ABSTRACT

Title of dissertation: DESIGNING FOR THE HUMAN IN THE LOOP: TRANSPARENCY AND CONTROL IN INTERACTIVE MACHINE LEARNING

Alison Renner
Doctor of Philosophy, 2020

Dissertation directed by: Dr. Jordan Boyd-Graber
Department of Computer Science

Interactive machine learning techniques inject domain expertise to improve or adapt models. Prior research has focused on adapting underlying algorithms and optimizing system performance, which comes at the expense of user experience. This dissertation advances our understanding of how to design for *human-machine collaboration*—improving both user experience *and* system performance—through four studies of end users' experience, perceptions, and behaviors with interactive machine learning systems. In particular, we focus on two critical aspects of interactive machine learning: how systems explain themselves to users (transparency) and how users provide feedback or guide systems (control).

We first explored how explanations shape users' experience of a simple text classifier with or without the ability to provide feedback to it. Users were frustrated when given explanations without means for feedback and expected model improvement over time even in the absence of feedback.

To explore transparency and control in the context of more complex models and subjective tasks, we chose an unsupervised machine learning case, topic modeling. First, we developed a novel topic visualization technique and compared it against common topic representations (e.g., word lists) for interpretability. While users quickly understood topics with simple word lists, our visu-

alization exposed phrases that other representations obscured.

Next, we developed a novel, "human-centered" interactive topic modeling system supporting users' desired control mechanisms. A formative user study with this system identified two aspects of control exposed by transparency: adherence, or whether models incorporate user feedback as expected, and stability, or whether other unexpected model updates occur.

Finally, we further studied adherence and stability by comparing user experience across three interactive topic modeling approaches. These approaches incorporate input differently, resulting in varied adherence, stability, and update speeds. Participants disliked slow updates most, followed by lack of adherence. Instability was polarizing: some participants liked it when it surfaced interesting information, while others did not. Across modeling approaches, participants differed only in whether they noticed adherence.

This dissertation contributes to our understanding of how end users comprehend and interact with machine learning models and provides guidelines for designing systems for the "human in the loop."

DESIGNING FOR THE HUMAN IN THE LOOP: TRANSPARENCY AND CONTROL IN INTERACTIVE MACHINE LEARNING

by

Alison Renner

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2020

Advisory Committee:
Dr. Jordan Boyd-Graber, Chair/Advisor
Dr. Leah Findlater, Co-Advisor
Dr. Mihai Pop
Dr. Hernisa Kacorri
Dr. Naomi Feldman

Dedication

To my husband for always making me laugh.

Acknowledgements

I first want to thank my amazing advisors, Leah Findlater and Jordan Boyd-Graber for their guidance and tremendous support for too many years to count. Thanks to Leah for fostering (and reigning in) my excitement and for always knowing which stats test to run and to Jordan for his endless wisdom and writing standards. And, thanks to you both for compromising on latex v. word and navigating remote, cross-time zone research relationships from Boulder, Seattle, Zürich, and my near refusal to travel to campus around rush hour (i.e., any weekday). This dissertation would not have been possible without both of you.

I'm extremely grateful to the brilliant group of people I've been privileged to work with over the years. Specifically, Kevin Seppi for his invaluable contributions and advice, Varun Kumar for his software development and algorithm expertise and the continued support even after finishing his degree, Jim Nolan for the unwavering support in both my research and professional career, and to Simone Stumpf for pulling me into the explainability world and for all of the brainstorming sessions along the way. And, of course, special thanks to my advisory committee, Mihai Pop, Hernisa Kacorri, and Naomi Feldman, for their helpful feedback and kind words.

Many thanks as well to all my collaborators, colleagues and friends in my extended research groups at the University of Maryland, University of Washington, Decisive Analytics Corporation, and beyond. Whether for the thoughtful discussions, practical contributions, or general words of encouragement, I especially want to thank Sherry Wu, Dan Weld, Brian Lim, Tak Yeon Lee, Advait Sarkar, Ben Shneiderman, Wilson Fern, Ron Fan, Melissa Birchfield, Thad Goodwyn, Rob Rua, Mike Colony, Tim Hawes, Peter David, Yuening Hu, Forough Poursabzi-Sangdeh, Niklas Elmqvist, and Sana Malik.

I could not have come this far without the love and relentless support of my family and friends.

Thanks especially to my Mom and Mike for your encouragement without asking for too many progress updates, to Ellie and George for the good genes and my undergraduate degree, to Dan and Kayla for fostering my competitiveness, to my Dad for introducing me to Math and video games, and to my new family (Lambie, the OG Dr. Renner, Ali, GP, Joey, Jessica, and Robby) for welcoming me in and the much-needed hiking trips. And, of course, many thanks to my running buddies for keeping me in shape and my amazing group of friends for keeping me sane (ooh la la, you know who you are).

Finally, I am forever grateful to my incredible husband, teammate, and best friend, Steve; thank you for your constant patience, encouragement, and always reminding me to eat.

This work was supported in part by the National Science Foundation under award IIS-1409287 and by Decisive Analytics Corporation.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Machine Learning (ML) is common in today's data-rich society. ML algorithms determine which news headlines and advertisements we see, suggest movies for us to watch, and estimate home sale prices when we consider moving. These techniques are also used in more critical settings, such as medical decision support tools, home security, and autonomous vehicles. ML techniques build models of data and can be *supervised*, meaning they learn from labeled training data, or *unsupervised*, meaning they learn to find patterns when labels are not provided.

End user involvement is necessary with ML: users evaluate models, provide training data, determine whether to listen to system's recommendations or decisions, or adjust models through implicit or explicit feedback. However, ML systems are primarily designed from an "algorithm-centric" view, optimizing system performance at the expense of user experience, which results in systems that are not accessible to the general public and design guidelines that do not faithfully represent general user perceptions and experience.

While ML-based tools are typically geared toward algorithm developers, or ML experts, making tools accessible for non-ML experts will allow more people to understand the capabilities and limitations of ML, to not only take advantage of ML, but to do so responsibly.

This dissertation advances our understanding of how to design for the non-ML expert end user, or the "human in the loop," through studies of end users' experience, perceptions, and behaviors

with ML systems. Here, the goal is to improve both user experience *and* system performance. In particular, we focus on two critical aspects of ML: how systems exposes or explain themselves to users (*transparency*) and how users provide feedback or guide systems (*control*).

1.1 Motivation

While ML models may have high accuracy on held-out test sets or demonstrate utility on a few cases, users need to know *how* they are working. If models are right for the right reasons, users can be more confident that they will generalize and are operating without bias (Dodge et al., 2019). System *transparency* or automatic model explanations—such as "why" and "why not" justifications (Lim et al., 2009) and feature visualizations (Kulesza et al., 2015)—can provide intuition and increase user confidence and trust (Bunt et al., 2007; Pu and Chen, 2006), human task performance (Feng and Boyd-Graber, 2019; Schmidt and Biessmann, 2019; Stowers et al., 2017), satisfaction (Biran and McKeown, 2017), and system acceptance (Herlocker et al., 2000). Ongoing government research programs (Gunning, 2016), focused academic conferences,[1] and recent legislation on the "right to explanation" (Goodman and Flaxman, 2017) have also fueled a general push for ML transparency. However, how to best expose systems' inner workings or explain their decisions is not a solved problem, particularly in unsupervised ML settings, which lack training labels and human-understandable features for shared communication.

Additionally, transparency is not an unmitigated good. Complex explanations may promote over-reliance when they are convincing (Stumpf, 2016) or lower user satisfaction when they are confusing (Narayanan et al., 2018). Exposing system uncertainty or algorithmic limitations may negatively affect users' perceptions (Cai et al., 2019; Lim and Dey, 2011; Stowers et al., 2017),

[1] ACM Conference on Fairness, Accountability, and Transparency (https://fatconference.org/))

and users may ignore explanations entirely if the benefit to attending to them is unclear (Kulesza et al., 2013).

A particular complication of transparency is that it highlights model deficiencies, which are common, because ML models are rarely perfect: data are noisy, models are limited, and humans' needs and understanding sometimes conflict with ML output (Amodei et al., 2016). In these cases, a human–machine collaboration is required to iteratively improve and adapt models. Users can *control*, or interactively improve, models by providing input such as additional training labels (Settles, 2010), weak supervision (Ratner et al., 2017), re-weighting features, or modifying the underlying data representation (Andrzejewski et al., 2009; Vaughan, 2018). Such approaches fall under the umbrella of Interactive Machine Learning (IML), which are methods that support users in iteratively improving or adapting ML models.

Transparency is beneficial in IML settings: users who understand models better can also better correct models' mistakes (Kulesza et al., 2015; Rosenthal and Dey, 2010). However, we hypothesize that increased transparency has another effect in IML: when users provide input to transparent models, they can see what the models do with their input, how they update, and whether their input is incorporated as they expect. Therefore, with transparent models we cannot simply provide users with control mechanisms and expect for a positive outcome—we must also consider *how* models update and what cascading side effects might occur. This need introduces a problematic tension: models must balance respecting user inputs and faithfully modeling the data.

Finally, IML systems must be designed with end user needs in mind and not simply based on what algorithm developers *think* users want or what is best for system performance. Thus, this dissertation takes a human-centered approach to design and evaluation of IML, focusing on *control* and *transparency*, in particular.

1.2 Goals

The goals of this dissertation are to determine effective mechanisms for control and transparency in IML and to provide a better understanding of how these constructs affect end users' experience, perceptions, and behavior, in supervised and unsupervised ML settings. This dissertation broadly covers two primary IML research areas: (1) desired methods for understanding and interacting with ML and (2) effects of transparency and control on users' experience with ML.

1.3 Approach and Overview

To satisfy the goals of this dissertation, we took a human-centered approach to control and transparency (and their interaction), with a focus on both user experience and system performance. In particular, we explored how end users were affected when transparent systems did not adhere to user input or even support it, or when feedback was requested without an adequate explanation, and how users could best understand and interact with complex, unsupervised models, such as topics. We additionally developed interpretable and interactive machine learning systems and distilled design guidelines for supporting the human in the loop. For this dissertation, we focused on two specific cases of IML: a supervised ML technique—interactive text classification—and an unsupervised one—interactive topic modeling.

1.3.1 Interactions of Explanations and Feedback in Supervised ML

We first built on prior findings that transparency increases users' understanding of how ML models work and the errors they make (Kulesza et al., 2013; Lim et al., 2009) and explored whether this insight in turn increased users' desires to "fix" those errors, and therefore reduced satisfaction

if they were unable to do so (Chapter 4). In particular, we investigated how explanations shape users' perceptions of ML models with or without the ability to provide feedback to them: (1) does revealing model flaws increase users' desire to "fix" them; (2) does providing explanations cause users to believe—wrongly—that models are introspective, and will thus improve over time?

We performed two controlled experiments—varying model quality—of users interacting with a simple, supervised ML system (interactive text classification). Participants reviewed predictions made by the classification model with or without explanations (highlighting important words) and with one of three levels of user feedback to the model: none, instance-level (correcting or confirming the model's prediction), and feature-level (telling the model *how* to predict). We showed how the combination of explanations and user feedback impacts perceptions, such as frustration and expectations of model improvement, and feedback quality. Of particular importance to the remainder of this dissertation, we demonstrated that, when possible, explanations should be paired with feedback: explanations without opportunity for feedback reduced satisfaction with a lower accuracy model, and requesting detailed feedback without explanation reduced satisfaction in a higher accuracy model. Additionally users expected model correction, regardless of whether they provided feedback or received explanations.

1.3.2 Visualizations for Topic Interpretability

The controlled experiments presented in Chapter 4 provided insights into users' experience given varied feedback mechanisms (control) and explanations (transparency) for a simple, supervised ML case. We were also interested in exploring control and transparency in more detail and under different settings, such as more subjective tasks and complex models; therefore, we switched our focus to unsupervised ML. In particular, we chose unsupervised topic modeling, which is a

common technique for organizing and understanding large text corpora by the themes, or topics (i.e., sets of words), they discuss. However, promoting end-user understanding of topics remains an open research problem.

To address this, we developed a novel topic explanation technique and performed a comparative evaluation of topic representations for interpretability of unsupervised topic models (Chapter 5). For this study, we considered whether users could quickly and confidently label topics as a measure of their interpretability. In particular, we examined (1) which topic representations (e.g., word lists, word lists with bars, word clouds, etc.) are quickly, confidently, and correctly interpreted, and (2) how do human-generated labels for topics compare to labels generated using automatic labelling methods? Simple visualizations allowed participants to quickly understand topics, while our new, more complex visualization took longer but exposed multi-word expressions that simpler visualizations obscured. Automatic topic labeling techniques also far under-performed human-generated labels.

1.3.3 User Experience with Transparent and Interactive Systems

While the controlled experiments presented in Chapter 4 explored control in terms of whether or how users provided feedback, they did not consider users' reactions when their feedback was not applied *predictably*, or as expected—a case that may be exposed through system transparency, as controls are easier to validate. To address this, we performed two studies to examine users' experience and perceptions when they provide input to systems and observe how they update and how their inputs are incorporated. Specifically, we used interactive topic modeling. The goals of these studies were to (1) determine how users want to control (or *refine*) topic models in real world settings, (2) understand users' perceptions regarding system characteristics, such as latency,

unpredictability, and quality, and (3) determine which characteristics of interactive topic modeling systems users find most and least frustrating.

To support these research goals, we implemented a novel interactive topic modeling system. We designed this system to include the interpretable explanations identified in Chapter 5 and the control mechanisms desired by users in our prior work (Lee et al., 2017).[2] In Chapter 6, we first conducted a formative, exploratory study with twelve participants to explore users' trust and perceptions of system characteristics (e.g., latency, unpredictability, and model complexity). Although users experienced unpredictability, their reactions varied from positive to negative, and surprisingly, overall users trusted our system and in some cases perhaps trusted it too much or had too little confidence in themselves. This formative study also identified two specific aspects of control exposed by transparency: adherence, or whether models incorporate user feedback as expected, and stability, or whether other unexpected model updates occur.

We then built on the findings of the formative study to explore users' perceptions and reactions to *predictable control* (i.e., adherence and stability) in more detail. To do so, we chose three distinct interactive topic modeling approaches, which differ in how user input is incorporated, resulting in varied system characteristics: adherence, stability, update speeds, and model quality. We conducted a comparative study where 100 participants performed a document organization task with one of three topic modeling approaches (Chapter 7), and we asked participants whether they noticed the different system characteristics and which they liked the most and least. Participants disliked slow updates most, followed by lack of adherence. Instability was polarizing: some participants liked it when it surfaced interesting information, while others did not. Across modeling approaches, participants differed only in whether they noticed adherence.

[2]This prior work, of which the author was a primary contributor, is not included in this dissertation.

1.4 Summary of Contributions

This dissertation makes the following primary contributions to our understanding of end users' desires, perceptions, and experience regarding control and transparency in IML, which cover two broad research areas:

1. **Desired methods for understanding and interacting with ML:**

   (a) A controlled experiment showing that users are more satisfied and provide better quality feedback given particular feedback mechanisms for an interactive text classification tool (Chapter 4).

   (b) A novel topic visualization, which highlights phrase relationships in topics (Chapter 5).

   (c) A controlled experiment showing that users can quickly and easily understand topics with a simple word list visualization and that human-generated topic labels far outperform automatically generated ones.

   (d) A novel interactive topic modeling system and two studies evaluating users' experience and performance when using the system for a document organization task (Chapters 6 and 7).

   (e) User-centered design principles for transparent, interactive systems; in particular, interactive topic modeling systems (Chapters 6 and 7).

