Cover Sheet

Title: CHS: Small: Interactive Machine Learning for Text Analysis

PI: Jeffrey Heer, Associate Professor
Phone: 206-543-6781
E-Mail: jheer@uw.edu

Senior Personnel: Jason Chuang, Post-Doctoral Researcher
Phone: 650-450-0924
E-Mail: jcchuang@cs.washington.edu

Contents

List of Figures

Project Summary

CHS: Small: Interactive Machine Learning for Text Analysis

Jeffrey Heer
University of Washington

Text – including original documents, online correspondence and transcribed speech – is a fundamental data type in a variety of research domains. Tasks requiring text analysis include identifying medical terms in research papers or patient-authored text; finding linguistic markers of affect, politeness or leadership in online discourse; tracking policies across pieces of legislation; and determining consumer sentiments about a product from social media. Across these examples, analysis involves the *recognition* and/or *classification* of phrases or textual categories: researchers iteratively develop or use pre-existing labeling schemas; annotate terms, sentences or full documents; and train and apply statistical classifiers to analyze data at scale.

As automated text mining approaches improve, the *process* of text analysis remains dominated by human effort and supervision. Researchers must collect and manage large text collections, select or develop coding schemes, annotate a subset of the data (either directly or by training coders), identify predictive textual features, tune algorithm parameters, and assess the results of applying automated methods to the full dataset. This process does not proceed in a linear fashion, instead requiring iteration within and across phases, often switching among tools in a manner that stymies provenance tracking and replication.

We propose to develop an integrated, interactive software system to support the process of classification-oriented text analysis. We hypothesize that novel interfaces and supporting algorithms can reduce time and effort and make text analysis methods more accessible to researchers, while retaining – and likely improving – the quality of the resulting classifiers. We will develop an end-to-end system that includes management of documents and metadata; a visual interface for integrated and iterative schema generation, text annotation and model evaluation; and a runtime for managing and comparing multiple learned classifiers.

Core intellectual challenges include the design and evaluation of visual analysis and interactive machine learning techniques, which enable domain experts who may lack training in statistical machine learning to effectively analyze text data. We envision a "virtuous cycle" in which analysts formulate schemas and provide annotations, visualizations facilitate understanding of data and models, and automated methods generalize user input and suggest additional data and features for annotation. We aim to help users also track their progress and replicate analyses. We hope to significantly enhance existing practices of text analysis.

Intellectual Merit: Our research will develop new technical contributions and experimental results. On the technical front, we will investigate system architectures for mixed-initiative text classification; novel user interfaces and visualizations for annotation and model evaluation; interactive techniques for improved feature selection; active learning methods for adaptive sampling of instances and features to label; and facilities for collaborative and crowdsourced labeling. We will conduct evaluations with domain scientists and crowdsourced workers to assess how our methods affect the time and effort required for text analysis, the quality of the resulting classifiers, and the potential biases introduced by automated methods.

Broader Impacts: Our work will lower barriers to entry and enable faster and higher-quality text analysis. The resulting tools can positively impact disciplines that analyze text data. We will work hand-in-hand with our collaborators in multiple domains (health & addiction studies, political science, psychology, sociology) to substantiate these benefits. We will share our tools as open source software runnable as a web service, and leverage our software platform in classroom teaching and undergraduate research.

Keywords: text analysis; data visualization; interactive machine learning; active learning; crowdsourcing.

Project Description

1 Introduction

> *"[N]othing can substitute here for the flexibility of the informed human mind. Accordingly, both approaches and techniques need to be structured so as to facilitate human involvement and intervention... Some implications for effective data analysis are: (1) that it is essential to have convenience of interaction of people and intermediate results and (2) that at all stages of data analysis the nature and detail of output need to be matched to the capabilities of the people who use it and want it."* – John W. Tukey & Martin B. Wilk, 1966 [124]

Though Tukey & Wilk voiced these sentiments nearly 50 years ago, they ring true today: to effectively *facilitate human involvement at all stages of data analysis* is a grand challenge for our age. We seek to address this challenge in the context of text analysis. Across many domains, particularly in the social sciences, text is a primary data source for scholarly research. Tasks requiring text analysis include identifying medical terms in research papers or patient-authored text [21, 78, 95, 132]; finding linguistic markers of affect [17,41,125], politeness [31] or support-seeking [128] in online discourse; tracking reactions to political events [15,65] and predicting elections [125]; and determining consumer sentiments about products or cultural artifacts [84, 115]. Across these examples, analysis involves the *recognition* and/or *classification* of phrases or textual categories: researchers iteratively develop or use pre-existing labeling schemas; annotate terms, sentences or full documents; and train and apply statistical classifiers to analyze data at scale.

The massive amount of text available to researchers now dwarfs their ability to read, comprehend and synthesize the content. Accordingly, researchers are increasingly turning to visualization, natural language processing (NLP) and machine learning (ML) methods to scale text analysis [101, 118]. Yet as automated text mining approaches improve, the *process* of text analysis remains dominated by human effort and supervision [26, 28]. Researchers must collect and manage large text collections, select or develop coding schemes, annotate a subset of the data (either directly or by training coders), identify predictive textual features, tune algorithm parameters, and assess the results of applying automated methods to the full dataset. This process does not proceed in a linear fashion, instead requiring iteration within and across phases [60], often switching among tools in a manner that stymies provenance tracking and replication.

Intellectual Merit: We envision a "virtuous cycle" in which analysts formulate schemas and provide annotations, visualizations facilitate understanding of data and models, and automated methods generalize user input and suggest additional data and features for annotation. We propose the following:

- **Interactive System for Text Analysis**: We will develop an end-to-end web-based system with which researchers can more rapidly perform robust and replicable analyses of English text. We will provide facilities for document and metadata management; interactive text annotation and classifier construction; and export of the products of the analysis process, such as classifiers, annotated text and provenance records. The system will also provide a platform for investigating a variety of research problems.
- **Integrated Visual Coding and Validation**: We will explore novel user interface designs that enable analysts to author label schemas, annotate text and assess coverage and classification results in an integrated, iterative manner. Research challenges include (1) structuring the labeling process to minimize input effort and reduce error, (2) leveraging intermediate classifiers to augment annotation work, and (3) visualizing data and models to assist sample selection, model performance and process convergence.
- **Feature Selection and Refinement**: Text classification relies on extracted features, including counts of words and other linguistic markers. We will (1) develop methods for presenting and evaluating large feature spaces, and (2) investigate the use of unsupervised learning methods (such as continuous word

embedding models [82]) to help analysts augment their analyses with effective domain-specific features.

- **Active and Weakly-Supervised Learning**: In addition to interface design, active learning [90, 110] – such as adaptive sampling of instances or features to label – can accelerate the annotation process [35, 89, 106]. We will explore two forms of interactive learning: (1) preferential sampling of unlabeled instances with high classifier uncertainty and (2) feature-based supervision that enable domain experts to input salient terms, dictionaries or feature constraints enforced via model regularization [35, 39].
- **Collaboration & Crowdsourcing**: Analysts may need to involve multiple annotators. Putting issues of data scale aside, having multiple annotators can reduce bias, evaluate agreement and provide more robust results. When appropriate, crowdsourced workers can also be employed to accelerate and scale the labeling process [78, 113, 115]. We plan to (1) develop a multi-user system with task assignment and management methods to track contributors and assess inter-rater reliability, and (2) build a subsystem for submitting jobs to crowdsourcing platforms such as Amazon's Mechanical Turk and analyzing the resulting labels, addressing research problems of generating task instructions and assessing label quality.

Broader Impacts: This proposal will enable faster and higher-quality text analysis while lowering barriers to entry. If successful, our tools will enable domain experts who lack training in statistical machine learning to effectively analyze text data at scale. We will work hand-in-hand with our collaborators in multiple domains (health & addiction studies, political science, psychology, sociology and studies of scientific collaboration) to substantiate these benefits. We will release our system as open source software, and leverage our software platform in classroom teaching and undergraduate research.

Our previous research projects span model-driven text analytics [25–28, 80, 98]; state-of-the-art classifiers for medical term identification [78] and sentiment analysis [115]; web-based collaborative analysis environments [51, 55, 134]; methods for crowdsourced experiments and data analysis [42, 52, 69, 78, 133]; and popular open-source systems for data transformation [54, 59, 61] and visualization (e.g., D3.js [16] and Prefuse [53]). These experiences give us the necessary background skills to successfully conduct this effort. We seek to bring together these areas of expertise to support the process of classification-oriented text analysis in a systematic, user-centered fashion. In the rest of this proposal, we first describe selected application domains and related prior work. We then describe the research goals outlined above in greater detail.

2 Text Analysis Domains & Collaborating Researchers

To guide and ground our efforts, we are collaborating closely with domain experts in five text analysis areas (see letters of commitment). We have existing collaborative relationships with each team, and (with the sole exception of Intel) have proposal team members physically co-located at each institution.

Patient-Authored Medical Text (with Dr. Anna Lembke, School of Medicine, Stanford). As described later, we have conducted prior research on analyzing patient-authored medical text from online support forums [78] and have a data sharing agreement with MedHelp.org, the world's largest online public health forum. We are working with addiction specialist Dr. Lembke to analyze public posts describing substance abuse behaviors often inaccessible to the professional medical community. Tasks include classifying drugs of choice, phases of addiction, and support-seeking rationale (e.g., information or emotional support [128]).

Open Government Data (with Prof. John Wilkerson, Political Science, University of Washington). Prof. Wilkerson is researching the 2007-08 U.S. financial crisis to identify actors and causes and analyze their relationships. We have access to a large repository of data including transcripts from the Federal Reserve and Financial Crisis Inquiry Commission, copies of major legislation, and hearings leading to the TARP and Dodd-Frank bills. In addition to typical named entities (people and organizations), we seek to recognize collective stakeholders (e.g., home buyers, real estate agents), organizational actions (e.g., mark-to-market accounting), and public sentiments (e.g., collective delusion on continued housing price increases).

Affect in Social Media (with Dr. Douglas Carmean & Dr. Margaret Morris, Intel Research). Our collaborators are mining Twitter text to study emotional expression and arousal across language communities. Their current analysis involves dictionary matches of LIWC terms [122] and an additional "arousal" category that they have developed. While useful, this form of analysis requires constant review and revision to add new terms and features (e.g., emoticons) from additional languages. The team is eager to apply statistical methods, including our proposed feature augmentation technique (§6.2), for improved generalization.

Communication in Distributed Scientific Collaboration (with Prof. Cecilia Aragon, Human-Centered Design & Engineering, University of Washington). Geographically distributed collaboration is increasingly common, and understanding the expression of emotion in computer-mediated communications is crucial to the study of team interactions and processes. Prof. Aragon is working to quantify affect (emotions) expressed by physicists who collaborate remotely across the globe, based on chat logs with over a half million messages [17]. Her team has applied LIWC [122] and found the results unsatisfying. They wish to build a representative set of affect codes, identify predictive features, and classify the desired affects.

Tracking Theories and Methods in Academic Discourse (with Prof. Dan McFarland, Education & Sociology, Stanford). Prof. McFarland is studying academic discourse across Ph.D. theses, including a corpus of over 1M U.S. dissertations. A primary goal is to analyze the dissemination of theories and methods (e.g., statistical or computational techniques) across research communities. Our earlier collaborative work applied topical analysis to track textual similarities among disciplines over time [28,80,98]. We have found that topic models augmented with departmental affiliation metadata provide a useful but coarse-grained overview. We now wish to conduct more fine-grained analyses capable of resolving labeled concepts.

3 Background & Motivation

We first describe related work in text analysis and interactive machine learning (more specific prior work is included in later sections). We also present two examples from our own work that motivate this proposal.

