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

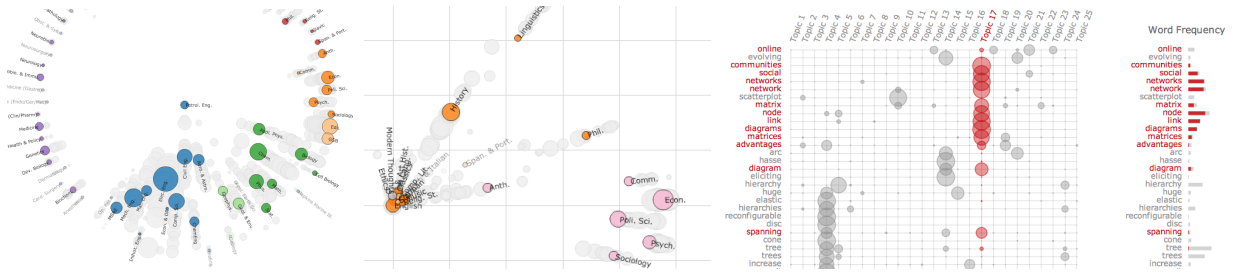

Figure 1: Visual text analysis of academic publications. (a) Left: Similarity between Stanford departments based on published theses. Petroleum Engineering is centered; radial distances convey textual similarity to the other departments. (b) Center: Departments viewed using LDA topic similarity, focused 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

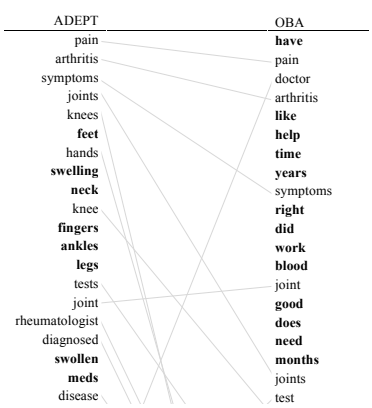

Figure 2: Comparison of terms identified as medically-relevant by different models. (a) Left: comparison of five models (classified terms shown in black), including our CRF-based ADEPT model. OBA and MetaMap runs use the SNOMED CT ontology. (b) Right: Term rankings for ADEPT and OBA on Arthritis forum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

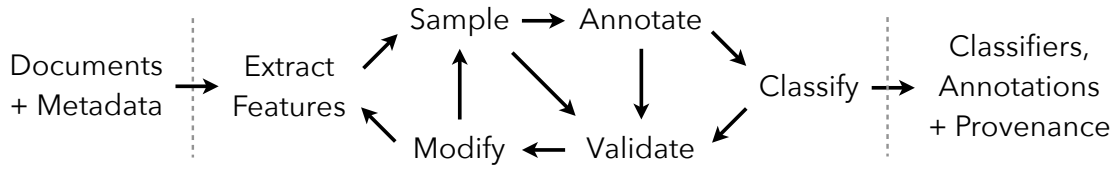

Figure 3: Proposed interactive text analysis workflow.

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.

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 crowdsourced 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.