2. **Effects of transparency and control on users' experience with ML:**

   (a) A controlled experiment showing that, for a simple model and task, users want the opportunity to provide feedback, regardless of model quality or whether they received

explanations. And, that explanations without the opportunity for feedback result in an especially negative user experience (Chapter 4).

(b) An exploratory study exposing how IML system characteristics such as adherence, instability, latency, and performance affect users' experience and usage of an interactive topic modeling system (Chapter 6).

(c) A comparative study exploring users' perceptions and likes and dislikes of IML systems' characteristics, specifically finding that users disliked long wait times followed by lack of adherence, and that only perceptions of adherence differed between the systems (Chapter 7).

1.5 Organization

In Chapter 2, we review user experience with IML. This includes an overview of IML, as well as IML system characteristics that are important to this research: transparency, control, predictability, and latency. Chapter 3 then covers background on the two specific IML techniques evaluated in this dissertation: interactive text classifiers and interactive topic models. Chapter 4 presents a study of how users' satisfaction and feedback quality is affected given varied combinations of explanations and feedback in a simple, supervised ML system (i.e., interactive text classification). Chapters 5–7 switch the focus to unsupervised ML, in particular topic models, to explore user experience and system performance for more complex models and subjective tasks. Chapter 5 presents a novel visualization technique for topics and compares it to common topic representations for interpretability. Chapter 6 introduces a new interactive topic modeling system and presents a formative study exploring end users' experience and perceptions of exposed system characteristics, such as unpredictability, latency, and quality. This work identifies two particular

aspects of control exposed by transparency: adherence and instability. Chapter 7 builds on the formative study to explore adherence and instability in more detail by comparing users' experience and perceptions of our interactive topic modeling system backed by three distinct modeling approaches, which vary in terms of adherence to input, model stability, latency, and quality. Finally, Chapter 8 concludes with a discussion of the research presented in this dissertation, summarizes design guidelines, and suggests opportunities for future research.

Chapter 2: Background on User Experience with Interactive Machine Learning

This chapter summarizes related work in Interactive Machine Learning (IML). First we describe the goals of IML and how it is applied in the ML space. We then discuss particular IML system characteristics, such as transparency, control, predictability, and latency, and their interactions: how transparency affects feedback quality and how transparency exposes unpredictability related to control. Throughout this chapter, we review IML techniques from the perspective of the human in the loop, focusing on how users interact and how models are exposed in these settings, but not on algorithm details. We discuss specific IML techniques and algorithm details in Chapter 3.

2.1 Interactive Machine Learning

Fails and Olsen (2003) were the first to use the term "interactive machine learning" when introducing their Crayons system, in which users provide interactive feedback to improve a pixel classifier. Compared to classical machine learning, "interactive training allows the classifier to be coached along until the desired results are met" (Fails and Olsen, 2003). Where classical ML focuses on static, pre-defined labels (or classes) and datasets, in IML a model is trained through rapid end-user interaction (Amershi et al., 2014). IML covers interaction at any stage, from algorithm developers iteratively training models for a downstream task to non-ML expert end users

providing domain expertise to improve or adapt systems. This dissertation focuses on end users who are not ML experts, as our goal is to better understand and improve experience with ML for the general population.

IML produces higher quality (Raghavan et al., 2006; Settles, 2010) or personalized models (Amershi et al., 2012; Głowacka et al., 2013) or models that are better aligned with users' understanding of the data or the domain (Andrzejewski et al., 2009; Lee et al., 2012). However, user interaction can have negative effects, such as decreased system performance (Ahn et al., 2007; Wu et al., 2019) or inconsistent mental models (Bansal et al., 2019).

The interactive or "human-in-the-loop" approach has been applied across the ML space, which can be broadly categorized by three types of techniques: supervised ML, unsupervised ML, and reinforcement learning (RL). Supervised ML algorithms operate on large amounts of initial labeled training data (e.g., residential properties labeled with their price or emails labeled as whether or not they are spam) and learn functions to map from sample input data to output labels. Common examples of supervised ML are spam detection, information retrieval, and image classification. Typically these algorithms learn "features" of the inputs that map to the associated labels. For example, residential properties with higher values for certain features, such as acreage or number of rooms likely have higher prices. Similarly, spam emails contain particular words or phrases (features) more often than non-spam emails. The supervised paradigm provides intuitive mechanisms for user feedback, such as providing additional training examples (Fiebrink et al., 2009) or by reacting to model predictions with instance-level (i.e., correcting or confirming predictions (Culotta et al., 2006; Fails and Olsen, 2003)) or feature-level feedback (i.e., denoting features indicative of each class (Kulesza et al., 2015; Raghavan et al., 2006; Settles, 2011)).

Unsupervised ML lacks initial labeled training data; instead algorithms learn to organize or cate-

gorize data based on similarities or patterns. Unsupervised ML is frequently used for clustering, such as organizing large datasets into similar groups or for anomaly detection, that is, determining which data is different from the rest. In unsupervised settings, users can influence models' organization, such as in interactive topic modeling (Hoque and Carenini, 2015; Hu et al., 2014; Lee et al., 2012) and interactive clustering (Awasthi et al., 2017; Balcan and Blum, 2008).

Finally, RL algorithms (or agents) mimic how humans learn; they interact with their environment and observe the results (positive or negative) of their interactions. These actions and rewards are stored as part of a decision policy for how the RL agents should perform in the environment. Artificial agents for gaming, such as chess and Go (Silver et al., 2017), utilize RL and typically outperform humans, because they have observed and know how to react to nearly all imaginable situations. In RL settings, a "human in the loop" can provide training through reward functions (Knox and Stone, 2012) or interventions (Saunders et al., 2018).

In this dissertation, we explore human interaction and experience with both supervised and unsupervised ML. In Chapter 4, we explore user experience with a supervised text classification system that supports instance- and feature-level feedback, similar to Settles (2011) and Kulesza et al. (2015). In Chapters 6 and 7, we explore user experience with an unsupervised interactive topic modeling system similar to Hu et al. (2014), which allows users to refine a topic model through topic- and model-level refinements, such as adding words or merging topics, respectively. We describe these interactive systems and their implementations in more detail in Chapter 3.

2.2 Primary IML System Characteristics

Prior researchers have put forth IML design guidelines (Amershi et al., 2019; Holmqvist, 2017), similar to those prescribed for user interfaces more generally (Shneiderman et al., 2009). These

guidelines enumerate primary system characteristics to consider when designing for user experience with IML, including transparency, predictability, control, and latency. For example, Holmqvist (2017) recommends designing to "ensure transparency" and "account for unpredictability." Here, transparency refers to whether the user can understand what the ML system is doing, and unpredictability means that ML systems may behave in unexpected ways. And, Amershi et al. (2019) outline design guidelines regarding control, particularly that users should be able to customize or direct ML models, and latency (i.e., that interactions should be efficient). In the following sections, we discuss these characteristics in more detail and how they relate to the research questions of this dissertation.

For concreteness, we discuss each characteristic as it applies in a specific IML case: an interactive linear regression model for predicting home property values. For this case, suppose a multiple linear regression model is provided a large set of labeled instances, where each instance is a property represented as a set of attributes (features) and labeled with its price. A property might have continuous features (e.g., "lot size"), categorical features (e.g., "number of rooms"), or binary features (e.g., "is on market"). Each property instance is represented as a vector of its features. From these instance vectors, the model learns to predict the prices of new properties given their features (e.g., "lot size," "number of rooms," and "whether for sale"). While we do not evaluate such a model in this dissertation, we choose this example as regression is a simple ML technique of which predicting home property values is a reasonable application.[1] Linear regression has also been studied by others in the context of ML transparency (Poursabzi-Sangdeh et al., 2018).

[1] As of the writing of this dissertation, Kaggle has been running a long term public competition for predicting house precises using regression techniques intended as an ML tutorial (see https://www.kaggle.com/c/house-prices-advanced-regression-techniques)

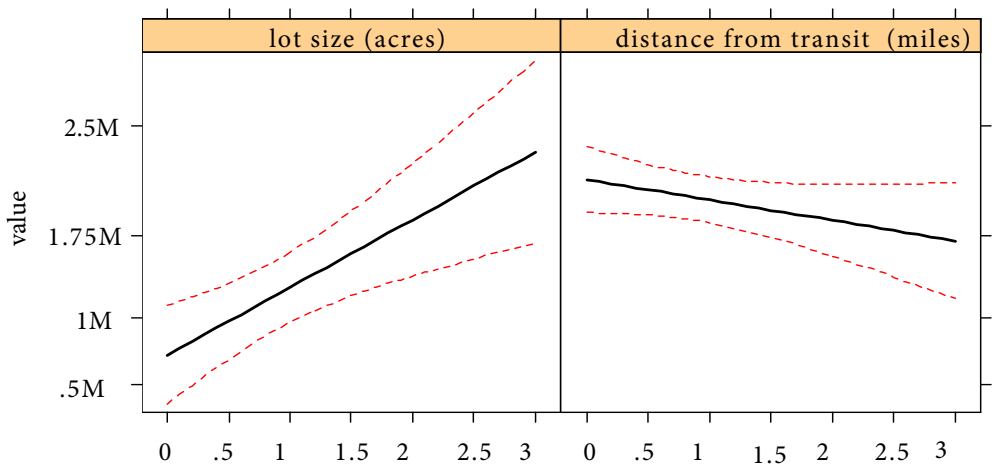


Figure 2.1: An example of a global model transparency technique for a regression for predicting home prices, where the effects of two individual features (lot size and distance from transit) are visualized. This figure was adapted from Fox (2003).

2.2.1 Transparency

Transparency (or explainability) in ML refers to what of a model's inner workings or decision making is exposed or explained. Transparency is also closely related to the concepts of intelligibility and interpretability, which refer to the ability of users to understand how systems works. ML intelligibility has received growing attention as ML models take on more important responsibilities in society and non-ML experts need to understand and trust these systems. More complex models are often more accurate. Thus, intelligibility research both develops global explanations, such as more *transparent* models (Alvarez-Melis and Jaakkola, 2018; Caruana et al., 2015; Lage et al., 2018; Si and Zhu, 2013) or black-box explanations (Lakkaraju et al., 2019), and local explanations of individual algorithm decisions (Bilgic and Mooney, 2005; Biran and McKeown, 2017).

Models can provide overall (or global) transparency by exposing their inner workings to give

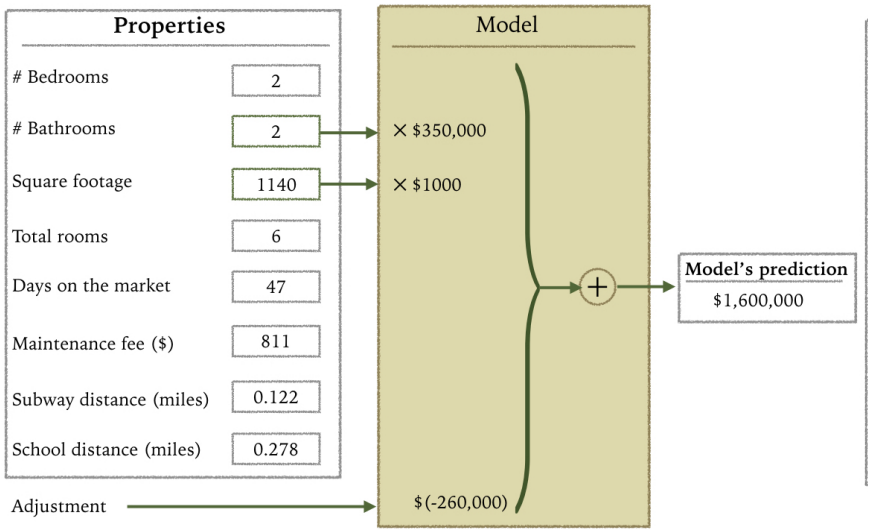


Figure 2.2: An example of a local explanation technique for a multiple linear regression for predicting home prices, where the effects of two individual features (number of bathrooms and square footage), as well as an adjustment value are shown. This figure is from Poursabzi-Sangdeh et al. (2018)).

insight into how they model underlying data for a deeper understanding of how they operate. For example, Simonyan (2013) increased the transparency of deep Convolutional Networks by producing artificial images representative of learned image classes. For our home pricing tool example, we might improve system transparency by exposing the underlying regression effect visualizations (Fox, 1987), which provide an estimation of how changing each feature affects predictions (assuming all other features are held constant). Figure 2.1 shows an example of such a visualization for two features: "lot size" and "number of rooms."

Alternatively, models can provide local explanations of individual algorithm decisions, such as rationales, interpretations, or justifications. These explanations can include input evidence (Feng and Boyd-Graber, 2019; Lei et al., 2016), localizations (Park et al., 2016; Selvaraju et al., 2017, or attention or saliency maps), natural language explanations (Camburu et al., 2018; Ehsan et al., 2019; Gkatzia et al., 2016), or local approximations (Ribeiro et al., 2016). A local explanation of

a predicted price in our home pricing tool, for example, might include information about which features most affected the property value. Poursabzi-Sangdeh et al. (2018) use such an explanation technique in their apartment pricing tool (Figure 2.2).

In this dissertation, we explore both global system transparency and local explanations: we focus on local explanations in Chapter 4, specifically, highlighting important words, and in Chapters 5 through 7, we expose a global model representation through visualization.

As transparency increases, end users form better mental models of how systems work, which in turn increases trust and satisfaction, and leads to continued usage (Herlocker et al., 2000; Kulesza et al., 2013; Lim et al., 2009; Pu and Chen, 2006; Sinha and Swearingen, 2002). Intelligibility is promising for supporting fairness and bias assessments (Dodge et al., 2019), improving perceived understanding (Kocielnik et al., 2019), convincing users to accept recommendations (Cramer et al., 2008), and motivating users to contribute to online communities (Rashid et al., 2006). However, explanations can decrease users' perceptions when algorithmic limitations or uncertainty are portrayed (Cai et al., 2019; Lim and Dey, 2011; Stowers et al., 2017) In particular, the depiction of a system's *uncertainty* of a decision or output can have a negative impact on trust, even when the system behaves as expected (Lim and Dey, 2011; Stowers et al., 2017). Transparency can have other negative effects, such as users' over-reliance on systems (Stumpf, 2016) and inability for users to detect systems' mistakes (Poursabzi-Sangdeh et al., 2018). ML-based systems can better set expectations (and appropriate trust) by exposing accuracy (Yin et al., 2019) or anticipated system mistakes (Kocielnik et al., 2019). In Chapter 4, we explore whether these insights in turn increase users desire to fix mistakes and improve systems.

Prior transparency research has also explored the effect of explanations on mental models, in particular on users' ability to *predict* how models would behave (Bunt et al., 2007; Chandrasekaran

et al., 2018; Poursabzi-Sangdeh et al., 2018), finding conflicting results For example, Poursabzi-Sangdeh et al. (2018) used an apartment pricing tool to explore whether users could better simulate the models' predicted apartment prices when shown model internals (i.e., a linear regression model with visible coefficients). Such explanations improved the ability of users to predict model behavior. Similarly, explanations improved predictability for a GUI customization tool (Bunt et al., 2007), but did not have an effect for a visual question answering system (Chandrasekaran et al., 2018). This discrepancy could be because users expected the ML model to change and therefore were less successful at predicting future model behavior. In Chapter 4, we explore this concept of *expected change* by asking users whether they think the system they evaluated will perform better, the same, or worse on new data.

Transparency is important in the context of IML, where users improve or guide models: users need to understand how models work to best fix them (Amershi et al., 2010; Fiebrink et al., 2009; Kulesza et al., 2012) and how models are explained changes user feedback (Kulesza et al., 2015; Rosenthal and Dey, 2010). Explained in the context of our home pricing tool, users who better understand how predictions are made (e.g., which of the home's features they are based on) can better provide input, such as re-weighting or adding features, to fix subsequent errors.