3.1 Related Work: Text Analysis & Interactive Machine Learning

Whether through exhaustive manual coding or the combination of partial labeling and automated classification, text analysis has been applied to a variety of domains. Examples include predicting elections [125], measuring media response to terrorist threats [15], tracking Chinese censorship [65], determining gender and language from tweets [8], analyzing personality from Facebook news feeds [104], detecting fake consumer reviews [84], identifying spam webpages [88], and detecting sarcasm [41] or politeness [31]. Text analysis is at times performed simply by counting the frequency of terms that match pre-defined dictionaries for a category of interest (e.g., for positive or negative sentiment, sexual content, swear words, etc). Example systems and corresponding dictionaries include Linguistic Inquiry and Word Count (LIWC) [122] and the General Inquirer [118]. By generalizing classification rules from a set of provided examples, statistical machine learning methods provide an attractive alternative to the inherent scalability limits of exhaustive annotation and the brittleness of dictionary techniques. Most machine learning formulations assume that (1) a set of label classes are given and (2) a set of examples belonging to each class are provided, as demonstrated by the use of benchmark datasets [91, 99, 126] and evaluation contests [85, 119] to drive research.

However, in many real world applications, the *process* of analysis includes determining a set of labels and then labeling the data. Analysts may not know the appropriate number or specificity of labels at the start of their analysis [70,79]. In some cases, the investigative goal is to evaluate the fit of an existing schema to actual data. Consequently, analysts need to construct an independent set of codes [92]. In other cases, analysts may explore a corpus to determine what codes *can* be extracted from the text, before deciding whether the corpus is relevant to their investigation [45]. Acquiring additional data [9,46] may improve performance, but is often overlooked as an option in tool development. Existing efforts typically address only individual components of the process (e.g., interfaces for labeling data [13, 17], studies of the reliability of human coding [73], and topic modeling to aid human coding [101]) without providing analysts an integrated and interactive system to assist with iterative label formation and annotation.

Text classification performance also depends heavily on *feature selection*, converting unstructured textual content into numerical measures. Text features typically consist of a large set of empirically-determined linguistic markers (e.g., words, substrings of words, part-of-speech tags, capitalization) supplemented with a small set of hand-crafted features. While the former can provide statistics across many types of text, authors of top-performing teams in recent semantic evaluation contests [10, 85] report that the latter contribute significantly to their results. Custom-built features can be especially effective in the analysis of short or domain-specific text, such as the detection of emoticons in social media [11, 103], word shortening to signify dialects [37], or repeated letter sequences to indicate emotional valence [17]. Applying a manually-optimized lexicon can improve classifier performance as much as an improved inference algorithm [36]. Designing custom features, however, can be time consuming [131], error-prone [64], and inaccessible to users who may be unaware of the statistical properties of high-quality discriminative features.

Research on interactive machine learning seeks to effectively integrate ML methods into interactive systems. Much of the work-to-date focuses on specific end-user applications, such as entity resolution [62], metric learning for image search [2,3,38], network event triage [5], and social group generation for content sharing [4]. The Jigsaw system [117] provides interactive visualizations of the output of existing black-box entity recognizers, but does not support labeling or model building. In contrast, we will develop a general text analysis pipeline involving code formation, annotation, classifier evaluation and feature diagnostics. A few interactive tools [89, 106, 109] combine labeling and learning, providing a simple annotation interface and facilities for training classifiers. However, these systems do not support other critical parts of the process such as determining class labels, evaluating the resulting classifiers, and tuning classifier performance.

Other efforts support the general application of ML methods. The popular Weka [48] framework provides a library of algorithms and facilities for conducting experiments to compare models via cross-validation. Mühlbacher and Piringer [83] demonstrate how an integrated visual workbench can accelerate the design and validation of regression models for univariate prediction. The Gestalt system [93] provides an environment for software engineers to both implement and evaluate classifiers, including the use of visualizations to diagnose errors (e.g., confusion matrices linked to source data). These features were found to significantly improve developers' ability to find and fix bugs in machine learning systems. The EnsembleMatrix [120] system demonstrates how human assessment of visualized classifier errors can elicit feedback that leads to more accurate ensembles built of multiple classifiers. We similarly seek to create an interactive system for application and assessment of classifiers, but for domain researchers performing text analysis tasks.

3.2 Example: Topical Analysis of Academic Discourse

In prior research, we have developed tools and methods for supporting large-scale topical analysis of document collections, with a focus on academic text. Our research began with a concrete analysis question in computational social science: can we assess the flow of ideas across academic disciplines, as reflected in the texts they produce? In collaboration with NLP and social science researchers, we developed models and

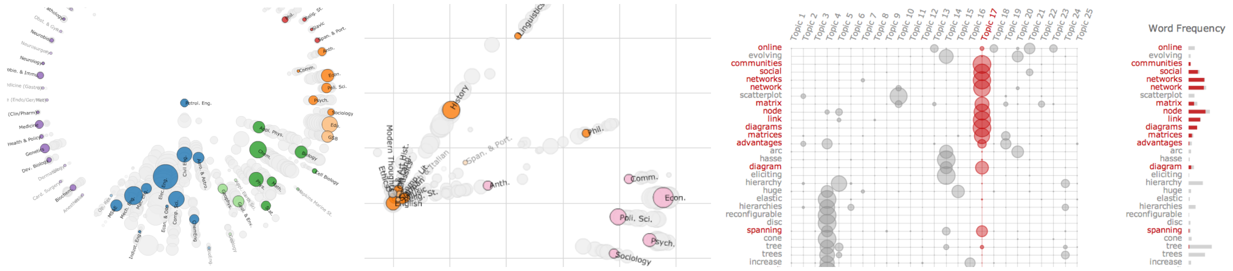


Figure 1: Visual text analysis of academic publications. (a) Left: Similarity between Stanford departments based on published theses. Petroleum Engineering is centered; radial distances convey textual similarity to the other departments. (b) Center: Departments viewed using LDA topic similarity, focused on the English department. We see that the humanities have been clustered far too aggressively. (c) Right: Termite matrix visualization of term-topic distributions for InfoVis research papers learned by LDA.

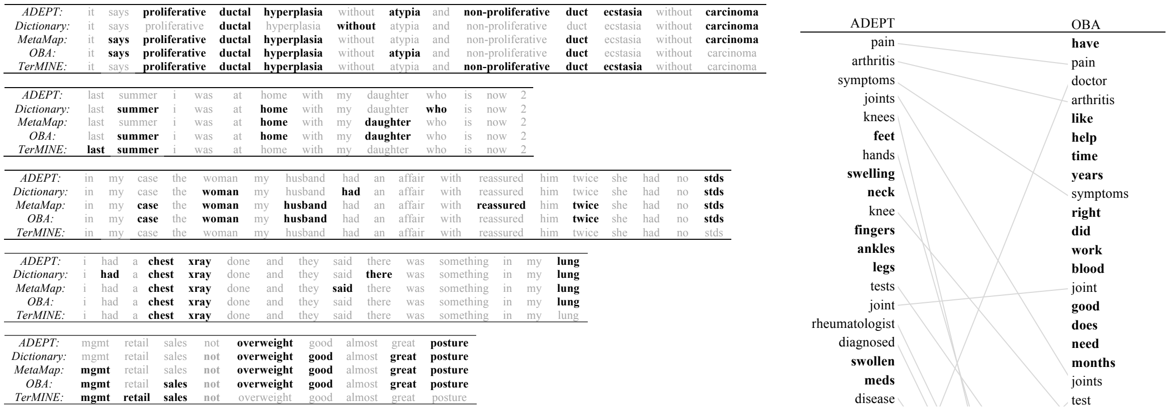
interactive visualizations to explore similarities between academic disciplines over time: first using over 15 years of Stanford dissertations [28] and later expanding to over 1 million U.S. dissertations [80].

We initially envisioned an interface backed by existing NLP methods, such as similarity among tf-idf or LDA (latent Dirichlet allocation [14]) topic vectors. However, we quickly arrived at a visualization that revealed shortcomings in these models: the visualizations laid bare dubious similarities and highly sensitive model parameters (see Figure 1a-b). In turn, we developed new models that better reflect expert opinions of departmental similarity. Through an iterative design process, we formulated an asymmetric "word borrowing" measure that leverages the machinery of Labeled LDA [97], a supervised topic modeling method. This measure better matched the judgments of domain experts (professors) as they assessed departmental similarities. Our final visualization has been used by a varied audience of university administrators and the general public, including coverage in a number of design and science venues (e.g., Discover Magazine). Informed by this experience and other text visualization efforts (e.g., [19, 29, 43, 117, 129]), we have developed a set of design guidelines for the integrated development of statistical models and interactive visualizations [28].

We next investigated how to make topic models more interpretable and relevant to real-world analysis. Reviewing the use of topic models in practice (e.g., [44, 47, 87, 121]), we identified numerous bottlenecks in their application, which despite the unsupervised nature of the algorithms, is dominated by interpretation, parameter tuning and language model modification by people. In response, we developed Termite (Figure 1c), a novel visualization system for assessing topic model output [26]. This work introduced a *term saliency* measure for identifying probable yet distinctive terms, and a *term seriation* algorithm that arranges terms to reveal groupings of related words and preserve phrases to aid rapid scanning. Termite has been released as open-source software and is now in use by a community of data scientists and machine learning researchers.

While Termite enables visual assessment of topic model output, we wished to scale model assessment to thousands of models. This led to the development of a human-centered diagnostics model for evaluating inferred topics [25]. We first conducted an experiment in which domain experts articulated their own mental models of topics in a research domain. The collected data allows us to compare "expert-constructed" topic models to those produced by automatic methods. We can then measure the correspondence between a set of latent topics and a set of reference concepts to quantify four types of topical misalignment: junk, fused, missing and repeated topics. We have applied this method to analyze thousands of topic models, informing choices of model parameters, inference algorithms, and intrinsic measures of topical quality.

Though topic models usefully identify recurring themes, they are too coarse to resolve specific entities of interest, such as research methods referenced in academic text. We are now shifting our focus to fine-grained classification tasks. Analogous to our topic modeling work, we seek to facilitate an analysis processes with significant human involvement: text codification, labeling, classifier construction and assessment.



ADEPT:	it	says	**proliferative**	**ductal**	**hyperplasia**	without	**atypia**	and	**non-proliferative**	**duct**	**ecstasia**	without	**carcinoma**
Dictionary:	it	says	proliferative	**ductal**	hyperplasia	**without**	atypia	and	non-proliferative	duct	ecstasia	without	**carcinoma**
MetaMap:	it	**says**	**proliferative**	**ductal**	**hyperplasia**	without	atypia	and	non-proliferative	**duct**	ecstasia	without	**carcinoma**
OBA:	it	**says**	**proliferative**	**ductal**	**hyperplasia**	without	**atypia**	and	non-proliferative	**duct**	ecstasia	without	carcinoma
TerMINE:	it	says	**proliferative**	**ductal**	**hyperplasia**	without	atypia	and	**non-proliferative**	**duct**	**ecstasia**	without	carcinoma

ADEPT:	last	summer	i	was	at	home	with	my	daughter	who	is	now	2
Dictionary:	last	**summer**	i	was	at	**home**	with	my	daughter	**who**	is	now	2
MetaMap:	last	summer	i	was	at	**home**	with	my	**daughter**	who	is	now	2
OBA:	last	**summer**	i	was	at	**home**	with	my	**daughter**	who	is	now	2
TerMINE:	**last**	**summer**	i	was	at	home	with	my	daughter	who	is	now	2

ADEPT:	in	my	case	the	woman	my	husband	had	an	affair	with	reassured	him	twice	she	had	no	**stds**
Dictionary:	in	my	case	the	**woman**	my	husband	**had**	an	affair	with	reassured	him	twice	she	had	no	**stds**
MetaMap:	in	my	**case**	the	**woman**	my	**husband**	had	an	affair	with	**reassured**	him	**twice**	she	had	no	**stds**
OBA:	in	my	**case**	the	**woman**	my	**husband**	had	an	affair	with	reassured	him	**twice**	she	had	no	**stds**
TerMINE:	in	my	case	the	woman	my	husband	had	an	affair	with	reassured	him	twice	she	had	no	stds

ADEPT:	i	had	a	**chest**	**xray**	done	and	they	said	there	was	something	in	my	**lung**
Dictionary:	i	**had**	a	**chest**	**xray**	done	and	they	said	**there**	was	something	in	my	**lung**
MetaMap:	i	had	a	**chest**	**xray**	done	and	they	**said**	there	was	something	in	my	**lung**
OBA:	i	had	a	**chest**	**xray**	done	and	they	said	there	was	something	in	my	**lung**
TerMINE:	i	had	a	**chest**	**xray**	done	and	they	said	there	was	something	in	my	lung

ADEPT:	mgmt	retail	sales	not	**overweight**	good	almost	great	**posture**
Dictionary:	mgmt	retail	sales	**not**	**overweight**	**good**	almost	**great**	**posture**
MetaMap:	**mgmt**	retail	sales	**not**	**overweight**	**good**	almost	**great**	**posture**
OBA:	**mgmt**	retail	**sales**	not	**overweight**	**good**	almost	**great**	**posture**
TerMINE:	**mgmt**	**retail**	**sales**	not	overweight	good	almost	great	posture

Figure 2: Comparison of terms identified as medically-relevant by different models. (a) Left: comparison of five models (classified terms shown in black), including our CRF-based ADEPT model. OBA and MetaMap runs use the SNOMED CT ontology. (b) Right: Term rankings for ADEPT and OBA on Arthritis forum data. Terms occurring in both lists are connected by a line.

