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Machine Learning Conference*, pages 406–414, Bled, Slovenia, 1999. URL http://www.cs.utexas.edu/~ml/papers/active-nll-ml99.pdf.

Eugene Webb and Karl E. Weick. Unobtrusive measures in organizational theory: A reminder. *Administrative Science Quarterly*, 24(4):650–659, 1979. doi:10.2307/2392370.

Elaine Welsh. Dealing with data: Using NVivo in the qualitative data analysis process. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 3(2), 2002. URL http://www.qualitative-research.net/index.php/fqs/article/view/865/1881.

Andrea Wiggins, James Howison, and Kevin Crowston. Replication of FLOSS research as eResearch. In *Proceedings of the Oxford e-Research Conference*, Oxford, England, 11–13 September 2008. URL http://crowston.syr.edu/content/replication-floss-research-eresearch.

Tianhao Wu and William M. Pottenger. A semi-supervised active learning algorithm for information extraction from textual data. *Journal of the American Society for Information Science & Technology*, 56:258–271, 2005. doi:10.1002/asi.20119.

Ozgur Yilmazel. *Empirical Selection of NLP-Driven Document Representations for Text Categorization*. PhD thesis, Syracuse University, 2006. URL http://www.proquest.com/. (publication number AAT 3241873).

Ozgur Yilmazel. *NLP-Driven Document Representations for Text Categorization: Empirical Selection of NLP-Driven Document Representations for Text Categorization*. VDM Verlag Dr. Mueller e.K, Saarbrücken, Germany, 2008.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003. URL http://www.jmlr.org/papers/volume3/zelenko03a/zelenko03a.pdf.

Tong Zhang, Fred Damerau, and David Johnson. Text chunking based on a generalization of Winnow. *The Journal of Machine Learning Research*, 2:615–637, 2002. URL http://www.jmlr.org/papers/volume2/zhang02c/zhang02c.pdf.