Rosenthal and Dey (2010) explored this concept in their email categorization tool, with which users were asked to classify a stranger's emails into folders. Presenting their system's prediction and low-level context (e.g., the email has keywords "A" and "B") helped users give effective feedback to better classify the emails. Kulesza et al. (2015) introduced their EluciDebug tool, based on the concept of "explanatory debugging," in which models provide explanations in a form that users can interact with to provide feedback. EluciDebug explains its classifier's binary predictions to users in the form of important input words and proportion of the data labeled as each class. Users in turn inform the classifier by correcting the prediction—*instance feedback*—or saying which

words are important for each class—*feature feedback*. Explanations and feedback—in particular feature feedback—allowed users to both better understand and correct the system's mistakes compared to system alternatives without explanations or feedback. While these studies tell us that explanations foster better feedback, prior work has not investigated how user perceptions—such as frustration and trust—are shaped by the presence or (sometimes more importantly) the absence of feedback mechanisms given explanations. Therefore, we address this in Chapter 4, using a similar data set, task, explanation, and feedback mechanisms as Kulesza et al. (2015).

2.2.2 Predictability

User interface design guidelines prescribe that interaction with systems should be *predictable* (or that systems should function as expected) to support user confidence and understanding (Hoekman, 2007); however, IML often violates this principle (Amershi et al., 2014) as these systems follow complicated algorithms or make decisions based on unseen knowledge. For example, imagine that a home pricing tool predicts the price of two similar homes to be vastly different. This behavior might appear unpredictable to a user who does not realize the homes differ by a highly weighted feature (e.g., "age of roof").

Kangasrääsiö et al. (2015) compared a predictable and random algorithm for an interactive search interface. They define predictability as whether algorithms follow a strategies that users can easily understand. Allowing users to see predicted effects of their actions resulted in small improvements in user acceptance, perceived usefulness, and task performance, likely because such a technique mitigated surprise when unexpected updates occurred.

Gajos et al. (2008) examined the impacts of predictability and system accuracy on user experience. Increasing the predictability and accuracy of an adaptive user interface led to strongly improved

satisfaction. However, the accuracy improvement had a stronger effect on performance, utilization, and some satisfaction ratings than predictability.

In this dissertation, we explore predictability as it relates to transparency and control.

2.2.3 Control

In ML, *control* refers to the ability of the user to control or affect change in the underlying model. Typically control is in the form of user input, such as to adapt, personalize, or improve ML models. For example, imagine two instances of a home pricing tool: one that gives users control and one that does not. The "control" version might allow users to re-weight features or correct or confirm predicted prices, whereas the "non-control" version would restrict users to only viewing the predictions. As IML describes end user interaction to train or adapt models, some level of control is implied (we detailed specific mechanisms for interacting with, or controlling, ML in Chapter 2.1).

End users want and need mechanisms for control, for user interfaces in general (Shneiderman et al., 2009), and for ML-based systems (Amershi et al., 2019). Specifically, providing end user control can manage user expectations (Kocielnik et al., 2019) and increase confidence (Du et al., 2017) or satisfaction (Roy et al., 2019; Vaccaro et al., 2018). Du et al. (2017) compared user experience with three variants of their PeerFinder system based on the control given to the user. Users were more confident in the results and engaged with the system when given more control even with the negative effect of complexity.

Prior research on *control* primarily considers (1) whether users can provide input to models or (2) the particular control mechanisms that are supported (e.g., instance vs. feature feedback). However, it is not always the case that systems can support users' desired control, particularly

because IML systems must balance unseen knowledge, such as previously learned models or data, with users' specifications—meaning they must support shared control between algorithm and user (Holmqvist, 2017). This discrepancy suggests additional dimensions of control: *adherence*—how well models apply user specifications during updates—and *instability*—whether models make any other changes. For example, suppose users specify that lot size should not impact value in a home pricing tool (i.e., by setting the weight of the "lot size" feature to zero). The updated model may not *adhere* to this input exactly, as doing so would greatly reduce prediction accuracy. Additionally, re-weighting the "lot size" feature may have cascading effects (e.g., promoting other features), thus making the model appear unstable.

Transparency is particularly important when users are given control (Kulesza et al., 2010; Rosenthal and Dey, 2010), as making users aware of how models work in turn makes them better at providing feedback. However, increased transparency also means that users can better discern what models do with their feedback, or whether models incorporate it *predictably*. For opaque systems, providing "difficult-to-validate" controls, whether or not they work, can increase satisfaction (Vaccaro et al., 2018). But how do users react to unexpected behavior when controls are easier to validate? For example, after users specify that "lot size" should have no weight on subsequent predictions, are they surprised if the model later *explains* a predicted price using the home's lot size?

Control and predictability are important considerations for intelligent systems (Höök, 2000), however, the interaction between the two has not been fully explored, particularly in transparent models where they are more easily perceived. In Chapters 6 and 7, we explore two specific aspects of control (adherence and instability) as they relate to predictability.

2.2.4 Latency

Latency refers to the time a user must wait for the system to perform a task. Prior work in IML has called for rapid interaction cycles (Amershi et al., 2014) to minimize attention loss (Horvitz, 1999) and reduce short-term memory load (Shneiderman et al., 2009). However, many IML systems do not provide real-time updates, where this *latency* is typically related to the size of the data and complexity of the computation. For example, our early interactive topic modeling implementation took between 5 and 50 seconds to update the model based on refinement operations (Hu et al., 2014). In IML, concerns about latency consider both how long the system takes to update as well as how often it does so. In Chapters 6 and 7, we explore whether latency affects end user experience and suggest methods for alleviating negative effects.

2.3 Summary

Our survey of related work in IML highlights both the user benefits of IML in general and of designing for particular system characteristics (e.g., transparency and control). We also identify gaps in prior work, in particular, that the interactions between control and transparency have not been fully explored. To this end, we explore how end users are affected when transparent systems do not adhere to user input (Chapter 6 and Chapter 7) or even support it and when feedback is requested without an explanation (Chapter 4).

This chapter focused on IML and its characteristics from the perspective of the human in the loop; in Chapter 3 we review the specific IML techniques explored in this dissertation.

Chapter 3: Background on Specific Interactive Systems

This chapter reviews the two particular interactive (or human-in-the-loop) system categories researched in this dissertation: interactive text classification (supervised) and interactive topic modeling (unsupervised). In particular, we review the text classification technique used for our study on the interaction of control and explanations in supervised ML (Chapter 4). We then provide a detailed background of interactive topic modeling, including underlying statistical topic modeling algorithms, measurements for topic model quality, topic model transparency, and finally interactive topic modeling methods.

3.1 A Supervised IML Case: Interactive Text Classification

Text classification is a common natural language processing task where documents are classified based on their content. Common examples of text classification include sentiment analysis and email classification. Text classification is supervised, meaning that algorithms are trained on labeled documents (e.g., emails previously labeled with categories), from which they learn to predict the label of new documents from the documents' *features*, such as words, phrases, or other metadata.

User input can be incorporated into text classification algorithms in the form of additional training labels, modified features, or both (Cohn et al., 1994; Kulesza et al., 2015; Raghavan et al., 2006;

Settles, 2011). Cohn et al. (1994) presented an early argument for "active learning," based on the idea that not all additional training labels provide the same value to an algorithm. Therefore, in active learning, algorithms direct users to label the most beneficial input instances. Raghavan et al. (2006) extended the traditional active learning framework to additionally incorporate feature feedback and found significant improvement over traditional active learning classifier performance by feature re-weighting. Settles (2011) built on this idea and implemented DUALIST, an interactive text classification system, which solicits and learns from instance and feature feedback. With DUALIST, end users produced high-quality classifiers with minimal effort, but Settles (2011) did not study user experience or compare system designs. To that end, Kulesza et al. (2015) implemented EluciDebug (introduced in Chapter 2.2.1) to explore whether certain explanation types yielded better feedback. EluciDebug uses a common text classification approach, the Multinomial Naïve Bayes model (MNB), for performing binary text classification of hockey and baseball emails. We similarly use MNB for classifying hockey and baseball emails in our study in Chapter 4.

The MNB classifier assumes that the text data was generated by a parametric *mixture* model, assuming each document is about only one topic (or class). To contrast, more complex *admixture* models, such as Latent Dirichlet Allocation (LDA), assume documents are about multiple topics. We discuss LDA in more detail later in this chapter.

MNB estimates the mixture model parameters from labeled training documents (e.g., emails labeled as "hockey" or "baseball"). Given these estimates, MNB classifies new documents by calculating the posterior probability that each class would have generated the document, $P(c|d)$, and choosing the most probable class. MNB calculates these probabilities using Bayes Theorem:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{3.1}$$

In MNB, documents are sets of words drawn from the same vocabulary V. This approach treats documents as "bags of words," assuming all features are independent, specifically that the probability of each word in a document is independent of the word's context and position. Therefore, each document is drawn from a multinomial distribution of words; $P(d|c)$ is then the product of the conditional probability of each of document's words, w, given class c. Additionally, we can drop $P(d)$ from Equation 3.1, as it is constant given the input; this gives us:

$$P(c|d) \propto P(c) \prod_{1<k<n_d} P(w_k|c) \tag{3.2}$$

Here, our goal is to find the best class for the document, which is the most likely class or the *maximum a posteriori* (*MAP*) class, c_{map} (Manning et al., 2008):

$$c_{map} = \underset{c \in C}{\operatorname{argmax}} \hat{P}(c|d) = \underset{c \in C}{\operatorname{argmax}} \hat{P}(c) \prod_{1<k<n_d} \hat{P}(w_k|c) \tag{3.3}$$

The parameters of the generative component for each class are the probabilities for each word, $\hat{P}(w_k|c)$, or how much evidence the word w_k contributes that c is the correct class. MNB estimates $\hat{P}(w_k|c)$ as the relative frequency of the word w_k in training documents belonging to class c. That is, we count the number of times the word w_k appears in a document of class c in the training set divided by the total count of all words for class c.[1] The class prior parameters, $\hat{P}(c)$, or the prior probability of class c, are estimated as the relative frequency of the class c in the training set. Finally, the output class c with the highest estimated $\hat{P}(c|d)$ is assigned to the input document d.

In Chapter 4, we study user interaction with an MNB model for classifying hockey and baseball emails. Here, we *explain* the resulting classification to end users by highlighting the terms that

[1] We additionally use a smoothing prior $alpha > 0$ to prevent zero probabilities resulting from terms not being present in the training set.

contributed the most evidence to either classification, meaning we choose from all of the document's words w, those with the highest $P(w|c)$ for either c.

Additionally, in Chapter 4, we support two types of user feedback: instance-level (i.e., confirming or correcting predicted categories) and feature-level (i.e., specifying important words for the correct classification). For instance-level feedback, we treat each corrected document as another training example of class c and re-train the classifier. This effectively increases $P(c)$ and $P(w|c)$ for all words w in the document. Feature-level feedback is incorporated by first training the classifier on the training data (i.e., estimating $P(w|c)$ from relative counts) and then explicitly adjusting $P(w|c)$ for each provided word and class by a fixed k (either increasing it by k for the specified class or decreasing it by k for the alternate class). We then use the adjusted probabilities to classify the test set.[2]

3.2 An Unsupervised IML Case: Interactive Topic Modeling

In the following, we describe topic modeling, focusing on a common technique called Latent Dirichlet Allocation (Blei et al., 2003, LDA). We then describe automatic methods for evaluating topic models, topic model transparency from both automatic labels and topic visualizations, and finally end user control of topic models in the form of interactive topic modeling.

3.2.1 Topic Modeling and Latent Dirichlet Allocation

Topic modeling algorithms are statistical, unsupervised models that discover key themes in large corpora of documents without labeled training data (Blei, 2012). These approaches discover thematic structure from large corpora and organize documents by the themes they are about. In this

[2]In the MNB-based interactive text classification implementation we use in Chapter 4, k is 20% or 0.20.

way, topic modeling provides users with a high-level overview of the topics discussed in their documents, where the individual topics can link back to the original documents for directed exploration. While topic modeling is commonly applied as a basis for higher-level tasks, such as sentiment analysis (Lin and He, 2009), word sense disambiguation (Boyd-Graber et al., 2007), or behavior mining (Hospedales et al., 2009), we focus on corpus understanding tasks where topic model outputs are presented and interacted with directly by end users.

LDA is a common topic modeling approach, which, like MNB, follows a generative process where documents are treated as "bags of words," meaning order is ignored. Where MNB is a mixture model, meaning documents are assumed to be about only one topic, LDA is an *admixture* model, meaning documents are assumed to be a mixture of multiple topics.

LDA assumes that each document d is generated from a fixed set of k topics, where each topic is a multinomial distribution, ϕ_z, over a vocabulary of size V. Each document d is an admixture of topics θ_d. Each instance of a word, or token, w_i, indexed by i in document d, is generated by first sampling a topic assignment $z_{d,i}$ from the document's topic distribution θ_d and then sampling a word token from the corresponding topic's distribution ϕ_{z_i}. The multinomial distributions θ and ϕ are drawn from Dirichlet distributions that encode sparsity—how many words you expect to see in a topic or how many topics in a document—and α and β are the Dirichlet priors over θ and ϕ, respectively. Discovering the latent topic assignments z from the observed words w requires inferring the posterior distribution of the latent variables that best explain the observed data, $p(z, \phi, \theta | w, \alpha, \beta)$. For LDA, the latent variables of interest are the topics (z), the documents' distributions over topics (θ), and the topic-word assignments (ϕ). This computation is intractable, so it is typically approximated (Blei et al., 2003). We describe two common approximation methods: collapsed Gibbs sampling (Griffiths and Steyvers, 2004), which takes repeated samples from the conditional distribution for each latent variable in turn and then updates the parameters, and

variational Expectation-Maximization (Blei et al., 2003, EM), which uses an EM-like algorithm to optimize the parameters.

Collapsed Gibbs sampling does not explicitly represent the per-document distribution over topics (θ) or the topic distribution over words (ϕ) as parameters to be estimated, as these are integrated out, and instead considers the posterior distribution over the assignments of words to topics, $P(z|w)$ (Griffiths and Steyvers, 2004). This approach iteratively samples a topic assignment, $z = t$ given an observed token w in document d and all other topic assignments, $z\_$, with probability

$$P(z = t|z\_, w) \propto (n_{d,t} + \alpha) \frac{n_{w,t} + \beta}{n_t + V\beta} \tag{3.4}$$

Here, $n_{d,t}$ is the number of times topic t is in document d, $n_{w,t}$ is the count of token w in topic t, and n_t is the marginal count of tokens assigned to topic t.

For traditional topic models, the Gibbs sampler assigns latent topics Z for all tokens in the corpus, going over all the documents until the algorithm converges. The state of the sampler represents the algorithm's best guess of the topic assignments for every token.

Alternatively, variational EM turns posterior inference into an optimization task and approximates the posterior with a simpler distribution, $q(z)$; here the idea is to pick a single $q(z)$ that best approximates the posterior using a tractable family of distributions over the parameters and latent variables by first defining a mean field variational distribution:

$$q(z, \phi, \theta|\lambda, \gamma, \pi) = \prod_{k=1}^{K} q(\phi_k|\lambda_k) \prod_{d=1}^{D} q(\theta_d|\gamma_d) \prod_{n=1}^{N_d} q(z_{dn}|\pi_{dn}) \tag{3.5}$$

Here, γ_d and π_d are local variational parameters of the distribution q for document d, and λ_k is

a global variational parameter for topic k. Inference minimizes the KL divergence between the variational distribution and true posterior. In Variational EM inference, we estimate distributions of the topics ϕ_k and the documents θ_d, namely the parameter λ_k of the Dirichlet prior over the topics' words ϕ_k, and the parameter γ_d of the Dirichlet prior over the documents' topics θ_d.