3.3 Example: Extracting Medical Terms from Patient-Authored Text

Our proposal is motivated by our ongoing work developing classifiers for patient-authored medical text. Online health-seeking behavior is growing rapidly: 59% of U.S. adults looked for health information online in the past year, and 35% attempted to diagnose a health condition online [94]. One result of this trend is the accumulation of patient-authored text (PAT) in the form of blog posts, online health forum discussions and email. Analysis of online health behaviors can lead to new medical insights and assist tasks such as tracking disease trends [18, 22] and discovering previously unknown links among conditions and/or treatments [21, 130, 132]. However, PAT is difficult to analyze due to lexical, semantic and conceptual differences from text authored by medical experts, limiting the utility of existing tools such as MetaMap [7] and OBA [58].

A data-sharing agreement with MedHelp (`www.medhelp.org`), the world's largest online health forum, gives us access to hundreds of thousands of patient-authored discussion posts, covering roughly 200 topics. An initial challenge is to extract medically-relevant terms (such as conditions and treatments) for further analysis. However, medical experts (doctors, nurses) have limited time, making it difficult to get copious labeled data. In response, we have investigated how to direct crowds of non-experts (workers on Amazon's Mechanical Turk) to label medically-relevant terms in PAT with accuracy comparable to annotations we collected from registered nurses. Achieving consistent labeling required several iterations of the task prompt and examples, as well as experimentation to determine optimal voting schemes. For example, asking users to only tag words/phrases that they thought *doctors* would find interesting mitigated numerous inconsistencies. We then used over 10,000 crowd-labeled sentences to train a conditional random field (CRF) classifier. Our model widely outperforms prior state-of-the-art tools for medical term extraction (F1-score of 77.7% versus OBA's 47.2%, MetaMap's 39.1% and a dictionary baseline of 38.7%). Our annotation method and results were recently published in the Journal of the American Medical Informatics Association (JAMIA) [78].

In ongoing work, we are investigating how to use weak supervision as an alternative to term-level annotation. Given existing dictionaries of conditions and treatments, can we bootstrap effective, generalized classifiers? Lexico-syntactic pattern learning [50], an effective but less-popular technique for term extraction, outperforms existing MetaMap and OBA tools, as well as a CRF trained using dictionary matches as positive examples. We are able to discover several novel terms not in existing dictionaries or ontologies.

In collaboration with addiction specialist Dr. Anna Lembke, we are now focusing on patient-authored text regarding substance abuse, which documents abuse behaviors and detoxing strategies otherwise inaccessible to medical professionals. After extensive open coding to determine medically-relevant concepts, we have

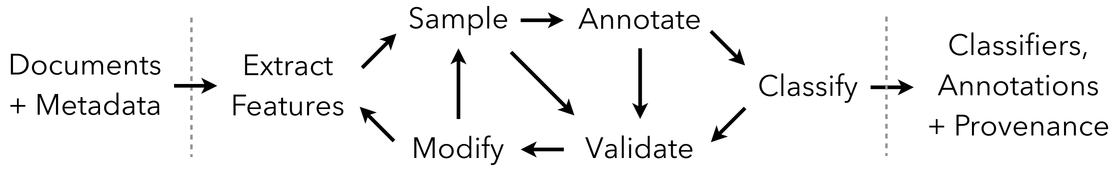


Figure 3: Proposed interactive text analysis workflow.

had initial success training a logistic regression classifier for drug of choice (F1=81.4%). These labels are highly context sensitive, as substances (e.g., Xanax, Methadone) may serve either as helpful treatments or as abused substances. We are now exploring document-level logistic regression and CRF models for identifying information vs. emotional support seeking and phase of addiction (e.g., using, quitting, etc).

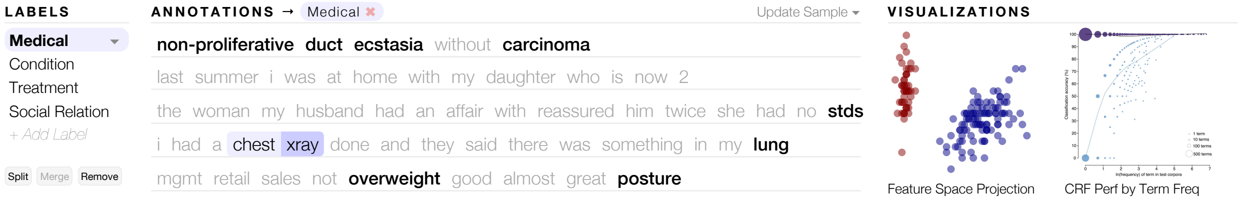
Across these activities, developing custom classifiers has proven time-consuming and labor-intensive. Labeling data is not only tedious, it requires careful analysis and iteration to ensure agreement among annotators, involving modification of the labeling rubric and reassessment of prior labels. Similarly, authoring effective prompts and examples for crowdsourced workers required much iteration. Experimenting with models and features also has consumed significant effort. For the substance abuse data, hand-engineered features based on observed patterns have contributed substantial improvements to classifier accuracy. There is little support for the overall process of analysis: each of the above phases requires switching among different tools and manual record keeping of the results across numerous iterations (e.g., labeling disagreements, features assessed, classifier errors). Interactive tools that integrate data profiling, annotation, model training and assessment can vastly accelerate development while also recording provenance and enabling replication. Moreover, we would like to empower our collaborators to conduct such analyses on their own.

4 An End-to-End Interactive Text Analysis System

Our goal is to develop an interactive system with which domain experts can conduct, evaluate and publish state-of-the-art text analyses. We will provide integrated support for the *process* of text analysis. Our end-to-end system will provide usable tools for collaborating domain scientists, enable empirical study of the text analysis process, and alleviate the accessibility, overhead and provenance-disrupting costs of current practices involving disparate tools. We believe such a system is timely: not only are scientists increasingly interested in scalable text analysis methods, we are at an opportune point in time to leverage developments in visualization tools, active and constraint-based learning, and crowdsourcing systems. We intend for our system to provide a test-bed framework for the research activities described in this proposal as well as for additional future work. Figure 3 shows a basic schematic of the text analysis workflow of our proposed system. Many of the components are discussed in detail in subsequent sections. Here, we briefly describe aspects which require engineering effort but not necessarily new research.

One critical piece of infrastructure is document and annotation management: we will provide support for importing text documents and metadata. Example inputs include ASCII, HTML, or PDF files, relational tables with text fields, and external metadata such as dictionaries, ontologies and term resolution maps. Upon ingest, we will perform optional segmentation (e.g., by sentence), text processing (e.g., tokenization, stemming) and feature extraction (e.g., capitalization status, word-grams, part-of-speech tagging). Following existing language toolkits [12, 116], we will manage extensible *annotations* for documents and terms.

Another aspect is classifier and experiment management. We will initially focus on the use of logistic regression for classification and conditional random field models for sequence labeling. However, we will design the system with appropriate interfaces to enable the extension to additional classifiers (e.g., random decision forests, support vector machines, ensemble methods) in the future. We will also include runtime support for applying classifiers, exporting results, and evaluating them via cross-validation.



non-proliferative duct ecstasia without carcinoma

last summer i was at home with my daughter who is now 2

the woman my husband had an affair with reassured him twice she had no stds

i had a chest xray done and they said there was something in my lung

mgmt retail sales not overweight good almost great posture

Figure 4: Interface mockup with label management, annotation and visualization. Annotation is currently focused on a single binary label ("Medical"). Hovering over the term "xray" triggers selection previews: the dark blue region is labeled upon single click, the full blue region (a noun clause) upon double click. Visualizations show dimensionality reduction of terms (left) and error analysis of current classifier accuracy vs. term frequency (right); users can lasso regions to sample or batch label instances.

We will implement a two-tier system: a server-side component for text management and analysis, and a client-side component for visualization and interaction. We plan to write the server-side component in Java, using well-established tools such as the Stanford CoreNLP framework [116] and the Apache Lucene [6] search engine. Our research team has used both extensively in prior work. We will also use a relational database for persistence and querying of extracted features and metadata, as well as event logging and user session management. While backend scalability is not the primary focus of this proposal, as needed we will work with collaborator Carlos Guestrin (see letter of commitment) and his group's GraphLab system [74,75] for distributed, large-scale machine learning. The client-side interface will be an HTML5 single-page web application, with visualizations built using the D3.js (Data-Driven Documents) [16] library created by our research lab. The two tiers will communicate using a web services API, facilitating reuse of our server by other client systems. The API will include logging facilities at the input and application event levels both to record provenance for replicability and to enable analysis of usage data.

4.1 Summary of Tasks and Goals

- **End-to-end system**: We will build an integrated system for importing text documents, performing annotation, training classifiers and evaluating the results in an iterative loop. The system, consisting of a server and web client, will provide a platform for the research efforts discussed in the following sections.
- **Text and metadata management**: Our system will support import, segmentation, feature extraction, indexing and annotation management. The server will act as a data source for client interfaces.
- **Publishing results**: The system will support export of learned classifiers, labeled text data and evaluation results to enable both publication and dissemination of results.
- **Provenance & replication**: The system's logging architecture will enable review and reapplication of user annotations to support replication and reuse on new or evolving data sets.

5 Integrated Visual Coding and Validation

At the heart of our proposed system is a user interface for authoring label schemas, annotating text data (either documents or individual terms) with those labels and then using the annotations to train and evaluate classifiers. We propose to combine these processes within an integrated user experience. For example, our system should support open coding through evolving label schemas, accelerate annotation to reduce tedium, and facilitate validation throughout the analysis process. Figure 4 contains a mockup of one early-stage design idea for combining label schematization, rapid annotation and data visualization. We will explore multiple alternative designs and evaluate them in an iterative design process. Here, we discuss some of the research and design challenges we intend to investigate. In subsequent sections, we will go into further details regarding feature selection (§6) and active learning (§7) components.

5.1 Annotation Acceleration

Our interface will enable analysts to annotate either text segments or terms with class labels. To accelerate this process, we will investigate multiple strategies for accelerating annotation actions and reducing errors.

Text selection: In addition to keyboard shortcuts, we will explore efficient text selection methods. We will analyze usage data for recurring selection patterns. For example, part-of-speech tags might guide multi-click selections in which the first click selects a term, and the second click selects an encompassing noun phrase.

One-class-at-a-time annotation: Deciding among multiple class labels may require increased decision times or significant context-switching on behalf of the user [20]. We will experiment with annotation strategies that consider only a single label at a time, treated as a binary annotation. Prior work has found significant benefits for such "column-oriented" approaches in form entry applications [23], reducing input effort and increasing overall data quality. We hypothesize this strategy will prove helpful for term annotation in particular; and useful for parallelization and task simplification when crowdsourcing annotations (§8).

Reduce annotation to confirmation: Our system can progressively train classifiers as users produce annotations; alternatively, application of dictionaries or feature-space annotations can provide initial, albeit crude, labels. We will explore the utility of applying such intermediate classifiers to turn annotation tasks into one-click (or one-keystroke) confirmation tasks. If a document or term is labeled correctly, the user might take no action, and only disconfirm inaccurate labels (or vice versa). We will investigate if such an approach is generally useful or limited to tasks such as validation of labels with high classifier confidence (§7).

Batch annotation: We will explore approaches for annotating multiple instances simultaneously by automatic clustering of similar instances and selecting documents and feature space regions within data visualizations. For example, one might associate specific words, dictionaries or features with a given class label.

5.2 Data and Process Validation

Our system will train classifiers as users label data, both to drive active learning (§7) and to support validation throughout the analysis process. Classifiers are typically evaluated using measures such as precision, recall and F1 score (their harmonic mean). While valuable, these measures have limitations: they do not reflect upstream errors such as annotator mislabeling or provide diagnostic information for improving a classifier. In isolation, these measures do not establish either lower or upper performance bounds. What if the annotations cannot be predicted by the available features? To aid human-in-the-loop analysis, we will investigate interaction and visualization techniques to aid labeling and validation.

Text data visualization: We will investigate visualization methods for viewing instances of input text data (e.g., documents or terms) in the context of extracted features and provided labels. For example, visualizations of how instances distribute across features or related statistics (e.g., corpus term frequency, Figure 4) may help guide feature selection and sample coverage. We will also explore the use of dimensionality reduction methods [105, 127] to plot feature-space representations of documents or terms (as in Figure 4). Such views can reveal clusters of similar instances. We can further explore techniques for labeling regions (or user selections) in the projected view by dominant features contributing to instance similarity. As annotations are collected, instances may be correspondingly colored to assess label-feature correlations. As classifiers are trained, we can rank features by their current contribution to a model (e.g., coefficients from logistic regression). While useful in isolation, such visualizations are especially powerful in combination. We will support common interaction techniques such as linked selection (i.e., "brushing and linking") and details-on-demand (e.g., retrieving source text for selected data points) to facilitate exploratory analysis.

Schema validation and refactoring: To assess label schemas we will visualize correlations among labels and annotators. Inter-rater agreement statistics can provide a baseline for classifier evaluation. Visualizing systematic patterns of disagreement can inform schema design and instructions. For individual annotators, comparing highly-similar or intentionally duplicated instances may aid assessment. To facilitate evolving schemas, we will identify labels with high error rates or poor discrimination under current classifiers, and support user interface operations to merge or split labels (splitting may be assisted by a combination feature-space clustering and active re-labeling), and to retrain classifiers on a reduced subset of labels.

Process assessment and error analysis: To assess current classifier performance we can plot statistics (e.g., cross-validated accuracy, precision, recall, or F1) over increasing sample sizes. Such plots can help assess the rate of classifier improvement. Are additional labels likely to further improve performance? As appropriate, assessment can include comparison of multiple classification algorithms and/or parameter settings. We will also incorporate visualizations for fine-grained exploration of current classifier performance. For example, confusion matrices [93, 120] can reveal common misclassification patterns among multiple labels, while plotting classifier performance against predictors such as frequency (see Figure 4) can help assess if misclassification may be due to insufficient examples of rare instances.

5.3 Summary of Tasks and Goals

- **Integrated annotation and validation**: Design novel interfaces that integrate schema authoring, annotation and classifier evaluation to facilitate iterative, human-in-the-loop analysis.
- **Annotation acceleration**: Design to reduce input effort and error: augment selection, explore single-class annotation strategies, supplant labeling with confirmation and investigate batch annotation.
- **Data and process validation**: Visualize text data according to extracted features and supplied labels. Support label schema modification, including splitting and merging of existing codes. Design classifier performance and diagnostic plots to assess progress and convergence.

6 Feature Selection and Refinement

Text classification requires extracting linguistic features from unstructured text, which then serve as input data to learning algorithms [49]. Classifier performance depends heavily on whether the extracted features are sufficiently expressive with respect to the text corpus and sufficiently discriminative with respect to the user-supplied schema. Our system will include components to help users manage, author and evaluate effective textual features specific to their analyses. We will investigate the design of visualizations and interfaces to support feature exploration and to evaluate the contribution of features.

6.1 Feature Management and Assessment

Our system will include several classes of features, along with tools to help users evaluate and refine the feature space. Following current best practices, we will automatically extract empirically successful features such as the counts of words, n-grams, and character n-grams, as well as statistics derived from part-of-speech tagging and common named entity types. Our system will also provide user interfaces to manage manually-crafted dictionaries, a common way for users to express custom vocabularies relevant to their schema.

In many classification tasks, the number of labeled instances is smaller than the number of features. As a result, the ability to discriminate most instances may be attributed to multiple features, and over-fitting is a concern. The decision to include or exclude a feature often falls on the analyst who must assess whether a feature is expressive or is over-fitting the training data. As mentioned in §5.2, we will design visualizations to help users explore the space of features and to reveal patterns such as features that fire consistently. While visualization techniques exist for visualizing dozens or more continuous dimensions (e.g., parallel coordinates [56]), feature visualization involves a larger space of 10,000+ dimensions that are typically binary or discrete. We will integrate feature visualization with other schema- and document-based visualizations to help users determine correlation between features, original text, and annotations. We will also examine corresponding user interactions to support feature exploration. Given thousands of features, turning individual features on and off is infeasible on the whole. We will provide support such as ranking, grouping, filtering, and re-weighting to help users assess feature contributions. We will explore hierarchical organizations of features to help users manage groups of features at once.

6.2 Unsupervised Feature Learning and Refinement

An emerging line of research applies unsupervised techniques, such as deep learning [40, 77, 82, 115] or topic models [14], to improve domain-specific classification tasks. We will investigate the use of continuous word embedding and latent topics – automatically generated from a reference text corpus – as classifier features. While these word representations can improve classifier performance [77], users are often left with a take-it-or-leave-it decision, with few options to assess or refine these features. We will investigate multiple forms of support for incorporating such features. First, we will provide tools to help users identify and label unsupervised dimensions (such as latent topics) relevant to a task. For example, our prior work on topic models [25, 26] addressed how to visualize latent topics and align them with interpretable reference concepts. Second, we will provide tools to help users quickly augment lexicons, either to create improved dictionaries or form groups of semantically-related terms. In recent unpublished work, we have found that given a set of related seed terms, we can identify concept-specific axes (suitable for use as a classifier feature) within word embedding models. A user provides a dictionary or example terms, and we learn a word vector model subspace corresponding to a semantic category containing those terms (e.g., emotion words or country names). By subsequently identifying other terms in this learned space, we can automatically extend or adapt text analysis resources such as LIWC dictionaries. By propagating annotations from given terms to nearby terms in the vector space, we might also better amplify feature-space annotations (§7).

6.3 Summary of Tasks and Goals

- **Feature management & assessment**: Design visualizations to help analysts track and assess their exploration of the feature space. Develop interactions to help analysts effectively refine features.
- **Unsupervised feature learning and refinement**: Combine unsupervised feature learning with end-user refinement, so that analysts can more easily author effective domain-specific features.

7 Active and Weakly-Supervised Learning

A key goal of this proposal is to reduce tedium in supervising learning systems and provide interactive insight into their construction. Supervised learning has enabled major improvements to the accuracy and robustness of document analysis and information extraction. However, a primary obstacle is the limited availability of domain-specific *expert-labeled* data, which can require significant time and labor. *Active and weak supervision* methods [34, 39, 90, 110] provide an efficient alternative for creating accurate classifiers.

We plan to start with two common machine learning methods: logistic regression (which treats each instance as independent) and conditional random fields (which also model transition probabilities for label sequences). Both are widely-used and amenable to the feature-based supervision methods described below [34,39]. Going forward, we will consider expanding to other classifiers, such as random decision forests, support vector machines, or deep learning methods. Our initial implementation will use batch sampling and model updates; as needed, we will investigate improved interactivity through online learning methods. On these tasks we will collaborate closely with our faculty colleague and machine learning expert Prof. Carlos Guestrin (see included letter of commitment).

7.1 Active Learning to Sample Unlabeled Examples

Our learning process will interleave data exploration by an analyst, instance labeling and constraint authoring. To seed the process, the analyst can label an initial set of examples and/or features for each category or field. Our system will then use the current predictions of the model to assess which features are likely to reduce uncertainty about its predictions using expected information gain and its approximations [35, 81, 107, 112]. For example, a common approach is to sample instances with the the highest uncertainty or which lie closest to current classifier decision boundaries [110].

To optimize the use of an analyst's time and attention, selected examples should be both informative and diverse. Nearly redundant features and examples which dominate large-scale data will simply drown out the signal. To determine an appropriate initial sample, we will investigate alternatives to uniform random sampling. For example, hybrid active learning [76] first clusters instances in an unsupervised fashion and then uses the clusters to perform stratified sampling. We will experiment with augmenting this approach with analyst input through selection of desired features or clusters in overview visualizations, and use visualizations to select and label multiple instances simultaneously to perform batch active learning [108].

7.2 Feature-Based Supervision to Incorporate Domain Knowledge

Traditional forms of active learning sample unlabeled instances believed to be most informative for improving a model. However, labeling large numbers of examples may be inefficient, especially when an analyst possesses valuable domain knowledge about the feature space. Early work in this area applies boosting to features believed to be more informative [96], but does not associate features with specific classes. More recent work uses feature-space annotations (e.g., indicating specific words that are associated with a given class label) to adjust model priors [106, 109] or constrain inference [33–35, 39].

We propose to incorporate Ganchev et al.'s *posterior regularization* [39] framework to enable feature-based weak supervision. Posterior regularization incorporates partial supervision for latent variable models using moment constraints on model posterior distributions. For example, suppose we want to learn how to extract not just the polarity of a product review, but more specific aspects. In restaurant reviews, we might want to identify comments about food, service, and ambiance [114, 123]. Chain-structured models, such as CRFs, are the tools of choice for such tasks, where each word is associated with a variable corresponding to the field type (e.g., food, service, ambiance). In addition to choosing words indicative of each field, an analyst may specify that food descriptions typically come before service and ambiance, and often constitute over half the words in a review. In general, an analyst might specify a conjunction of such "features" that refer to states and roughly constrain their proportion (expectation under the model). Posterior regularization framework incorporates such constraints into model estimation without changing its structure or the complexity of inference. The learning algorithm resembles Expectation Maximization (EM), but involves an additional projection step which enforces constraints. Our interface will allow analysts to select features, annotate them to produce constraints, and see examples that these features impact most.

Browsing of constraints at interactive speeds will be enabled by approximate, incremental re-training of the model. Recent work on stratified sampling [34] has shown promising results in approximating feature relevance by using small, well chosen subsets of the data. For some features, the effect on predictions can be seen even using a very small subset of examples, but others require the entire data. Posterior regularization inherits properties of the EM algorithm that allow incremental and approximate updates [39, 86]. Our interface will allow the user to see the approximate results using a small, local subset of the data, while progressively more accurate results are computed in the background. Thus, the analyst can quickly modify the model if the approximate results do not seem promising.

7.3 Summary of Tasks and Goals

- **Selecting informative and diverse examples or features**: Incorporate active learning methods for sampling promising and non-redundant examples and feature constraints for analysts to evaluate.
- **Constraint-based supervision**: Design simple and effective visual interface and process for expressing constraints, which are then enforced via posterior regularization.
- **Fast evaluation of the impact of changes**: Construct approximations of constraint impacts for interactive model building, enabled by progressive model-refining in the background.

8 Collaborative & Crowdsourced Labeling

To annotate large unlabeled data sets, *collaborative*, and more recently *crowdsourced*, annotation procedures are common. Accordingly, our system must include support for integrating the contributions of multiple annotators. We will include a user model to track who is using the system and their annotations and actions. Our sampling procedures can use this information to request a set of redundant annotations to assess inter-rater reliability or evaluate the performance of assistants. We will also provide flexible aggregation methods (e.g., voting thresholds) to determine how to handle conflicting judgments.