Essentially, in the *E-step*, the model assigns latent topics based on the current value of λ, and in the *M-step*, the model updates λ using the current topic assignments. Specifically, the Variational EM algorithm is defined as follows (Geigle, 2016):

E-step: Minimize KL divergence from p to q for each document d by performing the following updates until convergence:

$$\pi_{d,n,i} \propto \lambda_{i,w_{d,n}} exp\left(\psi(\gamma_{d,i}) - \psi\left(\sum_{k=1}^{K} \gamma_{d,k}\right)\right) \tag{3.6}$$

$$\gamma_{d,i} = \alpha_i + \sum_{n=1}^{N_d} \pi_{d,n,i} \tag{3.7}$$

Where $\psi(\bullet)$ is the "digamma" function; q is now a good approximation to the posterior distribution p.

M-step: Using q, re-estimate λ. Specifically, since $\pi_{d,n,i}$ represents the probability that word $w_{d,n}$ was assigned to topic i, we compute and re-normalize expected counts:

$$\lambda_{i,v} = \beta_v + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \pi_{d,n,v} I(w_{d,n} = v) \tag{3.8}$$

Where $I(\bullet)$ is the indicator function that takes value 1 if the condition is true and value 0 otherwise.

One of the major differences between these approximation techniques is in how they perform

latent topic assignment. Specifically, in Gibbs sampling topics are assigned to each token directly (a hard topic assignment), and therefore the topic assignments $z\_$ can be used to compute ϕ and θ directly. However, Variational EM inference performs a soft assignment, therefore we estimate distributions of the topics ϕ_k and the documents θ_d.

In Chapter 3.2.4, Chapter 6, and Chapter 7, we describe techniques for injecting human knowledge and expertise into these topic modeling approaches to guide the LDA algorithm to better topics. We also use LDA to uncover the topics used to compare topic representations in Chapter 5, building off an existing LDA implementation with Gibbs sampling in Mallet (Yao et al., 2009).

3.2.2 Topic Model Evaluation

Topic models typically include topics of varying quality, or interpretability (Lau et al., 2014). For example, imagine two topics learned from a collection of emails: (1) {*puck, period, goal, ice, capitals*} and (2) {*respond, thanks, hello, best, nice*}. The first topic is clearly related to "hockey," but the second topic does not have a clear theme and is less interpretable. As qualitative assessment of interpretability by end users is effortful to collect, topic models are often evaluated using statistical methods. Prior work has evaluated topic model quality by *perplexity*, which measures how well a model can predict words in unseen documents (Wallach et al., 2009b). However, Chang et al. (2009) argued that evaluations optimizing for perplexity encourage complexity at the cost of human interpretability—a quality we are especially concerned with in this dissertation. We, therefore consider *topic coherence* methods as they better aligned with end-user topic assessments (Mimno et al., 2011).

Methods that compute topic coherence deem topics to be more coherent if they contain words that are more commonly found together than apart in a reference corpus (e.g., Wikipedia or Google);

the topic words in the first topic above are more *coherent* than those in the second topic. These methods estimate a topic's observed coherence by computing word co-occurrence[3] probabilities for the top-N topic words given the reference corpus.

Newman et al. (2010a) compared the effectiveness of different automatic topic coherence methods and reference corpora. They trained two distinct topic models, one from a corpus of books and the other a corpus of news articles, and evaluated how well varied automated coherence measurements correlated with human coherence ratings for the generated topics. Among the individual topic coherence measures they evaluated, a method based on pointwise mutual information (PMI) for computing word co-occurrence using a Wikipedia reference corpus (opposed to WordNet or Google corpora) best correlated to human topic coherence ratings. The authors posited that the encyclopedic nature of Wikipedia means it is robust to varied domains (e.g., books and news articles). Lau et al. (2014) extended this work by applying normalized pointwise mutual information (Bouma, 2009, NPMI) to reduce the bias of PMI for words of lower frequency. Following Lau et al. (2014), in this dissertation we calculate topic coherence, C, as the pairwise NPMI between topic words in the reference corpus:

$$C = \sum_{j=2}^{N} \sum_{i=1}^{j-1} \frac{log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}}{-logP(w_i, w_j)} \tag{3.9}$$

Where $P(w_j, w_i)$, is the probability of observing both w_i and w_j co-occurring in the reference corpus and $P(w_j)$ and $P(w_i)$ are the probabilities of observing w_j and w_i, respectively.

In Chapter 5, we use NPMI to assess the quality of the topics, comparing how well various topic *explanations* represent the "best" and "worst" quartile of topics. And, in Chapters 6 and 7, in addition to qualitative assessments, we again use NPMI-based topic coherence to measure task per-

[3]Word co-occurrence is computed using a sliding window of size K within each reference document.

formance by comparing model quality before and after users refine topics. These studies expose inconsistencies between computed differences in topic coherence and users' perceptions of improvement, which we discuss as a potential area for future work (Chapter 8.2).

3.2.3 Topic Model Transparency

Topic modeling inference outputs model parameters, which represent each document in a corpus as a distribution of topics and each topic as a distribution of words in the vocabulary. This output is inherently transparent since it exposes a representation of how the algorithm models the data; however, techniques are needed to support users in quickly and easily understanding topics. In the following sections, we review two particular mechanisms for topic model transparency explored in this dissertation: topic labels and topic visualizations.

Automatic Topic Labels

Topics can described by short labels (e.g., "sports" or "criminal activity"). Prior work has focused on automatic generation of such labels for *explaining* topics. Lau et al. (2011) used Wikipedia articles to automatically label topics, based on the assumption that for each topic there will be a Wikipedia article title that offers a good representation of the topic.

Aletras et al. (2014) used a graph-based approach to better rank candidate labels. They generated a graph from the words in candidate articles and used PageRank to find a representative label. In Chapter 5, we use an adapted version of the method presented by Lau et al. (2011) as a representative automatic labeling algorithm.

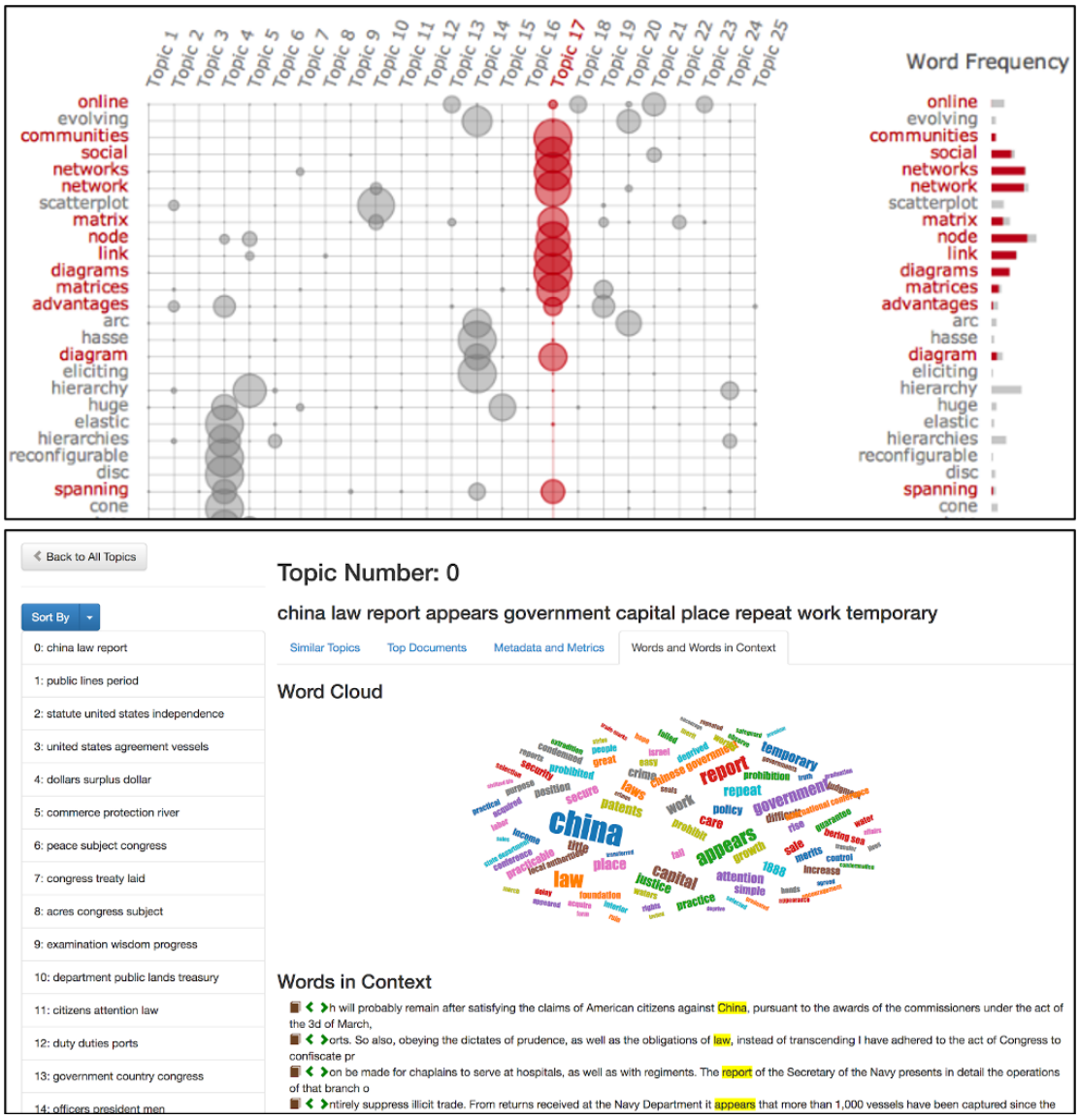


Figure 3.1: Two distinct topic model-based tools exemplifying variability in how topics are represented. Termite (top) uses word lists with bars (Chuang et al., 2012) where Topical Guide (bottom) uses simple word lists and word clouds (Gardner et al., 2010).

Topic Visualizations

Numerous distinct topic model representations have been introduced, which vary in how they present topic models and individual topics (Figure 3.1); however, there has been no systematic evaluation of them related to their *interpretability* or how they support end users in understanding

topics. Most of these existing visualizations represent individual topics with either simple word lists, word lists with bars (or other frequency representations), or word clouds.

Topical Guide (Gardner et al., 2010), Topic Viz (Eisenstein et al., 2012), and the Topic Model Visualization Engine (Chaney and Blei, 2012) were designed to support corpus understanding and directed browsing through topic models. They each display the model overview as an aggregate of underlying topic visualizations. For example, Topical Guide (Figure 3.1, bottom) uses horizontal word lists when displaying an overview of an entire topic model but used a word cloud of the top 100 topic words when displaying only a single topic. Alternatively, Topic Viz and the Topic Model Visualization Engine both represent topics with vertical word lists; the latter also uses set notation.

Other tools provide additional information within topic model overviews, such as the relationship between topics or temporal changes in the model. Sievert and Shirley (2014) included information about the relationship between topics in the model in their LDAVis tool. LDAVis uses multi-dimensional scaling to project the model's topics as circles onto a two-dimensional plane based on their inter-topic distances; the circles were sized by their overall prevalence. The individual topics are visualized on demand using a word list with bars. In our prior work, we developed Hierarchie (Smith et al., 2014b), which organizes topics hierarchically in a tree representation, such that users can "drill into" topics of higher granularity. Hiearchie displays individual topics on hover as vertical word lists.

Another of our prior research efforts developed TopicFlow (Malik et al., 2013),[4] which visualizes how a model has changed over time using a Sankey diagram (Riehmann et al., 2005). TopicFlow represents individual topics both as word lists in the model overview and as word lists with bars when viewing a single topic or comparing between two topics. Argviz (Nguyen et al., 2013) also

[4]This prior work, of which the author was a primary contributor, is not included in this dissertation.

visualizes topics over time, and specifically captures temporal shifts in topics during a debate or a conversation; Argviz presents individual topics as word lists in the model overview and using word list with bars for the selected topics. Finally, Klein et al. (2015) used a *dust-and-magnet* visualization (Soo Yi et al., 2005) to visualize the force of topics on newspaper issues. Their tool displays the temporal trajectories of several newspapers as dust trails in the visualization and displays individual topics as word clouds.

In contrast to these visualizations that supported viewing the underlying topics on demand, Termite (Chuang et al., 2012) provides a model overview using a tabular layout of words and topics, which also supports quick comparison across topics. Termite (Figure 3.1, top) organizes topic models into clusters of related topics based on word overlap, which the authors argued was both space-efficient and speeds corpus understanding.

While a diverse set of individual topic representations are commonly used in topic model visualizations, there has been no systematic evaluation of them for their *interpretability*, or how well they support users in understanding topics. To this end, in Chapter 5, we develop a novel topic visualization and compare it to three others: word list, word list with bars, and word cloud for topic interpretability.

3.2.4 Interactive Topic Modeling

Interactive topic modeling provides mechanisms for end users—specifically those who are not ML experts—to control (or refine) topic models as they are being generated to produce higher quality, domain and user-specific topic models. Numerous tools have been designed around this concept, each allowing users to perform a variety of model refinements (e.g., adding words to topics, splitting topics, or merging topics) with limited user studies to demonstrate whether they

are usable or useful on any real world-tasks.

Interactive topic modeling requires mechanisms for encoding user feedback to influence the model. Typically these mechanisms include both *forgetting* bad things the model has learned and *injecting* new knowledge into the model.

Andrzejewski et al. (2009) introduced formalized topic model *constraints* for specifying that words should or should not belong to the same topic in the form of "must-link" and "cannot-link" constraints between words using tree-based priors (Boyd-Graber et al., 2007).

Our prior work[5] developed an interactive topic modeling tool following this approach and based on the idea that topic coherence can be improved by correlating (or linking) similar words into topics and splitting (or unlinking) topic words that should not have been together (Hu et al., 2014). This early interactive topic modeling tool allowed users to add and remove words from topics and represented these operations to the model as a set of constraints: "must-link," or positive constraints encourage words to appear in the same topic (i.e., the words have similar probabilities for the same topic), and "cannot-link," or negative constraints push words into different topics (i.e., the words have differing probabilities for the same topic).

These constraints are *injected* into the model through tree-based priors, as in Andrzejewski et al. (2009). As exemplified by Figure 3.2 from Hu et al. (2014), internal nodes for words with positive correlations (like "drive" and "ride") have high transition priors β, meaning that a topic will have similar probability for having both (if the edge from root to internal node has high probability like in Topic 1) or neither (if the edge from root to internal node has low probability like in Topic 2) of the words. Alternatively, internal nodes for words with negative correlations have low transition priors (like "space" and "tea"), meaning a topic can have only either of the words, but not both. Hu

[5]This prior work, of which the author was a contributor, is not included in this dissertation.