8.1 Crowdsourcing Annotation Tasks

Crowdsourcing platforms, particularly Amazon's Mechanical Turk [57], have become increasingly popular for user studies [52, 66], text annotation [78, 113]), and even performing complex activities such as explanatory [133] and taxonomic [24] data analysis. By farming out annotation tasks to a pool of hundreds or even thousands of workers, researchers can scale labeling with dramatically improved time and cost. Still, ensuring high quality responses presents a serious challenge. Crowdworkers may misinterpret a prompt or task, exhibit varying levels of effort, or outright scam by rapidly producing inauthentic responses. Many studies engage crowdworkers to annotate documents on general topics such as movie reviews [115] or news articles [106, 109]; recruiting or training crowdworkers with domain expertise, however, remains difficult.

To assist such efforts, we will research methods for reliably eliciting and integrating high-quality crowdsourced labels in text analysis workflows. Prior crowdsourcing research has developed programming frameworks to support task allocation and adaptive jobs [1, 72]; tools for authoring complex, multi-phase crowd workflows [67, 68, 71]; and visualization tools for inspecting worker activity [32, 102]. We intend to provide more targeted support for guiding and evaluating text annotation tasks: we will provide facilities to submit jobs to Mechanical Turk, which in turn will direct crowdsourced workers to a version of our annotation interface. Our system will log worker actions, collect annotations and make the results accessible through existing visualization and collaboration facilities. After first eliciting judgments from a domain analyst, the system will have "ground-truth" labels with which to evaluate the quality of worker responses and determine appropriate aggregation schemes (e.g., corroborative vs. majority voting). Users will then be able to selectively include crowdsourced annotations in their analysis pipeline. Going forward, we envision our system facilitating the development and evaluation of more elaborate crowd management schemes (e.g., [30, 63]).

8.2 Semi-Automated Task Instruction

Providing understandable, unambiguous instructions is critical to facilitating high-quality annotations. In our own work we successfully employed workers on Mechanical Turk to label medically relevant terms in over 10,000 sentences [78], but doing so required multiple iterations of instruction design in which we clarified the nature of "medically relevant" (e.g., "what terms would a doctor be interested in") and presented suitably diverse, informative examples. Similarly, our prior work on crowdsourcing explanations for patterns in data [133] first required extensive validation of different task design strategies. Using active learning methods (§7), we can partially automate the process of instruction formation by suggesting diverse examples to include in worker instructions. To expedite convergence, we can also allow users to submit jobs with various prompts and analyze the resulting labels before running larger-scale annotation jobs.

8.3 Summary of Tasks and Goals

- **Collaboration support**: Our system will track and aggregate contributions from multiple users.
- **Crowdsourced labeling**: We will develop facilities for submitting annotation tasks to Mechanical Turk, visualizing worker activity and evaluating the responses.
- **Instruction generation**: We will research new methods to assist the generation and evaluation of task instructions to facilitate higher-quality responses.

9 Evaluation

In addition to ongoing usability studies, we will evaluate different configurations of our system through controlled experiments and long-term deployments with crowdsourced workers and domain researchers.

9.1 Controlled Experiments

To assess our system we will conduct a series of controlled experiments on real-world analysis tasks throughout the lifecycle of the project. With but a few exceptions [100, 106, 109], evaluations of active learning systems for text analysis use simulated user input drawn from pre-labeled data. Moreover, they assume that users are oracles with perfect accuracy. In contrast, we will ask subjects to interactively construct text classifiers and compare the results across different system configurations. We will draw on existing benchmark data sets from the text mining literature as well as data from our own prior work on patient-authored medical text. We will run initial experiments in person with collaborating research teams and their students. We will then conduct larger-scale experiments by recruiting crowdsourced workers as participants [52, 66]. In addition to scaling the participant pool, this strategy will allow us to compare domain expert and non-expert users and also compare the relative contributions of active learning methods and crowdsourced annotation.

Independent factors that we can manipulate include: (1) classification unit (document vs. term), (2) number of label classes, (3) labeling strategy (parallel vs. serial consideration of classes), (4) available visualizations, and (5) active learning support (random sampling vs. uncertainty sampling vs. feature constraints). Given the large space of possible experiments, we will conduct a series of accretive experiments, rather than a full-factorial design. Dependent variables of interest include classifier performance (precision, recall, F_1-score, accuracy), time on task, and the number and type of samples or features annotated. We will also conduct error analyses, in part to look for systematic biases that may result from the above manipulations. For example, do active learning methods result in different patterns of misclassification?

9.2 Longitudinal Case Studies

We will also conduct long-term case studies [111] with our collaborators (§2). We will make our system available to collaborators through a hosted web service which we will maintain, enabling interaction and event logging for usage analysis. We will schedule regular meetings with our collaborators to interview them on their experiences (when appropriate using usage data as an elicitation prompt), demonstrate new features, receive feedback and prioritize future efforts. In addition, we will solicit feedback from, and provide support for, external researchers who download and use open source releases of our software.

9.3 Summary of Tasks and Goals

- **Controlled experiments**: We will conduct controlled experiments with both domain experts and crowd-sourced workers to systematically assess our design decisions on classifier and user performance.
- **Longitudinal case studies**: Through long-term deployments with collaborating researchers we will assess tool usage and utility, with the goal of facilitating novel research results across varied domains.

10 Research Timeline

We will develop our system using a phased approach: we will start by scaffolding an end-to-end system, then refine it with more functionality. Doing so, we can explore multiple research questions in parallel, then integrate successful results. This strategy allows us to deploy and gain user feedback early in the process to adaptively prioritize the research. The research team will consist of PI Heer, Senior Personnel Jason Chuang, multiple PhD students (e.g., Diana MacLean, Jeff Snyder), undergraduate researchers and our collaborators. Year 1 effort will focus on an initial system supporting text ingestion, feature extraction, annotation management and classification support (logistic regression, CRF) on the server, and an application scaffolding and annotation interface for the web client (All, §4-5). We will deploy the system with our research collaborators and roll out new features as they mature. In parallel, we will investigate multiple model assessment visualizations (All, §5), feature augmentation methods (Chuang, §6) and active learning support (Snyder, §7). Moving forward into year 2, we plan to develop crowdsourcing and task generation support (MacLean, §8). We will further refine each research component, initiate controlled experiments (§9) and integrate new features with periodic software releases. In year 3 we will continue to refine and integrate additional features in response to our ongoing experiments and collaborator feedback. At this point, we will further package and document the system such that our open source release is usable by a larger community of researchers.

11 Results from Prior NSF Funding

PI Jeffrey Heer is an Associate Professor of Computer Science & Engineering at the University of Washington, and previously an Assistant Professor of Computer Science at Stanford University (2009–13). He has received two prior collaborative NSF grants: IIS-1017745 "HCC: Small: Graphical Preception Revisited: Developing and Validating Design Guidelines for Data Visualization" ($250k, 2010–13) and CCF-0964173 "DIC: Medium: Scalable, Social Data Analysis" ($333k, 2010–14). These awards have led to over a dozen papers in the top venues in Human-Computer Interaction and Information Visualization (CHI, UIST, InfoVis, VAST & EuroVis), including best paper or honorable mention awards in each of these conferences. NSF support for his work on interactive data transformation (CCF-0964173) led to founding Trifacta Inc. (with Joe Hellerstein & Sean Kandel), which has raised over $16M in venture capital. These NSF awards do not overlap with this proposal. Heer is also a Faculty Participant on NSF-1258485 "IGERT-CIF21: Big Data U: A Program for Integrated Multidisciplinary Education & Research for Big Data Science", led by PI Carlos Guestrin. The current proposal is complementary to the educational aims of the IGERT.

References Cited

[1] Salman Ahmad, Alexis Battle, Zahan Malkani, and Sepander Kamvar. The jabberwocky programming environment for structured social computing. In *ACM User Interface Software and Technology (UIST)*, pages 53–64, 2011.

[2] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. Overview based example selection in end user interactive concept learning. In *ACM User Interface Software and Technology (UIST)*, pages 247–256, 2009.

[3] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. Examining multiple potential models in end-user interactive concept learning. In *ACM Human Factors in Computing Systems (CHI)*, pages 1357–1360, 2010.

[4] Saleema Amershi, James Fogarty, and Daniel Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *ACM Human Factors in Computing Systems (CHI)*, pages 21–30, 2012.

[5] Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, and Blaine Christian. Cuet: Human-guided fast and accurate network alarm triage. In *ACM Human Factors in Computing Systems (CHI)*, pages 157–166, 2011.

[6] Apache lucene. http://lucene.apache.org/.

[7] Alan R Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[8] David Bamman, Jacob Eisenstein, and Tyler Schnoebelen. Gender in twitter: Styles, stances, and social networks. *CoRR*, abs/1210.4567, 2012.

[9] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 26–33, 2001.

[10] Lee Becker, George Erhart, David Skiba, and Valentine Matula. Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 333–340, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[11] Steven Bedrick, Russell Beckley, Brian Roark, and Richard Sproat. Robust kaomoji detection in twitter. In *Proceedings of the Second Workshop on Language in Social Media*, pages 56–64, Montréal, Canada, June 2012. Association for Computational Linguistics.

[12] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. OReilly Media, 2009.

[13] William Black, Rob Procter, Steven Gray, and Sophia Ananiadou. A data and analysis resource for an experiment in text mining a collection of micro-blogs on a political topic. In *LREC*, 2012.

[14] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *J of Machine Learning Research*, 3(1):993–1022, 2003.

[15] Tabitha Bonilla and Justin Grimmer. Elevated threat levels and decreased expectations: How democracy handles terrorist threats. *Poetics*, 41(6):650 – 669, 2013.

[16] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-driven documents. In *InfoVis*, pages 2301–2309, 2011.

[17] Michael Brooks, Katie Kuksenok, Megan K. Torkildson, Daniel Perry, John J. Robinson, Taylor J. Scott, Ona Anicello, Ariana Zukowski, Paul Harris, and Cecilia R. Aragon. Statistical affect detection in collaborative chat. In *ACM Computer-Supported Cooperative Work (CSCW)*, pages 317–328, 2013.

[18] J. S. Brownstein, C. C. Freifeld, B. Y. Reis, and K. D. Mandl. Surveillance sans frontieres: Internet-based emerging infectious disease intelligence and the healthmap project. *PLoS Medicine*, 2008.

[19] Nan Cao, Jimeng Sun, Yu-Ru Lin, D. Gotz, Shixia Liu, and Huamin Qu. FacetAtlas: Multifaceted visualization for rich text corpora. In *InfoVis*, pages 1172–1181, 2010.

[20] S. K. Card, T. P. Moran, and A. Newell. *The Psychology of Human-Computer Interaction*. Erlbaum, 1983.

[21] Alexandra Carmichael. Crowdsourced Health Confirms Infertility-Asthma Finding, September 2009.

[22] H. A. Carneiro and E. Mylonakis. Google trends: a web-based tool for real-time surveillance of disease outbreaks. *Clinical infectious diseases*, 2009.

[23] K. Chen, A. Kannan, Y. Yano, J. M. Hellerstein, and T. S. Parikh. Shreddr: pipelined paper digitization for low-resource organizations. In *ACM Computing for Development (DEV)*, 2012.

[24] Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. Cascade: Crowdsourcing taxonomy creation. In *ACM Human Factors in Computing Systems (CHI)*, pages 1999–2008, 2013.

[25] Jason Chuang, Sonal Gupta, Christopher D. Manning, and Jeffrey Heer. Topic model diagnostics: Assessing domain relevance via topical alignment. In *ICML*, 2013.

[26] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *AVI*, pages 74–77, 2012.

[27] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. ”without the clutter of unimportant words”: Descriptive keyphrases for text visualization. *ACM Trans. on Computer-Human Interaction*, 19:1–29, 2012.

[28] Jason Chuang, Daniel Ramage, Christopher D. Manning, and Jeffrey Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *CHI*, pages 443–452, 2012.

[29] P.J. Crossno, D.M. Dunlavy, and T.M. Shead. LSAView: A tool for visual exploration of latent semantic modeling. In *VAST*, pages 83–90, 2009.

[30] P. Dai, Mausam, and D. S. Weld. Artificial intelligence for artificial, artificial intelligence. In *AAAI*, 2011.

[31] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In *ACL*, 2013.

[32] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *ACM Computer Supported Cooperative Work (CSCW)*, pages 1013–1022, 2012.

[33] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proc. ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 595–602. ACM Press, 2008.

[34] G. Druck and A. McCallum. Toward interactive training and evaluation. In *Proc. ICKM*, pages 947–956, 2011.

[35] G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 81–90. ACL Press, 2009.

[36] Adnan Duric and Fei Song. Feature selection for sentiment analysis based on content and syntax models. *Decis. Support Syst.*, 53(4):704–711, November 2012.

[37] Jacob Eisenstein. Phonological factors in social media writing. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 11–19, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[38] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. Cueflik: Interactive concept learning in image search. In *ACM Human Factors in Computing Systems (CHI)*, pages 29–38, 2008.

[39] K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar. Posterior Regularization for Structured Latent Variable Models. *Journal of Machine Learning Research*, 2010.

[40] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *In Proceedings of the Twenty-eight International Conference on Machine Learning, ICML*, 2011.

[41] Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. Identifying sarcasm in twitter: A closer look. In *ACL-HLT*, pages 581–586, 2011.

[42] Spence Green, Jeffrey Heer, and Christopher D. Manning. The efficacy of human post-editing for language translation. In *ACM Human Factors in Computing Systems (CHI)*, 2013.

[43] Brynjar Gretarsson, John O'Donovan, Svetlin Bostandjiev, Tobias Höllerer, Arthur Asuncion, David Newman, and Padhraic Smyth. TopicNets: Visual analysis of large text corpora with topic modeling. *ACM Trans on Intelligent Systems and Technology*, 3(2):1–26, 2012.

[44] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *PNAS: Proceedings of the National Academy of Sciences*, 101(1):5228–5235, 2004.

[45] Justin Grimmer and Brandon M Stewart. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3):267–297, 2013.

[46] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009.

[47] David Hall, Daniel Jurafsky, and Christopher D. Manning. Studying the history of ideas using topic models. In *EMNLP*, pages 363–371, 2008.

[48] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[49] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[50] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

[51] Jeffrey Heer and Maneesh Agrawala. Design considerations for collaborative visual analytics. *Information Visualization Journal*, 7:49–62, 2008.

[52] Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *ACM Human Factors in Computing Systems (CHI)*, pages 203–212, 2010.

[53] Jeffrey Heer, Stuart K. Card, and James A. Landay. prefuse: A toolkit for interactive information visualization. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 421–430, New York, NY, USA, 2005. ACM Press.

[54] Jeffrey Heer and Adam Perer. Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. In *IEEE Visual Analytics Science & Technology (VAST)*, 2011.

[55] Jeffrey Heer, Fernanda B. Viégas, and Martin Wattenberg. Voyagers and voyeurs: Supporting asynchronous collaborative information visualization. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 1029–1038, New York, USA, 2007. ACM Press.

[56] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: A tool for visualizing multidimensional geometry. In *IEEE Visualization*, pages 361–378, 1990.

[57] Panos Ipeirotis. Mechanical turk: The demographics, 2008. http://behind-the-enemy-lines.blogspot.com/2008/03/mechanical-turk-demographics.html.

[58] Clement Jonquet, Nigam H Shah, and Mark A Musen. The open biomedical annotator. *Summit on translational bioinformatics*, 2009:56, 2009.

[59] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Wrangler: Interactive visual specification of data transformation scripts. In *ACM Human Factors in Computing Systems (CHI)*, 2011.

[60] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Enterprise data analysis and visualization: An interview study. In *IEEE Visual Analytics Science & Technology (VAST)*, 2012.

[61] Sean Kandel, Ravi Parikh, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Advanced Visual Interfaces*, 2012.

[62] Hyunmo Kang, Lise Getoor, Ben Shneiderman, Mustafa Bilgic, and Louis Licamele. Interactive entity resolution in relational data: A visual analytic tool and its evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):999–1014, September 2008.

[63] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1953–1961, 2011.

[64] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, 2004.

[65] Gary King, Jennifer Pan, and Margaret E. Roberts. How censorship in china allows government criticism but silences collective expression. *American Political Science Review*, 107:1–18, 2013.

[66] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *ACM Human Factors in Computing Systems (CHI)*, pages 453–456, 2008.

[67] Aniket Kittur, Susheel Khamkar, Paul André, and Robert Kraut. Crowdweaver: Visually managing complex crowd work. In *ACM Computer Supported Cooperative Work (CSCW)*, pages 1033–1036, 2012.

[68] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. Crowdforge: Crowdsourcing complex work. In *ACM User Interface Software and Technology (UIST)*, pages 43–52, 2011.

[69] Nicholas Kong, Jeffrey Heer, and Maneesh Agrawala. Perceptual guidelines for creating rectangular treemaps. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2010.

[70] Zornitsa Kozareva, Eduard H. Hovy, and Ellen Riloff. Learning and evaluating the content and structure of a term taxonomy. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 50–57, 2009.

[71] Anand Kulkarni, Matthew Can, and Björn Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *ACM Computer Supported Cooperative Work (CSCW)*, pages 1003–1012, 2012.

[72] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. TurKit: Human computation algorithms on mechanical turk, 2010.

[73] Matthew Lombard, Jennifer Snyder-Duch, and Cheryl Campanella Bracken. Content analysis in mass communication: Assessment and reporting of intercoder reliability. *Human Communication Research*, 28(4):587–604, 2002.

[74] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M. Hellerstein. Distributed GraphLab: A framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.*, 5(8):716–727, April 2012.

[75] Yucheng Low, Joseph Gonzalez, Aapo Kyrola, Danny Bickson, Carlos Guestrin, and Joseph M. Hellerstein. Graphlab: A new parallel framework for machine learning. In *Uncertainty in Artificial Intelligence (UAI)*, July 2010.

[76] Edwin Lughofer. Hybrid active learning for reducing the annotation effort of operators in classification systems. *Pattern Recognition*, 45(2):884–896, 2012. Uses unsupervised methods. Does not incorporate domain expert input.

[77] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Association for Computational Linguistics (ACL)*, 2011.

[78] D. L. MacLean and J. Heer. Identifying medical terms in patient-authored text: a crowdsourcing-based approach. *JAMIA*, 2013.

[79] Patricia Yancey Martin and Barry A. Turner. Grounded theory and organizational research. *The Journal of Applied Behavioral Science*, 22(2):141–157, 1986.

[80] Daniel A. McFarland, Daniel Ramage, Jason Chuang, Jeffrey Heer, Christopher D. Manning, and Dan Jurafsky. Differentiating language usage through topic models. *Poetics*, 41:607–625, 2013.

[81] P. Melville and V. Sindhwani. Active dual supervision: Reducing the cost of annotating examples and features. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 49–57. Association for Computational Linguistics, 2009.

[82] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[83] Thomas Mühlbacher and Harald Piringer. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962–1971, 2013.

[84] Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *WWW*, pages 191–200, 2012.

[85] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[86] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer, 1998.

[87] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *JCDL*, pages 215–224, 2010.

[88] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *WWW*, pages 83–92, 2006.

[89] Fredrik Olsson. *Bootstrapping Named Entity Annotation by means of Active Machine Learning A Method for Creating Corpora*. PhD thesis, University of Gothenburg, 2008.

[90] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. Sics report, Swedish Institute of Computer Science, 2009.

[91] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124, 2005.

[92] Chintan Patel, Kaustubh Supekar, Yugyung Lee, and E. K. Park. Ontokhoj: A semantic web portal for ontology searching, ranking and classification. In *In Proc. 5th ACM Int. Workshop on Web Information and Data Management*, pages 58–61, 2003.

[93] Kayur Patel, Naomi Bancroft, Steven M. Drucker, James Fogarty, Andrew J. Ko, and James Landay. Gestalt: Integrated support for implementation and analysis in machine learning. In *UIST*, pages 37–46, 2010.

[94] Pew Internet & American life project: Health online 2013. `http://pewinternet.org/Reports/2013/Health-online/Summary-of-Findings.aspx`. [Online; accessed 2-April-2013].

[95] Wanda Pratt and Meliha Yetisgen-Yildiz. A study of biomedical concept identification: Metamap vs. people. In *AMIA Annual Symposium Proceedings*, volume 2003, page 529. American Medical Informatics Association, 2003.

[96] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on both features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006. Boosting of features, not prior or regularization.

[97] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-label corpora. In *EMNLP*, pages 248–256, 2009.

[98] Daniel Ramage, Evan Rosen, Jason Chuang, Christopher D. Manning, and Daniel A. McFarland. Topic modeling for the social sciences. In *NIPS Workshop on Applications of Topic Models*, 2009.

[99] Reuters-21578 data set. `http://www.daviddlewis.com/resources/testcollections/reuters21578/`. [Online; accessed 4-January-2014].

[100] E. Ringger, M. Carmen, R. Haertel, K. Seppi, D. Lonsdale, P. McClanahan, J. Carroll, and N. Ellison. Assessing the costs of machine-assisted corpus annotation through a user study. In *Proc. International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association, 2008.

[101] Molly Roberts, Brandon Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Bethany Albertson, Shana Gadarian, and David Rand. Structural topic models for open ended survey responses. *American Journal of Political Science*, Forthcoming.

[102] Jeffrey Rzeszotarski and Aniket Kittur. Crowdscape: Interactively visualizing user behavior and output. In *ACM User Interface Software and Technology*, pages 55–62, 2012.

[103] Tyler Schnoebelen. Do you smile with your nose? stylistic variation in twitter emoticons. *University of Pennsylvania Working Papers in Linguistics*, 18, 2012.

[104] H. Andrew Schwartz, Johannes C. Eichstaedt, Margaret L. Kern, Lukasz Dziurzynski, Stephanie M. Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin E. P. Seligman, and Lyle H. Ungar. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PLoS ONE*, 8(9), 09 2013.

[105] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2634–2643, 2013.

[106] B. Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1467–1478. ACL Press, 2011.

[107] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1069–1078. ACL Press, 2008.

[108] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1289–1296. MIT Press, 2008.

[109] B. Settles and X. Zhu. Behavioral factors in interactive training of text classifiers. In *Proc. North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 563–567. ACL Press, 2012.

[110] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[111] Ben Shneiderman and Catherine Plaisant. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV)*, pages 1–7, 2006.

[112] V. Sindhwani, P. Melville, and R.D. Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *Proc. ICML*, 2009.

[113] R. Snow, B. OConnor, D. Jurafsky, and A. Ng. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263. ACL Press, 2008.

[114] B. Snyder and R. Barzilay. Multiple aspect ranking using the good grief algorithm. In *Proc. HLT-NAACL*, 2007.

[115] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

[116] Stanford corenlp: A suite of core nlp tools. http://nlp.stanford.edu/software/corenlp.shtml.

[117] J. Stasko, C. Görg, Zhicheng Liu, and K. Singhal. Jigsaw: Supporting investigative analysis through interactive visualization. In *VAST*, pages 131–138, 2007.

[118] Philip Stone. Thematic text analysis: new agendas for analyzing text content. In Carl Roberts, editor, *Text Analysis for the Social Sciences*. Lawerence Erlbaum Associates, 1997.

[119] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 70–74, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[120] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *ACM Human Factors in Computing Systems (CHI)*, CHI '09, pages 1283–1292, New York, NY, USA, 2009. ACM.

[121] Edmund M. Talley, David Newman, David Mimno, Bruce W. Herr, Hanna M. Wallach, Gully A. P. C. Burns, A. G. Miriam Leenders, and Andrew McCallum. Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods*, 8(6):443–444, 2011.

[122] Yla R. Tausczik and James W. Pennebaker. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010.