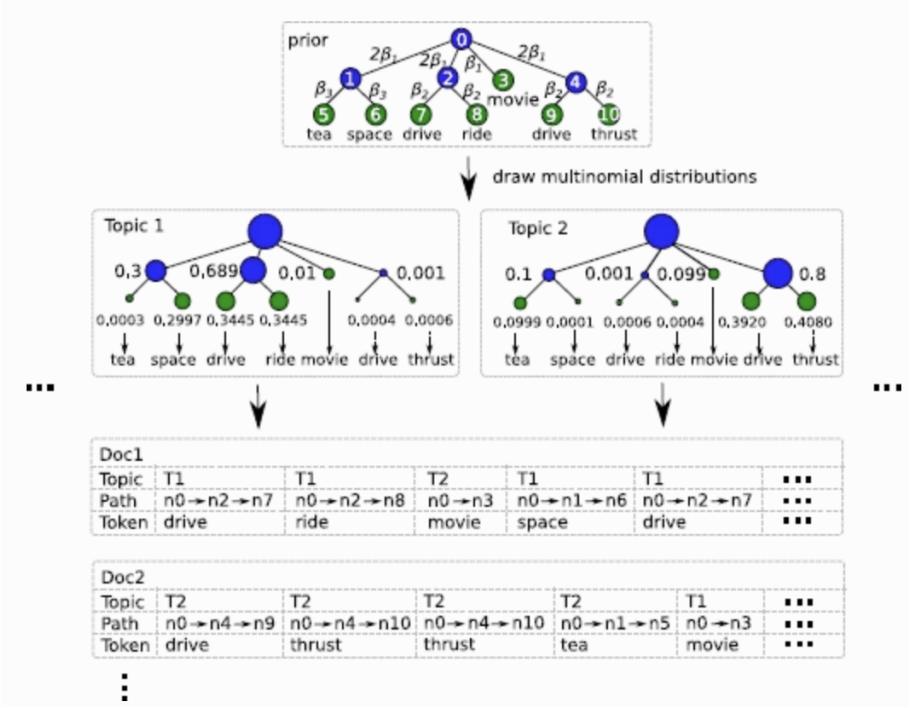


Figure 3.2: Figure taken from Hu et al. (2014), which provides an example of the generative process for drawing topics (first row to second row) and then drawing token observations from topics (second row to third row) for tree-structured prior topic models. In the second row, the size of the children nodes represents the probability in a multinomial distribution drawn from the parent node with the prior in the first row. Different hyperparameter settings shape the topics in different ways. For example, the node with the children "drive" and "ride" has a high transition prior β_2, which means that a topic will always have nearly equal probability for both (if the edge from the root to their internal node has high probability) or neither (if the edge from the root to their internal node has low probability). However, the node for "tea" and "space" has a small transition prior β_3, which means that a topic can only have either "tea" or "space," but not both.

et al. (2014) inject "must-link" and "cannot-link" word-level constraints into the model as positive and negative correlations, respectively.

Forgetting prior "bad" information is handled by strategic unassignment of states (or ablation); in the Gibbs sampler, a model's state is defined by the current topic assignments. To forget information about certain tokens, we set a token's topic assignment to an invalid topic (i.e., -1) and decrement any counts associated with that token ($n_{w,t}$ and $n_{d,t}$ in Equation 3.4) (Hu et al., 2014). After ablation, we continue inference. The Gibbs sampler treats all tokens with -1 topic assign-

ments as if it is seeing them for the first time. During inference, initial assignments are sampled for the "unassigned" tokens and the assignments are updated for the previously assigned tokens.

This project evaluated participants exploring a dataset with the interactive topic modeling tool and then answering questions about the dataset. Participants better understood the underlying data by working with the tool than by reading the documents alone. Although the interactive topic modeling tool showed promise for enhancing information seeking behavior, this study did not evaluate users' experience, performance, or explore the implemented refinements.

Chuang et al. (2013) applied "must link" and "cannot link" constraints within the Termite visualization tool (Chuang et al., 2012), which uses a matrix visualization to support topic comparison. Users updated the model by clicking on words in the matrix visualization to promote or demote them in the topics. Analysts working with their own datasets were shown early prototypes of interface, and their feedback suggested that these word-level constraints were useful, but additional topic-level refinements were preferred. Similarly, Bakharia et al. (2016) and Saeidi et al. (2015) implemented "must-link" and "cannot-link" constraint-based refinements in their interactive topic modeling systems applied qualitative content analysis and source code analysis, respectively.

Other interactive topic modeling systems have used different approaches for incorporating user feedback into models. Yang et al. (2015) introduced an efficient, factor graph framework for incorporating prior knowledge into LDA, Sparse Constrained LDA (SC-LDA). SC-LDA injects new information into the model using potential functions, $f_m(z, m, d)$ of the hidden topic z of word type w in document d. SC-LDA can be used to incorporate both word correlation and document label knowledge into topic models efficiently. However, SC-LDA has not been evaluated as part of an interactive topic modeling system with end users. To this end, we evaluate SC-LDA compared to other interactive topic modeling approaches in Chapter 7.

UTOPIAN (Choo et al., 2013) used a matrix factorization-based approach for a system that allowed users to change word weights, split and merge topics, as well as to create new topics. The authors presented a case study to demonstrate the tool on different data sets, but did not study usability of such operations. Alternatively, ConVisIT (Hoque and Carenini, 2015) took a graph-based approach and supported only splitting and merging topics. The authors performed a task-based user study to determine if ConVisIT was a preferrable interface for exploring conversations to two alternatives: ConVis, which is a similar conversation exploration tool but that lacks support for refining topics, and SlashDot,[6] a popular technical conversation blog. Participants identified more insightful comments with ConVisIT than with the counterpart systems. Participants also used the split topic operation more frequently than the merge topic operation.

None of these existing systems were created following a user-centered design process, meaning refinements were chosen and implemented without first understanding the needs of the end users, resulting in systems that may not meet users' needs and expectations. And, although in total these systems implemented a variety of refinement operations, none implemented nor evaluated a wide range of user-focused refinement operations. To address these issues, in our prior work,[7] we conducted a two-part study where, first, non-expert users explored a static topic model and provided input on what refinement operations they would want an interactive topic modeling system to provide, and, second, the most frequently requested operations were then provided to a new set of users who employed them in a wizard-of-Oz setting (i.e. without incorporating the refinements into the backing topic model) to refine individual topics (Lee et al., 2017). We identified a refinement set for non-expert users and highlighted patterns in how non-expert users interpret topics and apply refinement operations. However, an important limitation of that study was that it did not implement the refinements, thus the user actions did not affect the underlying model. As a result, the

[6] https://slashdot.org/

[7] This prior work, of which the author was a contributor, is not included in this dissertation.

findings may not reflect realistic usage in a truly interactive system, and participants did not face IML challenges such as lack of control and complex model updates. To this end, we implement these user-preferred refinements and evaluate them in a truly *interactive* topic modeling system in Chapter 6. This formative study exposes initial user reactions to system characteristics, such as instability, adherence, and latency (introduced in Chapter 2.2). We explore reactions to these characteristics in more detail, with a comparative study in Chapter 7.

Recall the two inference techniques discussed in Chapter 3.2.1: Gibbs samping and variational EM inference. These inference techniques yield different system qualities: Gibbs sampling-based methods can yield more coherent topics than variational inference (Nguyen et al., 2015); although others claim the inference technique does not matter, so long as hyperparameters are tuned (Asuncion et al., 2009). Additionally, Gibbs sampling and variational inference exhibit different convergence properties (Asuncion et al., 2009). While Gibbs sampling is often preferred for small datasets and interactive settings because of its low latency, variational inference can scale via parallelization to millions of documents (Hoffman et al., 2010; Zhai et al., 2012). Partly for these reasons, in Chapter 7, we vary the inference techniques used for our three distinct interactive topic modeling variants (two use Gibbs sampling and one uses variational inference) to explore how the ensuing system characteristics (e.g., speed, adherence, quality) affect end users.

3.3 Summary

In this chapter, we presented details on the two particular interactive systems explored in this dissertation: interactive text classification and interactive topic modeling. We use the interactive text classification system as the basis for our study on the interactions of explanations and feedback in Chapter 4.

We then switch the research focus to interactive topic modeling for the remaining chapters. We focus on gaps in prior work. First, our review of related work identified many different topic model visualizations, yet to our knowledge, no prior studies have compared these visualizations for interpretability. Therefore, in Chapter 5, we introduce a novel topic visualization and compare it to other common visualizations (e.g., word list and word cloud) to determine with which users can quickly and easily understand topics. Second, we implement a new interactive topic modeling system, based on users' desired refinement mechanisms and interpretable topic explanations, and evaluate this system with a formative study in Chapter 6. This study identifies two particular aspects of control, which are exposed when users interact with transparent systems: adherence and instability. Finally, in Chapter 7, we compare three interactive topic modeling approaches that vary in terms of adherence and instability to see whether users perceive and are affected by these characteristics.

Chapter 4: Interactions between Transparency and Control in Supervised ML: an Empirical Study[1]

In this chapter, we explore control and transparency in supervised ML. Here transparency is provided by automatically generated post-hoc explanations for a model's predictions and control refers to end users' ability to provide feedback to correct or improve the model.

Automatically generated explanations of how ML models reason can help users understand and accept them. However, explanations can have unintended consequences: promoting over-reliance or undermining trust. This chapter investigates how explanations shape users' perceptions of ML models with or without the ability to provide feedback to them: (1) does revealing model flaws increase users' desire to "fix" them; (2) does providing explanations cause users to believe—wrongly—that models are introspective, and will thus improve over time.

To study how explanations and varied supports for user feedback impact experience with a ML model, we conducted two crowdsourced experiments with 180 participants each. Both experiments used a common text-classification task of sorting emails as "hockey" or "baseball" (Kulesza et al., 2015; Settles, 2011).

Because we expected explanations and feedback would be particularly salient when the model

[1]The work in this chapter was performed in collaboration with Ron Fan, Melissa Birchfield, Tongshuang Wu, Jordan Boyd-Graber, Dan Weld, and Leah Findlater and was accepted to the ACM CHI Conference on Human Factors in Computing Systems (CHI) 2020.

could be improved, the first experiment used a *lower quality* model (~ 75% accuracy), trained on a handful of training documents. Participants reviewed predictions made by the classification model with or without *explanations*, and with one of three levels of user *feedback* to the model: none, instance-level (correcting or confirming the model's prediction), and feature-level (telling the model *how* to predict). We measured participants' subjective post-task satisfaction, including frustration and trust, as well as how they expected the model to change. The second study experiment was exactly the same as the first, but with a *higher quality* model (~ 95% accuracy) to understand the effects of model quality on our findings.

Our findings contribute the following observations to the nascent understanding of interactive and explainable machine learning: (1) users wanted the *opportunity* to provide feedback, regardless of model quality or whether they received explanations; (2) for the low-quality model, feedback reduced frustration and increased trust and acceptance, but explanations had the opposite effect; therefore, explanations without the opportunity for feedback resulted in an especially negative user experience; (3) for the high-quality model, users were not as frustrated, yet requesting feature-level feedback without an explanation reduced trust; (4) regardless of model quality, when users provided detailed feedback, they expected more improvement; yet, users generally expected model improvement even for conditions without any user feedback, demonstrating possible misconceptions of ML models by end users.

Despite the constrained setting (i.e., a classical, binary text classification task, with a simple explanation), we see this work as an important step in illustrating a key relationship between explanations and feedback. We conclude this chapter by discussing extensions to more complex tasks and models with more sophisticated explanation and feedback mechanisms.

4.1 Study 1: Understanding Explanations and Feedback with a Low Quality Model

With a crowdsourced, between-subjects experiment, we explored how explanations and support for feedback affect satisfaction and expectation of change with a low quality model.

4.1.1 Method

Simple models and tasks are a useful starting point to examine the intersection of explanations and feedback. Therefore, in this study, participants reviewed a simple text classification model's predictions with or without explanations and with one of three options for providing user feedback to the model: no feedback, correcting or confirming the model's predictions (instance-level feedback), or suggesting important words to the model (feature-level feedback).

Task, Model, Feedback, and Explanations

We chose a simple model and task that a large population of non-ML experts could use to interact with and evaluate ML models. Specifically, we chose text classification as it is prevalent in real-world use cases, such as document recommendation and search. Borrowing from prior work (Kulesza et al., 2015; Settles, 2011), we used a text classification algorithm to predict the category of emails from a data set of 2,000 "hockey" and "baseball" emails from the 20 Newsgroups corpus (Lang, 1995).

For text classification, we used the Naïve Bayes model (MNB) with unigram features (Lewis, 1998) that we introduced in Chapter 3. In particular, we used the MNB classifier from the `scikit-learn`

library (Pedregosa et al., 2011). We performed standard pre-processing procedures on the emails.[2] For this first experiment, we were interested in participants' experience when interacting with a lower quality model, so we trained the classifier on only a few (16 of the 1197) labeled training emails (eight from each class). The resulting model achieved 76.5% classification accuracy on the 796 emails in the test set.

The participants were to imagine that they had been assigned to sort their boss's email inbox. They were told that they would evaluate a ML model designed to help them. Would it be worthwhile to add the model to their workflow?

For this task, we built an interface where participants review emails with the model's "hockey" or "baseball" prediction (Figure 4.1).[3] The interface either displays an explanation of the model's prediction (or not) and supports either no user feedback, feature-level feedback, or instance-level feedback.

Our explanations tell users what the model regards as important for prediction: we highlight the three words that are most influential to the class prediction for a given email, $abs(p(w|baseball) - p(w|hockey))$. This method is purposefully simple and truthful to the classifier's methodology, two guidelines for good explanations (Kulesza et al., 2013; Narayanan et al., 2018). Additionally, we choose exactly three words as explanations should include *sufficient*, but not *extra*, low-level context (Rosenthal and Dey, 2010).

For *instance-level* feedback, participants correct or confirm each classification by telling the model whether the email is about "hockey" or "baseball." For *feature-level* feedback, participants tell the model what should be important by providing the top three words they think would be most useful in classifying a given email and specifying the class with which those words should be associated.

[2]We removed non-alphabetical characters, lowercased all words, tokenized by whitespace, and dropped "From:" lines from the emails to prevent the model from training on email addresses.

[3]https://github.com/rococode/bh-classifier

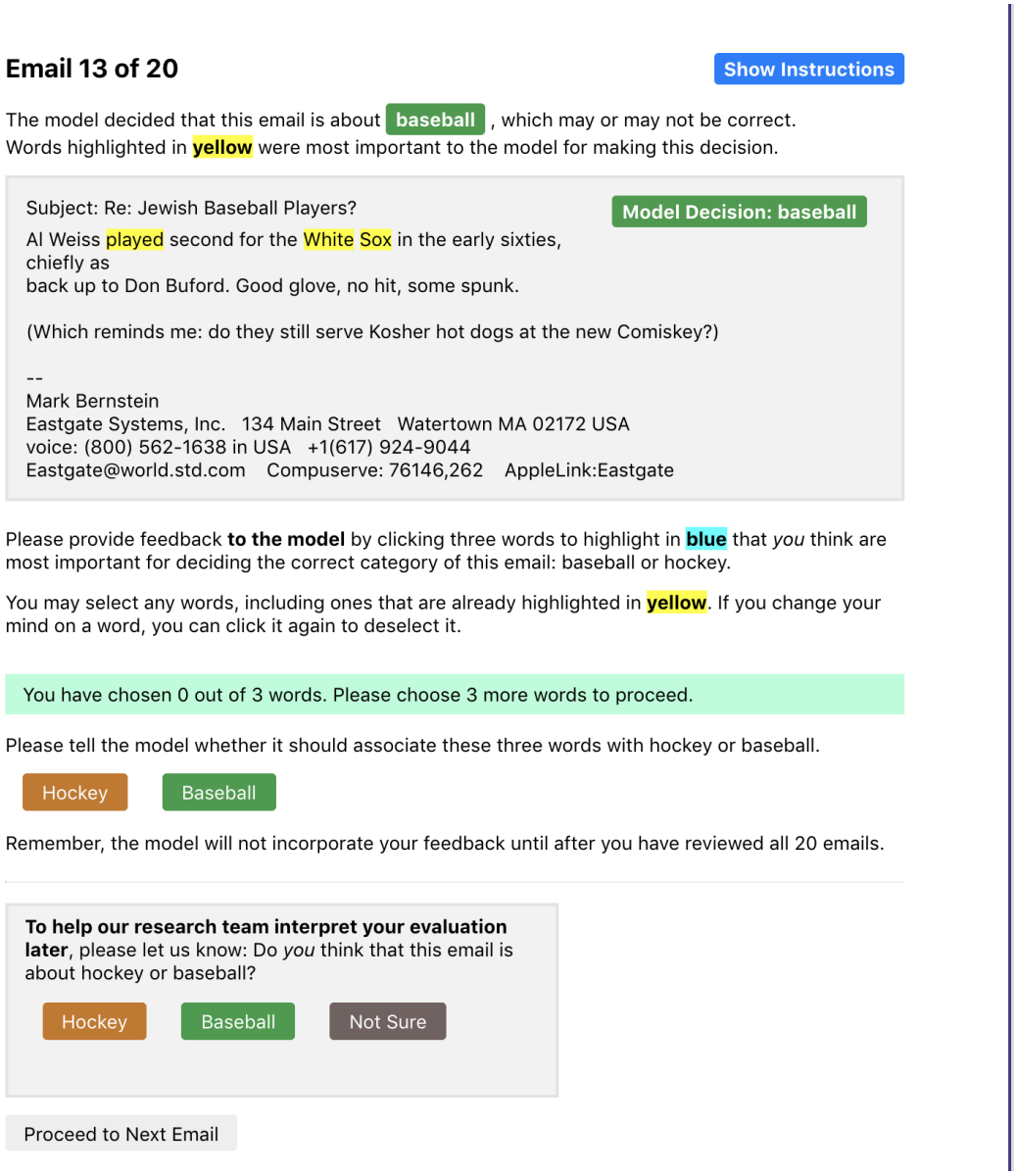


Figure 4.1: Screenshot of an email in the "interaction phase" for a participant in the feature-level feedback and explanation condition (E-F).