[123] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proc. WWW*, pages 111–120, 2008.

[124] John W. Tukey and Martin B. Wilk. Data analysis and statistics: An expository overview. In Lyle V. Jones, editor, *The Collected Works of John W. Tukey Volume IV: Philosophy and Principles of Data Analysis, 1965-1986*. Wadsworth & Brooks/Cole, 1966.

[125] A. Tumasjan, T.O. Sprenger, P.G. Sandner, and I.M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *ICWSM*, 2010.

[126] Twenty newsgroups data set. `http://qwone.com/~jason/20Newsgroups/`. [Online; accessed 4-January-2014].

[127] L. J. P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[128] Y. Wang, R. E. Kraut, and J. M. Levine. To stay or leave? the relationship of emotional and informational support to commitment in online health support groups. In *ACM Computer-Supported Cooperative Work (CSCW)*, 2012.

[129] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X. Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. TIARA: a visual exploratory text analytic system. In *KDD*, pages 153–162, 2010.

[130] R. W. White, N. P. Tatonetti, N. H. Shah, R. B. Altman, and E. Horvitz. Web-scale pharmacovigilance: listening to signals from the crowd. *JAMIA*, 2013.

[131] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM ’05, 2005.

[132] P. Wicks, T. E. Vaughan, M. P. Massagli, and J. Heywood. Accelerated clinical discovery using self-reported patient data collected online and a patient-matching algorithm. *Nature Biotechnology*, 2011.

[133] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Strategies for crowdsourcing social data analysis. In *ACM Human Factors in Computing Systems (CHI)*, 2012.

[134] Wesley Willett, Jeffrey Heer, Joseph Hellerstein, and Maneesh Agrawala. Commentspace: Structured support for collaborative visual analysis. In *ACM Human Factors in Computing Systems (CHI)*, 2011.

Data Management Plan

We expect this project to generate or collect multiple types of data: source code, text data and associated metadata, derived classifier parameters and classification results, and logged usage data from evaluations. Our plan is to make the source code publicly available to other researchers and practitioners. Text and derived data will be shared only when legally and ethically permissible, in accordance with copyright law, the wishes of third-party data providers and IRB policies. For the purposes of public demonstration, we will use text data available in the public domain.

Data Preservation: All project data and code will be stored securely using facilities provided by the University of Washington's Computer Science & Engineering department. The primary backup system is disk based and utilizes the ZFS file system. The system performs snapshots of data partitions and efficient replication of data to an offsite location on a dedicated 10 Gigabit link. In addition, the department uses the UWIT/eScience hosted tape archive service for catastrophic disaster recovery.

Data Security: All sensitive data – including proprietary text and experimental data – will be kept in centralized storage that is securely maintained by professional staff. Collected experimental data will be anonymized, and stored stripped of personally identifying information.

Data Coordination: Project code and test data will be accessed through Git repositories to facilitate coordination between project members.

Data Dissemination: All code developed as part of this project will be publicly distributed under the terms of a BSD Open Source license. Any scholarly publications for this project will likewise be made publicly available free of charge. Primary distribution will be through the Interactive Data Lab website (`http://idl.cs.washington.edu`) and via GitHub.com, a popular service for open source collaboration. As applicable, we will also place project data into repositories that are maintained by various publishers. For example, papers published in ACM venues will include project data and code as supplemental materials to be archived in the ACM Digital Library.

Post-Doc Mentoring Plan

Post-doctoral scholars supported on this project will work closely with the PI and his collaborators, who have experience mentoring postdocs. Postdocs will gain valuable career experience through this apprenticeship and mentoring which will include the following components.

Recruitment and Orientation. Postdocs will be recruited through an open recruiting process that will include students of under-represented backgrounds. We will discuss mutual expectations for (a) the amount of independence the Postdoctoral Researcher will have, (b) interaction with other team members, (c) productivity including the importance of scientific publications, (d) work habits and (e) documentation of research methodologies and experimental details so that the work can be continued by other researchers in the future.

Stimulating and Supportive Environment. The PI is a member of several interdisciplinary research groups at the University of Washington, including the Interactive Data Lab, the Design-Use-Build (DUB) HCI group, at the eScience Institute. The postdoc will have the opportunity to interact with faculty in all these affiliated groups as well as other groups on the campus. Postdocs at Washington will regularly interact with collaborating faculty at other institutions. This interaction will enhance the postdocs experiences and also benefit the coordination of our project.

Career Counseling. The PI has successfully placed postdocs and students in highly sought after, intellectually challenging jobs, including successful outcomes at both industrial research and faculty positions at peer institutions.

Grant Proposals and Publications. An important part of the mentoring plan is to give the postdoc experience in writing technical publications and grant proposals so they learn to successfully present their research and obtain funding to pursue their own research agenda as independent researchers. The PI will ensure that the postdocs take the lead on writing relevant papers and are involved in grant proposals (including this one!), planning the proposed research, supervising student researchers, considering budgetary and other management issues in the grant writing process, and following responsible professional practices. We have requested funding to send the postdocs to workshops and conferences, both to present research results and to network with other researchers.

Teaching and Mentoring Skills. Postdocs will be involved in teaching graduate courses and possibly also in undergraduate courses (through guest lectures). Several graduate and undergraduate classes taught by the PI are based largely on projects. These classes will provide opportunities for the postdocs to define projects and mentor students working on them. The postdocs will also assist the PI in advising graduate students, particularly those early in their careers, so that she/he receives training in providing new students with problems that best fit their interests and abilities. The PI has already demonstrated initial success in mentoring postdocs, preparing them for an academic or industry career, and broadening their areas of research activity beyond their Ph.D. topics.

Biographical Sketch: Jeffrey Heer

Associate Professor, Computer Science & Engineering
University of Washington
URL: http://jheer.org

PROFESSIONAL PREPARATION:

Jun 2001	**University of California, Berkeley** B.S., Electrical Engineering & Computer Science Honors Program Breadth Area: *Cognitive Science*
Dec 2004	**University of California, Berkeley** M.S., Computer Science
Dec 2008	**University of California, Berkeley** Ph.D., Computer Science Dissertation: *Supporting Asynchronous Collaboration for Interactive Visualization*

APPOINTMENTS:

2013–Present	**University of Washington** Associate Professor, Computer Science & Engineering Department
2012–Present	**Trifacta Inc.** Co-Founder and Chief Experience Officer (CXO)
2009–2013	**Stanford University** Assistant Professor, Computer Science Department

FIVE MOST RELEVANT PUBLICATIONS:

1. Identifying Medical Terms in Patient-Authored Text: A Scalable, Crowdsourcing-Based Approach. Diana MacLean, Jeffrey Heer. *Journal of the American Medical Informatics Association*, 2013.

2. Interpretation & Trust: Designing Model-Driven Visualizations for Text Analysis. Jason Chuang, Daniel Ramage, Chris Manning, Jeffrey Heer. *Proc. ACM Human Factors in Computing Systems (CHI)*, 2012.

3. Termite: Visualization Techniques for Assessing Textual Topic Models, Jason Chuang, Christopher D. Manning, Jeffrey Heer. *Proc. Advanced Visual Interfaces (AVI)*, 2012.

4. Topic Model Diagnostics: Assessing Domain Relevance via Topical Alignment. Jason Chuang, Sonal Gupta, Christopher D. Manning, Jeffrey Heer. *Proc. Intl Conf. on Machine Learning (ICML)*, 2013.

5. D3: Data-Driven Documents. Michael Bostock, Vadim Ogievetsky, Jeffrey Heer. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis'11)*, 2011.

FIVE OTHER PUBLICATIONS:

1. Enterprise Data Analysis and Visualization: An Interview Study. Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, Jeffrey Heer. *Proc. IEEE Visual Analytics Science & Technology (VAST)*, 2012. *Best Paper Honorable Mention*

2. "Without the Clutter of Unimportant Words": Descriptive Keyphrases for Text Visualization. Jason Chuang, Christopher D. Manning, Jeffrey Heer. *ACM Transactions on Computer-Human Interaction*, 19(3), pp. 1-29, 2012.

3. The Efficacy of Human Post-Editing for Language Translation. Spence Green, Jeffrey Heer, Christopher D. Manning. *Proc. ACM Human Factors in Computing Systems (CHI)*, 2013. *Best Paper Award*

4. Strategies for Crowdsourcing Social Data Analysis. Wesley Willett, Jeffrey Heer, Maneesh Agrawala. *Proc. ACM Human Factors in Computing Systems (CHI)*, 2012.

5. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. Jeffrey Heer, Michael Bostock. *Proc. ACM Human Factors in Computing Systems (CHI)*, pp. 203-212, 2010. *Best Paper Nominee*

EDUCATIONAL ACTIVITIES:

1. Visualization Courses. Developed and taught classes on Visualization at Stanford (2009-12) and UC Berkeley (2005-06). Developed special Visualization module for Social Science Methods course (2009-11). Developed new course on Research Topics in Interactive Data Analysis (2011).

2. Human-Computer Interaction Courses. Developed course on Social Software (2010). Re-developed and taught the classes Human-Computer Interaction Research (2009) and Interaction Design Studio (2011-13).

3. External course development. Co-organized half-day course on Visualization and Social Data Analysis at VLDB 2009. Co-organized half-day course on Computation and Journalism at SIGGRAPH 2008. Co-organizing half-day course on Visualization with D3 at InfoVis 2012.

4. Service. Participant in CHIME workshop at ACM CHI 2010 to promote computer science careers for students from historically disadvantaged backgrounds. Advisor in IEEE VisWeek 2010 Doctoral Colloquium.

SYNERGISTIC ACTIVITIES:

1. Developed *Prefuse*, *Flare*, *Protovis* and *D3.js* visualization tools, used across academia and industry by thousands of developers and millions of end users. D3.js is the 5th most "starred" project on GitHub.com.

2. Workshop organizer for Social Data Analysis workshops at ACM CHI 2008 and ACM CSCW 2010, perception workshop at CHI 2013, and workshop on language learning and visualization at ACL 2014.

3. Invited speaker to discuss recent trends in visualization and data analysis at the Conference on Innovative Data Systems Research (CIDR) 2009, ACM SIGMOD 2009, IBM NPUC 2010, HCIC 2010, Microsoft Faculty Summit 2010, NICAR 2011, The Economist Ideas Economy 2011, ASA Joint Statistical Meeting 2011, WikiSym 2011, DataEdge 2013, HCIC 2013 and Gordon Research Conference 2013.