Participants

We recruited 180 unique participants (77 male, 102 female, and one unspecified) from Mechanical Turk,[4] requiring participants with the "Masters" qualification, located in the United States, and having completed more than 500 HITs with approval rate 98% or higher. Two participants were

[4]http://mturk.com

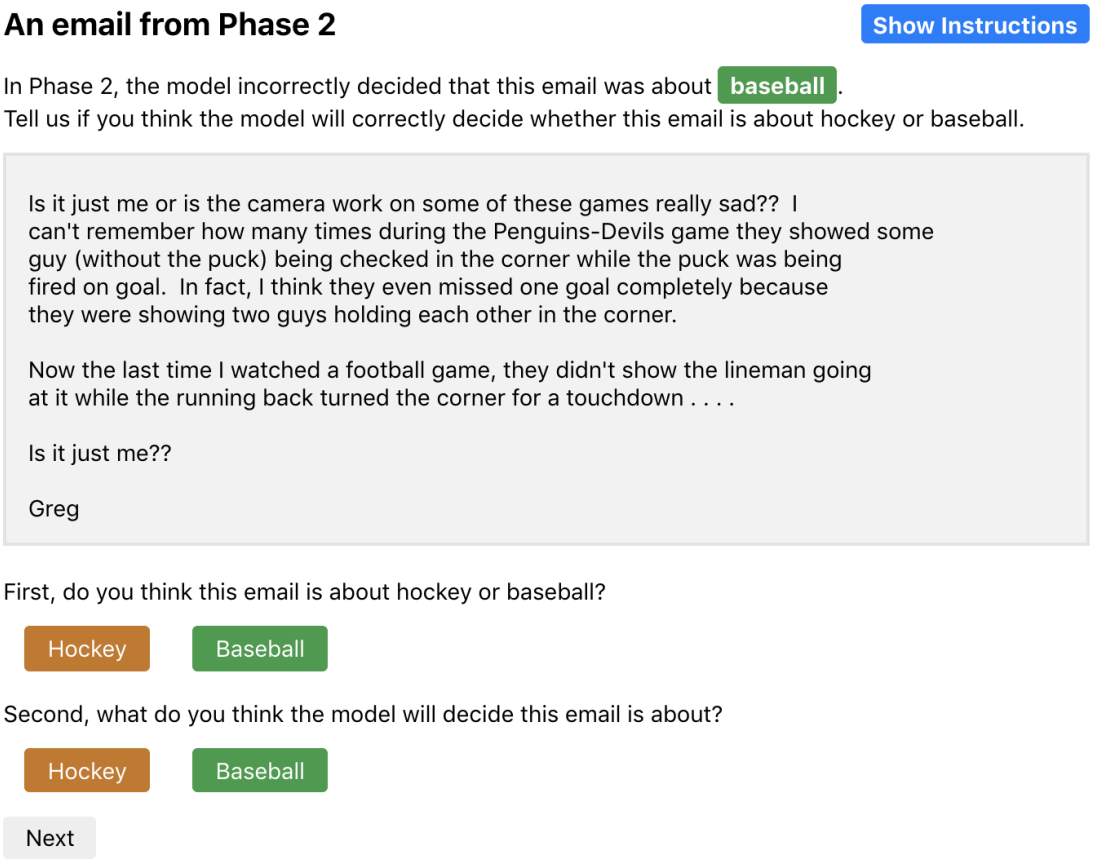


Figure 4.2: Screenshot of an email in the "evaluation phase," where participants predicted how the model would label an email that it had previously labeled incorrectly in the "interaction phase."

18–24 years old, 62 aged 25–34, 60 aged 35–44, 30 aged 45–54, 22 aged 55–64, and 4 aged 65–74. Participants rated their prior knowledge on five-point Likert scales for ML (65 had none, 67 had a little, 44 had some, four had a lot, and none had expert), hockey (15 had none, 78 had a little, 65 had some, 18 had a lot, and four had expert), and baseball (two had none, 43 had a little, 68 had some, 57 had a lot, and 10 had expert).

Procedure

Remote study sessions took on average 22.6 minutes (σ = 15.3). Participants completed three phases: (1) introduction, (2) "interaction" with the model, and (3) "evaluation" of the model.

To motivate quality work, participants were told that at least the top 50% of participants would be given a $2 bonus based on the thoroughness of their evaluations; unbeknownst to them, all ultimately received the bonus.

During the "interaction phase," participants reviewed 20 emails,[5] in randomized order per participant. The model provided a prediction ("hockey" or "baseball") for each email. Participants in the *explanation* conditions saw the model's top three words highlighted. Participants in the *instance-level feedback* conditions corrected or confirmed the model's prediction for each email, and participants in the *feature-level feedback* conditions specified their three important words for predicting the correct class. To determine whether participants knew the correct labels, as this might affect their evaluation, all participants also told us (not the model) the correct email label, with an option for "not sure."

During the "evaluation phase," participants responded to closed- and open-ended questions on satisfaction and model change expectations, including rating scales as shown in Table 4.1 paired with the follow up of "Why do you feel this way".[6] After these questions, participants were shown four "evaluation" emails and asked to predict how the model would classify them (Figure 4.2). These emails included two of the 20 from the "interaction phase" and two new ones that were similar to emails in the first 20, as measured by cosine similarity (Huang, 2008). For each email type (repeat or similar), we selected one that was previously labeled correctly and one that was previously labeled incorrectly by the model. These four emails allowed us to assess whether participants would expect the model's labels to change following the "interaction phase."

Importantly, feature- and instance-level feedback was **not** incorporated into the model during the

[5]We randomly select these 20 emails for Study 1, requiring even distribution between hockey and baseball predictions, five incorrect and 15 correct, and that emails be between 30 and 120 characters; we use the same set of emails for Study 2.

[6]An additional two rating scales of acceptable accuracy and expectations of learning are not reported on here due to space constraints and not being as directly related to our research questions.

Table 4.1: Seven-point rating scale statements for seven subjective measures. All are on a scale from "strongly disagree" to "strongly agree" aside from expected change, which is on a scale from "much worse" to "much better."

Measure	Statement
frustration	"I would feel frustrated if I were to use this model to automatically sort my boss's emails"
trust	"I would trust this model to correctly categorize my boss's emails that are about hockey or baseball"
accuracy	"The model is able to distinguish between hockey and baseball emails"
understanding	"I understand how this model makes decisions"
acceptance	"I would use this model to help me sort my boss's emails"
feedback importance	"If I were to use this model, it would be important to have the ability to provide feedback to improve it"
expected change	"If the model were now shown another set of emails, how well do you think it would categorize them?"

"interaction phase"; we reminded the feature- and instance-level participants of this with each email. This design choice isolates perceptions of explanations and feedback from how well the model incorporated that feedback. Instead, we told these participants that their feedback would be incorporated into the model *after* they had reviewed all 20 emails, so they would expect an updated model for the "evaluation phase".[7]

Study Design

This study used a 2×3 between-subjects experimental design, with factors of *Explanation*—feature (E), none (N)—and *Feedback*—feature (F), instance (I), none (N). An equal number of participants were randomly assigned to each condition.

Measures and Hypotheses

We report on seven main subjective measures, collected using seven-point rating scales (Table 4.1): three *user satisfaction* measures (frustration, trust, model acceptance), three *user perception* measures (expected model improvement, perceived model accuracy, perceived understanding of how the model works), and *desire to provide feedback* (feedback importance).

[7]However, we never incorporate feedback during the study protocol, but users were unaware as we did not show model predictions or explanations during the "evaluation phase."

While we explore the effects of feedback and explanation on user satisfaction in general, our primary user satisfaction hypothesis relates to **frustration**, as we hypothesize that users are frustrated without the ability to fix model errors exposed by explanations.

- **H1.1**: Feedback (instance-level or feature-level) reduces frustration compared to no feedback.
- **H1.2**: Explanations without feedback increase frustration compared to no explanation without feedback.

While prior work has explored effects of explanation on mental models and perceptions of quality (Bilgic and Mooney, 2005; Lim et al., 2009), we explore a new concept, **expected improvement**, or how users expect ML models to improve with or without explicit feedback. Intuitively, providing feedback should increase this expectation. Based on human behavior (Siegler et al., 2002), we also hypothesize that explanations might suggest a model is being introspective and could therefore *learn from its mistakes*.

- **H2.1**: Feedback (instance-level or feature-level) increases the user's expectation that the model will improve compared to no feedback.
- **H2.2**: Explanations increase the user's expectation that the model will improve compared to no explanation.

Data and Analysis

After disqualifying one participant who only filled out the demographics survey and another who skipped part of the post-task survey, our dataset includes 178 participants. We used separate 2×3 (*Explanation*×*Feedback*) ANOVAs with Aligned Rank Transforms for each main subjective measure—a test that is more appropriate for Likert scale data than a standard ANOVA (Wobbrock

et al., 2011). For significant main effects of feedback we used post-hoc Wilcoxon rank sum tests with continuity correction and Holm-Bonferroni adjustments. We report on all significant results, including pairwise comparisons.

We qualitatively coded the open-ended responses related to our primary measures: frustration and expected improvement. Two annotators individually read a subset of the responses to identify emergent codes, followed by a discussion period to generate a codebook. Then, the two annotators independently coded a random subset of 20 of the 178 responses; agreement was scored using Cohen's κ: $\kappa = .93$ (raw agreement: 95%) for frustration responses and $\kappa = .88$ (90%) for expected improvement responses. We refer to participants in this experiment with a lower quality model as LP1–LP178.

4.1.2 Results

Figures 4.3 and 4.4 show the rating scale responses for the seven main subjective measures by condition. Participants expected the model to improve, and they expected more improvement with feedback. Participants also thought the ability to provide feedback was important. Explanations hurt subjective satisfaction (frustration, trust, and acceptance ratings), while feedback helped. Participants were commonly frustrated by the model's low quality, and this was accentuated by explanations.

To judge user comfort with the task and dataset, we asked participants to tell us (i.e., the researchers . . . not the model) whether they thought each email was about hockey, baseball, or whether they were unsure. Participants did well: 91% of the 3,580 answers reported to us were correct, while 8% were "not sure" and only 1% were incorrect. In the following sections, we provide detailed results regarding satisfaction, expectations, perceptions, feedback quality, and users'

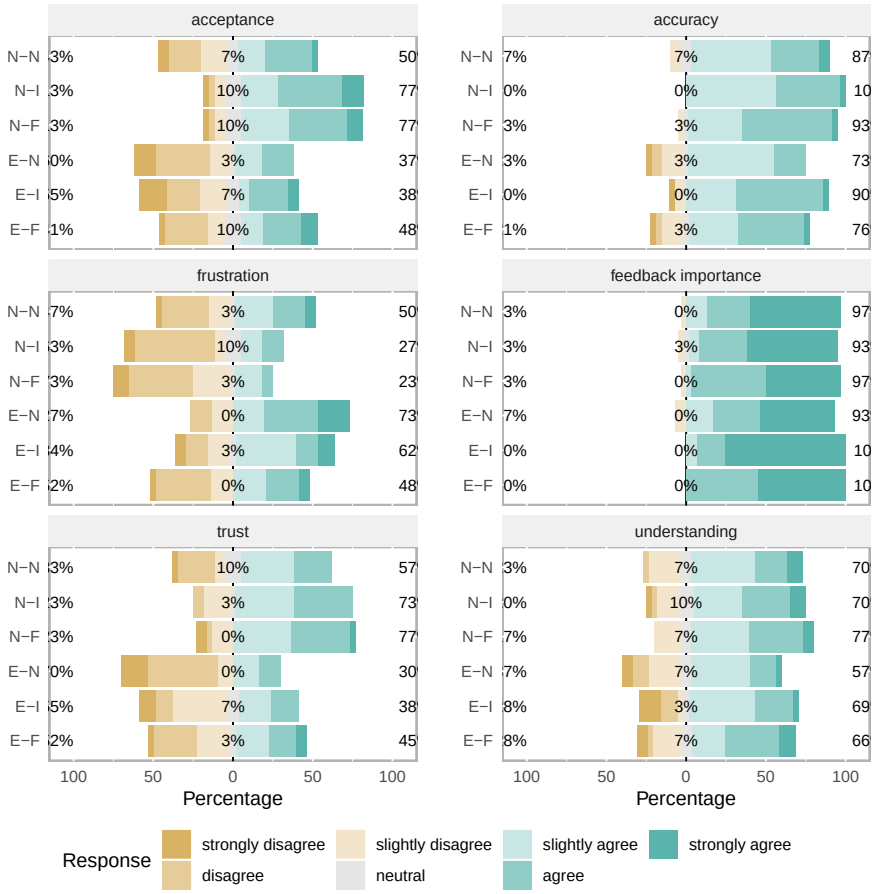


Figure 4.3: Study 1 seven-point rating scale responses for the main subjective measures (except expected change) from "strongly disagree" to "strongly agree." Responses reported by condition. For each measure, no explanation (N-) conditions are on the top (-N is with no feedback, -I is with instance-level feedback, and -F is with feature-level feedback) and feature explanation (E-) conditions are below Feedback (-I, -F) positively, and explanation (E-) negatively impact satisfaction measures (left).

desire to provide feedback.

User Satisfaction

Participants were neutral on average, but with high variability across conditions, for each of the user satisfaction measures: frustration ($M = 3.9$ of 7, $\sigma = 1.8$), trust ($M = 4.1, \sigma = 1.7$), and whether they would use the system (acceptance) ($M = 4.3, \sigma = 1.9$). Feedback significantly improved satisfaction, but explanations hampered it. Open-ended responses suggested that the low model

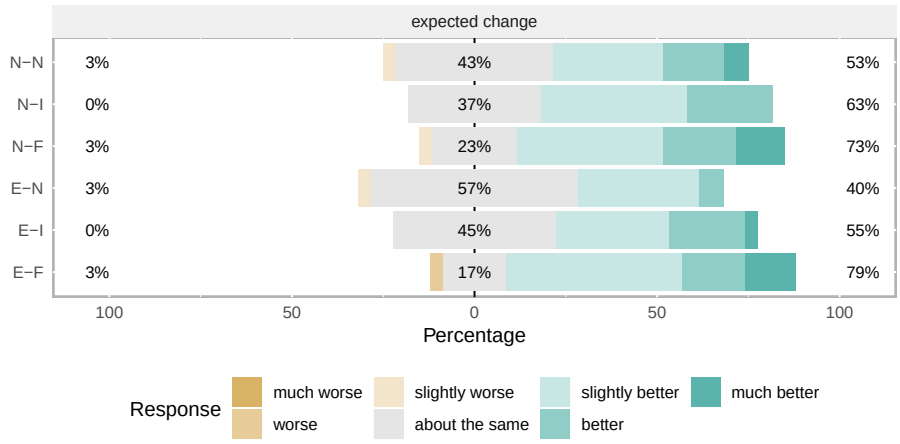


Figure 4.4: Study 1 participant responses for the subjective expected change measure reported by condition. Participants in general expected the model to improve. (See Figure 4.3 for a description of y-axis labels).)

quality—highlighted by explanations—frustrated participants.

Explanations increased frustration, while support for feedback reduced it

Participants who received explanations were more frustrated than those who did not; this difference was significant (main effect of *Explanation*: $F_{1,172} = 20.05, p < .001$). *Feedback* also significantly impacted frustration (main effect: $F_{2,172} = 7.92, p < .001$). Posthoc pairwise comparisons showed that no feedback resulted in significantly higher frustration than instance-level and feature-level feedback (both comparisons $p < .05$); this supports **H1.1** for frustration, which stated that feedback would reduce frustration. The interaction between *Explanation* and *Feedback* was not significant ($F_{2,172} = .06, p = .094$); thus, **H1.2** is only partially supported by the main effect of *Feedback*.

Many participants were frustrated by low quality, which was highlighted by explanations

We coded participants' open-ended reasons for their frustration ratings, resulting in six codes. Participants felt the model: was "not good enough" (40% of the 178), was "good enough" (27%), helped "save time" (13%), required "user review" of the decisions (11%), was "able to improve" (3%), or "other" reasons (6%).

Confirming the rating scale data, more participants who got explanations (81% of 89) thought the model was "not good enough" compared to those who did not get explanations (only 26% of 89). Participants who got explanations (E-) often expressed their frustration in terms of the model's bad reasoning, such as *"I don't think it highlighted the best words in many cases"* (LP3, E-I), while participants who did not see explanations (N-) were more likely to comment on the model's shortcomings in terms of accuracy, *"it made too many mistakes"* (LP175, N-N).

Less frustrated participants felt the model was "good enough" or would "save time", saying, for example, *"it would be much easier than sorting through them myself"* LP132 (E-N).

Trust and acceptance were reduced by explanations and increased by feedback

Reflecting the frustration findings, trust was significantly impacted by *Explanation* ($F_{1,172} = 14.57$, $p < .001$); participants who received explanations trusted the model less those who did not. There was also a significant main effect of *Feedback* on trust ($F_{2,172} = 4.27, p = .015$). Posthoc pairwise comparisons showed that both instance- and feature-level feedback increased trust compared to none (both comparisons $p < .05$). The *Explanation* × *Feedback* interaction was not significant ($F_{2,172} = .15, p = .863$).

Similarly, *Explanation* significantly impacted acceptance ($F_{1,172} = 19.49, p < .001$), where par-

ticipants who saw explanations accepted the model less than those who did not. *Feedback* also significantly impacted acceptance ($F_{2,172} = 3.76, p = .025$). Posthoc pairwise comparisons showed that feature-level feedback resulted in higher model acceptance compared to none ($p < .05$). The interaction between *Explanation* and *Feedback* was not significant ($F_{2,172} = .97, p = .38$).

User Expectations for and Perceptions of the Model

Participants provided subjective ratings of their model expectations and perceptions (Figure 4.3 and Figure 4.4). On average, they expected the model to improve ($M = 5.2, \sigma = .9$), thought it worked fairly well ($M = 5.2$, $\sigma = 1.1$), and were neutral regarding whether they understood how it works ($M = 4.7$, $\sigma = 1.6$). We also examined expectations through participants' *simulated model predictions*: how they thought the model would label the four evaluation emails shown at the end of the study. As detailed below, feedback caused participants to think the model was more accurate and would improve, but explanation did not. Moreover, some participants who were in the no feedback conditions thought the model would *self correct*.

Feature-level feedback increased expected improvement compared to no feedback

Feedback significantly increased users' expectations ($F_{2,172} = 5.29, p = .006$); posthoc comparisons showed that feature-level feedback raised expected improvement compared to no feedback ($p < .05$), partially supporting **H2.1**. The main effect of *Explanation* was not significant ($F_{1,172} = 1.28, p = .259$) (opposing **H2.2**), nor was the *Explanation* × *Feedback* interaction effect ($F_{2,172} = .42, p = .656$).

A substantial portion of participants expected model corrections, even without feedback

Across all three feedback conditions, about half or more of participants expected the model would improve (i.e., rating > 4): 76.3% of 59 who had feature-level feedback, 59.3% of 59 who had

instance-level feedback, and even 47% of 60 who had no feedback.

Participants' predictions about how the model would label the next four same/similar "evaluation" emails reflected this strong expectation of improvement. Recall that we ask users to predict how the model will label four additional emails, two that were the same and two that were similar (by cosine similarity) to emails in the original set. One of each pair of emails (same or similar) was previously labeled incorrectly and the other was labeled correctly in the original set. Our goal is to understand whether participants expect the model to change, and whether this is affected by feedback. Intuitively, participants who correct the model (instance-level feedback) should expect the model to label correctly the previously incorrect "same" email, and participants who provide feature-level feedback should expect the model to label correctly both the previously incorrect emails (same and similar). Participants who do not provide feedback, should not expect the model to change. For the previously correct emails, participants thought the model would now be incorrect in only 4 of 712 instances, and each of these was for a "similar" email rather than the email that was exactly the "same" as in the initial set of 20.

For the previously incorrect emails, "similar" and "same" follow a similar pattern (we do not see a difference in how participants predict the model will label these emails based on the type of feedback they provided), so we focus on the "same" email to provide a straightforward assessment of whether participants think the model will improve. Most (82%) of participants who provided feedback (N = 118) thought the model would get the previously incorrect email correct (Table 4.2), which is not surprising given that they had spent time trying to improve the model. More surprising, however, is that 53% of participants in the no feedback condition (N = 60) thought the model would somehow correct itself.

Table 4.2: Percentage of Study 1 participants (N=178) by condition during the "evaluation phase" who thought the model would now correctly label an email it had previously labeled incorrectly. Many participants in the *no feedback* conditions thought the model would self correct.

	Feedback		
Explanation	None	Instance	Feature
None	63%	80%	90%
Feature	43%	86%	73%

Participants described the model improving from their feedback or learning from its mistakes

We coded participants open-ended reasons for their expected change ratings, resulting in nine codes. Participants who felt the model would improve explained that it got "feedback" (29%), was capable of "self learning" (20%), was "high quality" (5%), or showed codeevidence of improvement (1%). Those who felt it would not improve cited that it received "inadequate feedback" (14%), showed "no evidence of improvement" (11%), had "nothing to learn from" (6%) or was of "low quality" (5%). 9% of participants gave "other" reasons.

Interestingly, of the 60 participants who did not provide feedback (-N), 17 (28%) still expected the model to learn from its mistakes, such as, *"it would take what it did wrong, learn from it, and apply it in future trials"* (LP141, N-N), or reported other misconceptions, including, *"these programs get better as they function and learn algorithms"* (LP154, E-N). In fact, only 13% of the 60 participants who did not provide feedback *correctly* identified that the model would not improve as it had "nothing to learn from", like, *"if it still used the same words to try to identify the correct sports emails, then it would still make the same amount of errors"* (LP87, E-N).

Feature-level feedback reduced perceived accuracy compared to no feedback

Overall, participants thought the system worked fairly well, giving it an average accuracy rating across all conditions of 5.2 out of 7 (σ = 1.1). However, counter to our other user experience measures, feature-level feedback had a negative effect on perceived accuracy. There was a significant main effect of *Feedback* on perceived accuracy ($F_{2,172} = 4.72, p = .010$), with posthoc pairwise comparisons showing that feature-level feedback reduced perceived accuracy compared to no feedback ($p < .05$). Neither the main effect of *Explanation* nor the *Explanation* × *Feedback* interaction effect were significant (respectively: $F_{1,172} = 1.59, p = .209$; $F_{2,172} = 2.20, p = .114$).

Quality of and Desire for User Feedback

Participants thought being able to provide feedback was important (M = 6.4 out of 7, σ = .9), regardless of condition (Figure 4.3); there were no significant main or interaction effects on this measure. However, do the experimental conditions impact feedback *quality*? To answer this question, we applied participants' feedback to the model after the study.

Feedback improved the model, regardless of explanation

We incorporated instance-level feedback by including the 20 emails labeled by the participant as additional training emails. To incorporate the feature-level feedback, we adjusted the classifier's weight for each word provided by the participant: the word weight was both increased by 20% for the specified class and decreased by 20% for the opposite class.

The feature-updated models were 86.2% accurate on average (σ = 2.7%), which is a 9.7 percentage point improvement over the initial low quality model. In comparison, the instance-updated models were 83.6% accurate (σ = 1.4%)—a 7.1 percentage point improvement. Instance and fea-

ture model improvements were similar regardless of whether the participants saw an explanation (difference in accuracy < .2%)

Participants did not agree with the words the model thought were important

The 59 participants who gave feature-level feedback highlighted a total of 3,533 words. Regardless of whether explanations were shown or not, we compared the model's top three words for each email (i.e., the words the model would have highlighted) to the three words selected by the participant. Most (76.9%) of the participants' words were not in the model's top set. This disagreement is likely due both to the model's low quality and because the explanation method can highlight words that are probable for the non-predicted class. Participants with explanations were more likely to reuse the model's words (28% of selected words overlapped with the model's) than the 30 participants who did not see explanations (21%).

4.1.3 Summary

Explanations significantly increased frustration, while feedback—especially feature-level—significantly decreased it (partial support for **H1.1** and **H1.2**). There were similar patterns for other user satisfaction measures (trust and acceptance). Therefore, the worst combination was explanation without feedback, and the best was no explanation with feedback. Open-ended responses suggested that frustration was primarily due to the low model quality exposed by explanations, and *not* their inability to provide feedback, as we had hypothesized; although, ability to provide feedback did temper some of the frustration. This general dislike for explanations confirms prior work where user perceptions were negatively impacted by explanations that exposed flaws and limitations (Cai et al., 2019). While this may seem inconsistent with our hypothesis at first blush, an alternate interpretation is that explanations can improve satisfaction *so long as users have a*

means for feedback.

Feedback also significantly increased expectations of model improvement, as hypothesized in **H2.1**, but particularly for feature-level feedback opposed to none. We did not find impacts of explanation on expected change, in contrast to **H2.2**. Also, somewhat surprisingly, most participants expected the model to improve, including many who did not provide feedback. We discuss this and general misconceptions regarding ML models in Chapter 4.3.

4.2 Study 2: Understanding Explanations and Feedback with a High Quality Model

In Study 1, expectations rose with feedback but not explanations and satisfaction fell with explanations but rose with feedback. As the Study 1 model's low quality appeared to overwhelm participants' subjective ratings, an additional study had a higher quality model. While we expected participants to be more satisfied with the higher quality model (e.g., observed and stated model accuracy can affect users' trust (Yin et al., 2019)), we retained the Study 1 hypotheses regarding our primary measures (frustration and expected change).

4.2.1 Method

This experiment was exactly the same as Study 1 with the exceptions described here. We trained the MNB classifier on 200 labeled training emails (100 from each class), which resulted in 94.4% accuracy on the test set. This model predicted the correct label for 18 of the 20 emails in the interaction phase. As in Study 1, we chose four emails for the evaluation phase (two "same" and two "similar"), but because of the higher accuracy of the model in Study 2 there were no available

emails that were "similar" to ones the model labeled incorrectly in the evaluation phase; thus, both of the "similar" emails were similar to previously correct ones.

As in Study 1, we recruited 180 participants (99 female, 78 male, 3 unspecified). Two participants were aged 18–24 years old, 46 aged 25–34, 66 aged 35–44, 43 aged 45–54, 16 aged 55–64, and 6 aged 65–74. Participants had varied prior knowledge of machine learning (63 participants had none, 65 had a little, 50 had some, two had a lot, and none had expert), hockey (23 had none, 64 had a little, 58 had some, 25 had a lot, and none had expert), and baseball (12 had none, 37 had a little, 66 had some, 54 had a lot, and 11 had expert).

Study sessions took on average 22.8 minutes (σ = 14.6), and we used the same measures and data analyses as in Study 1. Our dataset included all 180 participants. We used the Study 1 codes to code the open-ended responses for frustration and expected change. We refer to participants as HP1–HP180.

4.2.2 Findings

Figure 4.5 and 4.6 show the rating scale responses for the seven main subjective measures by condition. Overall, participants were less frustrated with the high quality model than the low quality one (Figure 4.3). The interaction between explanation and feedback was significant for other subjective measures: trust and acceptance. As in Study 1, feedback impacted expected change but explanation did not, and participants expected the model to improve and wanted the ability to provide feedback.

Regarding task difficulty, participants again performed well: 92% of their 3,600 answers to us were correct, while 7% were "not sure" and only 1% were incorrect. We provide detailed results regarding satisfaction, expectations and perceptions, and quality and desire for feedback in the

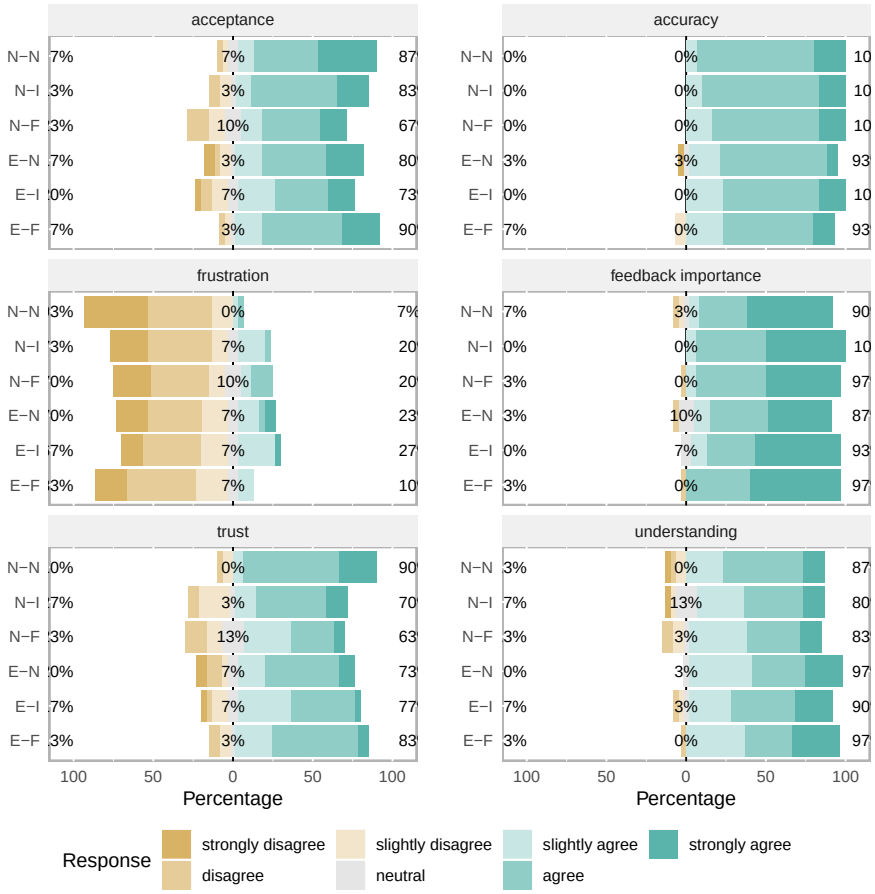


Figure 4.5: Study 2 responses by condition for the main subjective measures (except expected change). In general, participants were more satisfied, but trust suggests nuance (e.g., comparing E-N to N-N, without feedback, explanation has a negative impact). (See Figure 4.3 for a description of y-axis labels).

following sections.