COLLABORATORS:

Maneesh Agrawala (Berkeley), Cecilia Aragon (Washington), Magda Balazinska (Washington), Serafim Batzoglou (Stanford), Jeff Baumes (Kitware), Gill Bejerano (Stanford), Michael Bernstein (Stanford), Atul Butte (Stanford), Stuart Card (Stanford), Douglas Carmean (Intel), Bill Cleveland (Purdue), Jean-Daniel Fekete (INRIA), Li Fei-Fei (Stanford), James Fogarty (Washington), Emily Fox (Washington), Carlos Guestrin (Washington), Sonal Gupta (Stanford), Spence Green (Stanford), Pat Hanrahan (Stanford), Marti Hearst (Berkeley), Joseph Hellerstein (Berkeley), Bill Howe (Washington), Amy Jang (Google), Ashley Jin (Stanford), Dan Jurafsky (Stanford), Jessie Kennedy (Edinburgh Napier University), Scott Klemmer (UCSD), Monica Lam (Stanford), James Landay (Cornell Tech), Anna Lembke (Stanford), Jure Leskovec (Stanford), Jock Mackinlay (Tableau), Chris Manning (Stanford), Dan McFarland (Stanford), Miriah Meyer (Utah), Margaret Morris (Intel), Andreas Paepcke (Stanford), Adam Perer (IBM), Hanspeter Pfister (Harvard), Catherine Plaisant (U. Maryland), Dan Ramage (Google), Nathalie Riche (Microsoft), Will Schroeder (Kitware), Ben Shneiderman (U. Maryland), Arend Sidow (Stanford), John Stasko (Georgia Tech), Chris Stolte (Tableau), Maureen Stone (Tableau), Mike Stonebraker (MIT), Frank van Ham (IBM), Fernanda Viégas (Google), Martin Wattenberg (Google), Chris Weaver (Univ. of Oklahoma), Wesley Willett (INRIA), John D. Wilkerson (Washington), Terry Winograd (Stanford)

ADVISORS: Maneesh Agrawala (PhD, Berkeley), James A. Landay (MS, Berkeley)

DOCTORAL & POST-DOCTORAL ADVISEES:

Michael Bostock (Stanford PhD), Jason Chuang (Stanford PhD, UW Post-Doc), Cagatay Demiralp (Stanford Post-Doc), Sanjay Kairam (Stanford PhD), Sean Kandel (Stanford PhD), Zhicheng "Leo" Liu (Stanford Post-Doc), Diana MacLean (Stanford PhD), Arvind Satyanarayan (Stanford PhD), Jeffrey Snyder (UW PhD), Kanit Wongsuphasawat (UW PhD)

Biographical Sketch: Jason Chuang

Post-Doctoral Researcher, Computer Science & Engineering
University of Washington
URL: http://jason.chuang.ca

PROFESSIONAL PREPARATION:

May 2002 **University of British Columbia**
Bachelor of Science in Mathematics

June 2005 **Stanford University**
Master of Science in Scientific Computing and Computational Mathematics

April 2013 **Stanford University**
Doctor of Philosophy in Computer Science
Dissertation Topic: *Designing Visual Text Analysis Methods to Support Sensemaking and Modeling*

APPOINTMENTS:

2013–Present **University of Washington**
Post-Doctoral Researcher, Computer Science & Engineering

FIVE MOST RELEVANT PUBLICATIONS:

1. "Topic Model Diagnostics: Assessing Domain Relevance via Topical Alignment" by Jason Chuang, Sonal Gupta, Christopher D. Manning, and Jeffrey Heer. *Proc. International Conference on Machine Learning (ICML). Atlanta, Georgia, 2013.*

2. "Recursive Models for Semantic Compositionality Over a Sentiment Treebank" by Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. *Proc. Conference on Empirical Methods in National Language Processing (EMNLP). Seattle, Washington, 2013.*

3. "Interpretation and Trust: Designing Model-Driven Visualizations for Text Analysis" by Jason Chuang, Daniel Ramage, Christopher D. Manning, and Jeffrey Heer. *Proc. ACM Conference on Human Factors in Computing Systems (CHI). Austin, Texas, 2012.*

4. "Termite: Visualization Techniques for Assessing Textual Topic Models" by Jason Chuang, Christopher D. Manning, and Jeffrey Heer. *Proc. International Working Conference on Advanced Visual Interfaces (AVI). Capri Island, Italy, 2012.*

5. "'Without the Clutter of Unimportant Words': Descriptive Keyphrases for Text Visualization" by Jason Chuang, Christopher D. Manning, and Jeffrey Heer. *ACM Transactions on Computer-Human Interaction (TOCHI), 19 (3), pp. 1-29, October 2012.*

FIVE OTHER PUBLICATIONS:

1. "Differentiating Language Usage through Topic Models" by Daniel A. McFarland, Daniel Ramage, Jason Chuang, Jeffrey Heer, Christopher D. Manning, and Daniel Jurafsky. *Poetics: Special Issue on Topic Models and the Cultural Sciences, 41 (6). December 2013.*

2. "A Probabilistic Model of the Categorical Association between Colors" by Jason Chuang, Maureen Stone, and Pat Hanrahan. *Proc. Color Imaging Conference (CIC). Portland, Oregon, 2008.*

3. "RNA Sequencing Reveals Diverse and Dynamic Repertoire of the Xenopus Tropicalis Transcriptome Over Development" by Meng How Tan, Kin Fai Au, Arielle L. Yablonovitch, Andrea E. Wills, Jason Chuang, Julie C. Baker, Wing Hung Wong, and Jin Billy Li. *Genome Research, 23 (1), pp. 201-216. January 2013.*

4. "Document Exploration with Topic Modeling: Designing Interactive Visualizations to Support Effective Analysis Workflows" by Jason Chuang, Yuening Hu, Ashley Jin, John D. Wilkerson, Daniel A. McFarland, Christopher D. Manning, and Jeffrey Heer. *NIPS Workshop on Topic Models. Lake Tahoe, Nevada, 2013.*

5. "Topic Modeling for the Social Sciences" by Daniel Ramage, Evan Rosen, Jason Chuang, Christopher D. Manning, and Daniel A. McFarland. *NIPS Workshop on Applications for Topic Models. Vancouver, Canada, 2009.*

EDUCATIONAL ACTIVITIES:

1. Developed and taught a unit on text analysis and visualization at the Stanford Computational Social Science Workshop (Summer 2013)

2. Re-developed and taught lectures on color and text visualizations for the Stanford Visualization Class (Winter 2009, Fall 2011).

SYNERGISTIC ACTIVITIES:

1. Co-organizing a full-day workshop on Interactive Language Learning, Visualization, and Interfaces at ACL 2014

COLLABORATORS:

Jeffrey Heer (Stanford), Christopher D. Manning (Stanford), Daniel A. McFarland (Stanford), John D. Wilkerson (Univ. of Washington), Cecilia Aragon (Univ. of Washington), Pat Hanrahan (Stanford), Maureen Stone (StoneSoup Consulting), Daniel Jurafsky (Stanford), Andrew Y. Ng (Stanford), Christopher Potts (Stanford), Julie C. Baker (Stanford), Wing Hung Wong (Stanford), Jin Billy Li (Stanford), Geoffrey J. Gordon (CMU), Robert Rohling (Univ. of British Columbia), Septimiu E. Salcudean (Univ. of British Columbia), Martin Wattenberg (Google), Fernanda Viégas (Google), Holger Winnemöller (Adobe), Gary Bradski (Intel), Daniel Ramage (Stanford), Sonal Gupta (Stanford), Spence Green (Stanford), Mengqiu Wang (Stanford), Katie Kuksenok (Univ. of Washington), Richard Socher (Stanford), Alex Perelygin (Stanford), Jean Y. Wu (Stanford), Rebecca Weiss (Stanford), Ashley Jin (Stanford), Evan Rosen (Stanford), Meng How Tan (Stanford), Kin Fai Au (Stanford), Arielle L. Yablonovitch (Stanford), Andrea E. Wills (Stanford), Chih-Han Yu (Stanford), Brian Gerkey (Stanford), Stephen Okazawa (Univ. of British Columbia), Richelle Ebrahimi (Univ. of British Columbia), Yuening Hu (Univ. of Maryland, College Park)

PH.D. ADVISOR:

Jeffrey Heer (Stanford) and Christopher D. Manning (Stanford)

Facilities, Equipment, and Other Resources

University of Washington Computer Science and Engineering Department

General Resources

The Department maintains a wide variety of state-of-the-art computing facilities for research and instructional use, housed in the Paul G. Allen Center for Computer Science & Engineering. The Computer Science Laboratory coordinates the acquisition, maintenance, and operation of the computing equipment and network services. General-purpose research computing is provided by over 900 Windows and Unix-based workstations and servers, located in laboratories, machine rooms and offices. The back-end infrastructure is comprised of general-purpose compute, file, web, mail and print servers, operating as a well-integrated Linux and Windows 7 environment. In addition, around a dozen compute clusters are used by a range of research projects. Departmental networking utilizes 1 and 10 gigabit Ethernet connections to servers and desktop machines, and a dual-band wireless network provides 802.11b/g/n connectivity throughout the building and in surrounding exterior areas. Several large plasma screens and a 56" HDTV provide high-definition video display for networking and graphics research and for video conferencing.

Research Resources

Research in computer systems (including architecture, networking, operating systems, and distributed systems) involves a wide and constantly updated variety of hardware, software, and networks. Current hardware includes high-performance Intel multicore platforms, a 200-node Intel cluster with several tens of terabytes of networked storage, a networking testbed cluster, and PC workstations. Our facilities include Linux, FreeBSD, and Windows systems, and our clusters enjoy 1 and 10 gigabit switched Ethernet connectivity and an Abilene network feed. In addition, the Systems lab provides a common workspace for operating systems, networking, and architecture students, and features Windows workstations, a video projector, and floor-to-ceiling whiteboards.

Research in VLSI, digital hardware, and embedded systems is supported by a set of PC workstations and multiprocessor compute servers. A large collection of both commercial and university computer-aided design tools form the core of the design environment providing capabilities for the design of CMOS VLSI chips, FPGA and microprocessor-based systems, and printed-circuit boards. A variety of specialized equipment for the prototyping, debugging, and testing of microelectronic systems is also available and is housed within the Hardware and Embedded Systems Research Laboratory. These resources are utilized by research projects involved in the design of configurable computing architectures, devices to support ubiquitous and invisible computing, embedded systems, neurally-inspired computing and learning devices. Additional equipment and facilities are available in the W.T. Baxter Computer Engineering Laboratory, which is used for graduate and undergraduate courses including VLSI and embedded system design.

Research in graphics, image processing, and user interfaces, centered in the Graphics and Imaging Laboratories, utilizes a set of high-end graphics workstations, a multiprocessor compute server, and a variety of special-purpose devices, including a real-time motion capture system, digital cameras (still and video), a computer-controlled lighting grid, a desktop Cyberware 3D laser scanner, video projectors for shape capture, and rotational and translational motion control platforms. Most of the lighting and imaging hardware resides in a special-purpose scanning and imaging laboratory, which is ideal for experiments that require controlled illumination. The motion capture system resides in a large studio with ample space to capture running, walking, and jumping motions. The main lab spaces contain an array of workstations and an audio/video hardware suite with non-linear digital video editing capabilities. The workstations in the main labs are also used as development stations for experimental teaching software in graphics and vision.

Research in robotics is carried out in the Robotics and State Estimation Laboratory, which is equipped with several mobile robots, including one RWI B21r robot, three ActiveMedia Pioneer robots, nine ActiveMediaAmigoBots, and nine Sony AIBO robots. All robots utilize wireless networking to communicate with each other and the lab PCs running Linux. The B21 robot and all three Pioneer robots are equipped with SICK laser range-finders.

Research in data management is supported by a combination of laptops, desktops, and a machine-cluster all running a suite of software systems. The current hardware configuration for the cluster includes over 50 high-performance, Intel multicore servers with several tens of terabytes of storage and hundreds of gigabytes of RAM in total. The machines are configured with either Windows or Linux and run several state-of-the art database management systems including SQL Server, Oracle, DB2, and Hadoop. In addition, the Database lab provides a common workspace for students, and features Windows and Linux workstations, a video projector, and floor-to-ceiling whiteboards.

Many other research groups utilize equipment located in a set of research laboratories, plus about a dozen compute clusters with a total of around 2400 cores. Additional information can be found in the web pages for individual research projects, at `http://cs.washington.edu/research`.

Instructional Resources

Instructional computing is provided through laboratories and back-end services operated within the department. These include three general use laboratories with 75 Intel-based PCs running Windows 7 and Linux. Additional back-end resources are provided by Intel-based compute, web, database, and file servers, in an integrated Linux/Windows infrastructure.

The department also operates four special-purpose laboratories containing approximately 100 Intel Pentium PCs. To support digital system design courses, the Baxter Computer Engineering Laboratory and the Embedded Systems Project Laboratory, with over fifty Pentium workstations for design entry and simulation along with Tektronix logic analyzers, digital oscilloscopes and other test equipment. Capstone courses utilize the Capstone Computing Lab, containing 11 Intel quad-core workstations, and often specialized equipment to fit the needs of the course. The Laboratory for Animation Arts includes 22 Intel PCs and digital video production equipment, and is used for teaching interdisciplinary courses in computer animation. The Special Projects Lab contains 20 Intel quad-core workstations, and is used to teach capstone courses in operating systems and other courses requiring specialized equipment or dedicated access. The SPL runs different systems and software at different times, depending on course needs.