User Satisfaction

Overall, frustration was lower ($M = 2.6$ of 7, $\sigma = 1.5$) compared to the low quality model in Study 1 ($M = 3.9$, $\sigma = 1.9$). Perhaps accordingly, there were no significant main or interaction effects on frustration. Open-ended responses suggest explanations exposed the high quality model's *good* behavior. Trust and acceptance ratings were also relatively high compared to Study 1: 5.1 out of 7 on average for trust ($\sigma = 1.5$) and 5.4 for acceptance ($\sigma = 1.5$) here compared to 4.1 for trust

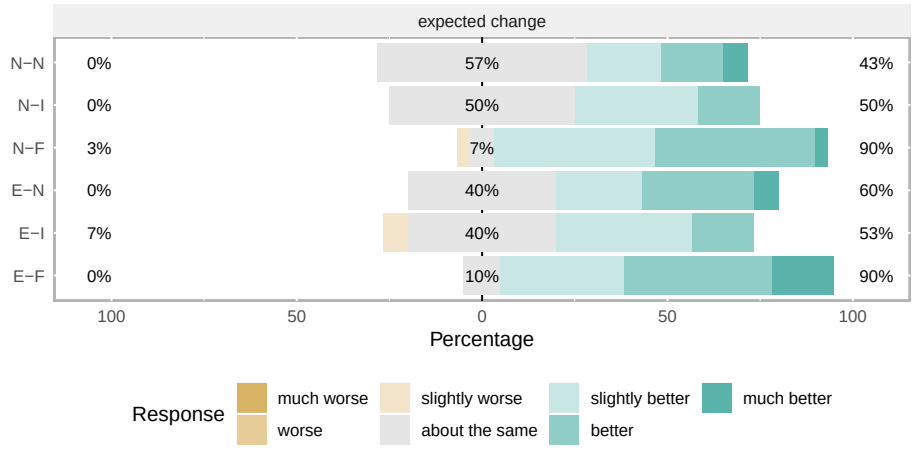


Figure 4.6: Study 2 responses for the expected change measure by condition, showing that in general participants expected improvements (green bars), but more in feature-level feedback conditions (E-F and N-F). (See Figure 4.3 for a description of y-axis labels).

(σ = 1.7) and 4.3 for acceptance (σ = 1.9) in Study 1. The interaction between explanations and feedback on these measures was significant.

Trust and acceptance were affected by the combination of explanations and feedback

Neither explanation nor feedback had a clear effect on trust; the main effects of *Feedback* ($F_{2,174}$ = 2.59, p = .078) and *Explanation* ($F_{1,174}$ = 2.00, p = .159) were not significant. However, the interaction between *Explanation* and *Feedback* was significant ($F_{2,174}$ = 5.69, p = .004), meaning that certain combinations of explanations and feedback impacted trust.

From the responses (Figure 4.5), when feature-level feedback is requested, not providing an explanation might decrease trust (N-N compared to N-F). And, without feedback, explanation might decrease trust (N-N compared to E-N). After a Holm-Bonferroni correction, only the former posthoc pairwise comparison was significant: participants trusted the model more with neither feedback nor explanation compared to a model with feedback but no explanation ($p < .05$).

Acceptance shows a similar pattern: while there is no clear effect of either explanation or feedback, some combinations do; the *Explanation* × *Feedback* interaction was significant ($F_{2,174}$ = 4.11, p =

.018), while the main effects of *Feedback* ($F_{2,174} = 1.23, p = .295$) and *Explanation* ($F_{1,174} = .036, p = .850$) were not. While Figure 4.5 shows similar trends for acceptance as for trust, no posthoc pairwise comparisons were significant after a Bonferroni correction, so further work is needed to explore this relationship.

Explanations may have shown participants that the model was behaving properly

Participants gave lower frustration ratings than in Study 1 (Figure 4.6); they said the model was "good enough" (49% of all participants) or would "save time" (23%). Only 15% of participants felt the model was "not good enough," that is, not of an acceptable accuracy for the task.

In Study 1, explanations exposed issues with the model's highlighted words, resulting in 81% of the 89 participants who had received explanations in that study thinking the model was "not good enough." Study 2 responses were the opposite: 80% (of 90) participants who saw explanations thought the model was "good enough," and explicitly described good model reasoning, such as *"...I was able to see the reasoning from the machine and I agreed with it most of the time"* (HP139, E-F). For Study 2 participants who did not see explanations, only 65% (of 90) felt the model was "good enough," emphasizing how explanations can improve perceptions of model quality with a higher quality model.

User Expectations for and Perceptions of the Model

Figure 4.5 and Figure 4.6 show responses for subjective rating scales regarding expectations and perceptions of the model. On average, participants expected the model to improve ($M = 5.0$, $\sigma = 1.0$) and thought they understood how it labels emails ($M = 5.5$, $\sigma = 1.2$) and thought it worked well ($M = 5.9$, $\sigma = .8$).

As detailed below, feature-level feedback caused participants to think the model would improve,

and explanation yielded higher perceived understanding. Neither explanation nor feedback had an impact on perceived accuracy. Open-ended responses suggest misconceptions regarding how ML models change over time, providing further explanation for why a substantial portion of participants, regardless of condition, expected the model to improve (Figure 4.6).[8]

Feature-level feedback increased expected improvement

As in Study 1, *Feedback* significantly impacted expected change ($F_{2,174} = 15.84, p < .001$). Posthoc pairwise comparisons showed that feature-level feedback resulted in higher expected improvement than instance feedback or none (both comparisons $p < .05$). *Explanation* did not have a significant impact on expected change ($F_{1,174} = .79, p = .375$) nor did the *Explanation* × *Feedback* interaction ($F_{2,174} = 1.41, p = .246$).

Participants described misconceptions for how ML changes over time

Participants gave similar reasons for expecting model change as in Study 1. 27% of all participants credited the "feedback" they provided while 19% suggested the model was "self learning". Many participants noted similar misconceptions, including, *"my understanding is these sorts of things just get better at what they do the more they do them"* (HP84, E-N) and, *"it learns with each new experience, and I choose the word 'experience' intentionally as the machine gains consciousness"* (HP62, N-I).

Similar to Study 1, 21 (12%) participants thought their feedback either was not good enough or they did not provide enough of it ("inadequate feedback"). 17 provided instance-level feedback (compared to three who provided no feedback and three who provided feature-level feedback), and suggested that they would have preferred to tell the model *why* it was wrong. For example,

[8]We do not report on participants' simulated model predictions due to space and because trends are in line with the rating data.

HP128 (E-I) said, *"simply telling it that it was wrong may make it less accurate, but it is unlikely to make it more accurate without knowing how it made its mistake."*

Explanations increased perceived understanding

Explanation significantly impacted perceived understanding ($F_{1,174} = 3.92, p - 0.49$). Participants thought they understood the model more when given an explanation (Figure 4.5). Neither the main effect of *Feedback* ($F_{2,174} = .13, p = .876$) nor the *Explanation* × *Feedback* interaction effect were significant ($F_{2,174} = .53, p = .591$).

Quality and Desire for User Feedback

Like in Study 1, participants wanted the ability to provide feedback ($M = 6.3$ of 7, $\sigma = 1.0$), regardless of condition (Figure 4.5). There were no significant main or interaction effects on this measure. But how useful was their feedback when the model was high quality?

Feedback provided only minor improvement

We incorporated participant's feature-level and instance-level feedback into the model. While the updated models in Study 1 greatly improved, in Study 2 they did not. The feature updated models averaged 95.8% accuracy ($\sigma = .8\%$), only a 1.4 percentage point improvement over the initial high quality model. The instance updated models had 95.1% accuracy ($\sigma = .5\%$; a .7 percentage point improvement). As in Study 1, instance and feature model improvements were similar regardless of whether the participants saw an explanation (difference in accuracy < .2).

Participants agreed more with the high quality model's words

Participants provided 3,589 words as feature-level feedback. Participants were similarly likely to provide new words (1,942) as reuse model words (1,647), unlike Study 1 participants who reused

less than 25% of the model's words. The 30 participants shown explanations reused provided words (52% overlap of their words to the model's important words), more than the 30 who did not see explanations (40%).

4.2.3 Summary

Neither feedback nor explanation impacted frustration, which was generally lower than in Study 1. For other user experience measures, there were no main effects either, although significant interaction effects on trust and acceptance suggest nuance in how explanations and feedback impact each other. As with the low quality model, feature feedback significantly increased expected change, this time over both instance and no feedback (confirming partial support for **H2.1**), but explanation did not have an effect. Again, participants generally thought the model would improve.

4.3 Discussion

We relate our findings to prior work and provide design recommendations for interactive and explainable ML systems. We also discuss limitations and extensions to more complex tasks, models, explanations, and feedback mechanisms.

Users want the opportunity to provide feedback, and in particular, provide more than just labels

In both studies and all conditions, participants felt strongly that the *opportunity* to provide feedback was important; however, this does not tell us how often or whether users will provide such feedback in practice. Although, successful commercial projects, such Common Voice,[9] exemplify

[9]https://voice.mozilla.org/en

that users might be willing to spend time improving models.

Our studies provide additional evidence for how different levels of feedback impact user behavior and subjective response. In particular, we confirmed Amershi et al. (2014)'s recommendation that "people naturally want to provide more than just data labels" to ML models. With both the low and high quality models, only those participants who told the model what words were important (i.e., provided feature-level feedback) and not those who corrected or confirmed the model's predictions (i.e., instance-level), expected the model to improve more than participants in the no feedback condition. Similarly, some participants who provided instance-level feedback described their feedback as inadequate in open-ended responses. Finally, not only was feature-level feedback better received by participants, for the low quality model it also improved accuracy more than instance-level feedback. This ability of non-ML expert participants to improve the models in our study beyond just labeling data supports the goals of machine teaching (Wall et al., 2019).

Explanations can reveal model flaws, which users desire to fix

Displaying uncertainty scores for model predictions negatively impacts users' perceptions (Lim and Dey, 2011); similarly, for the low quality model, explanations were frustrating, precisely because they exposed flaws, including *uncertainty* in the model's reasoning. Because feedback reduced frustration, the most frustrating combination of explanations and feedback for the low-quality model was thus a situation with explanations but no opportunity for feedback. Indeed, no explanations and no feedback may be the least frustrating design option; however, this combination would inherently limit the model's *potential* performance, and likely result in disuse over time. In such cases, explanations provide insight to how to solve model errors (Kulesza et al., 2015). Therefore, for similar models and tasks, when the model quality is low, feedback should

be supported alongside explanations.

Explanations and feedback complement each other

For the high quality model, explanations increased understanding and may have exposed model strengths. But, models are rarely perfect, and participants wanted the opportunity to provide feedback to improve models. Therefore, providing explanations without means for feedback may reduce satisfaction. Future work should explore this relationship between explanations and feedback in more detail. Feedback alone is not always positive either: asking participants for feature-level feedback without providing explanations reduced trust compared to when explanations were provided. Users may not want to provide detailed feedback without understanding why it is needed or how best to help the model. Therefore, to improve satisfaction, similar systems should neither request detailed feedback without explanation nor provide explanation without some means for feedback.

Preconceived ML expectations should be managed. Whether from prior experience or general misunderstanding, users may have misconceptions about whether and how much models can improve. In our experiments, many participants expected the model to improve regardless of whether they provided feedback. Open-ended responses provide insight: participants described their understanding that ML models *"get better as they function and learn algorithms"* (LP154, E-N), or even *"gain consciousness"* (HP62, N-I).

Interactive ML designers must ensure that these expectations are managed, such as by clarifying how model feedback is treated or what accuracy the model could achieve. Or if feedback is not supported, designers should ensure users do not think they are in some way providing feedback to the model.

4.3.1 Limitations & Future Work

Generalization from a tightly scoped domain. Our aforementioned findings are made in a tightly scoped domain, with a simple model and task (categorizing sports' emails). While this constrained setting provides a necessary first step in illustrating the relationship between explanations and feedback—it is simple enough to support a controlled experiment for non-expert users, and common enough in IML research to be compared to past studies—our findings should be generalized with caution. For example, explanations and feedback mechanisms in our studies were simple and intuitive. However, explanations in other domains, such as image classification, can be confusing or misleading (Adebayo et al., 2018).

We hypothesize that even for more complex models or subjective tasks, if users understand how models work and how they can better improve them, they will want the *opportunity* to do so and may be frustrated if such feedback is restricted. However, the *degree* of their frustration would likely vary along with their actual desire and ability to provide feedback in more realistic settings. All are likely affected by task and model complexity, task importance (and therefore user motivation), and domain expertise. Would users be eager to provide feedback (in lieu of abandonment) in an imperfect self-driving car? Would they be less able to detect systems' mistakes for more subjective tasks? Future studies should further explore the relationship between feedback and explanation.

The effect of explanation and feedback mechanisms. Motivated by prior work (Kulesza et al., 2013; Narayanan et al., 2018), our simple and truthful method chooses the top three overall important words for classification. This explanation method inherently exposes system uncertainty in the low quality model, as the method highlights words that have high probability (i.e., relative counts) for either hockey or baseball (not just the predicted class), and the lower quality model

has higher variance, since it has seen fewer examples, and therefore fewer total words. Future work could explore the effects of different, more advanced, explanation types and feedback mechanisms. For example, global explanations (e.g., differential explanations (Lakkaraju et al., 2019)) might be equivalently faithful, while better counteracting the user experience concerns. "Human-like" explanations may increase expectations of improvement, as human-like characteristics in ML systems can cause users to believe systems will act rationally or take responsibility for their actions (Höök, 2000). Furthermore, explanations that expose when models are right for the wrong reason might further increase frustration if adequate feedback is not allowed, as users would be unable to rectify apparent mistakes. For this case, to align the information received by the model and the user, feedback mechanisms should be changed accordingly.

Finally, we could have used an alternative to MNB, such as logistic regression. Logistic regression is a discriminative model which learns a direct map from input documents to class labels (Ng and Jordan, 2002). In logistic regression, feature (or term) importance to a classification can be obtained from a learned weight vector, and these weights can be used to highlight important terms to the classification. Logistic regression also supports techniques, such as regularization, to prevent overfitting to the training data.

4.4 Conclusion

In this chapter, we presented two controlled experiments to understand how the combinations of explanation and feedback affect users' satisfaction and expectations of improvement of high and low quality ML models. We found that, for the simple models and task of our studies, when possible explanations and feedback should be provided together: (1) while explanations negatively impacted user satisfaction with the low quality model, they can show users how to fix models,

and support for feedback had positive effects; and (2) for the higher accuracy model, requesting detailed feedback without explanations reduced trust. Additionally, regardless of model quality, feature-level feedback increased expectations that models would improve, yet users generally expected model correction, regardless of whether they provided feedback or received explanations.

The remainder of this dissertation builds on the studies in this chapter in two ways: First, this chapter focused on user experience with supervised ML, where much of the other prior research in transparency and control for IML has focused. And, we explored a simple task, model, and application of explainability. In the following chapters, we are interested in exploring user experience with transparent, interactive system in complex, real-world scenarios. Therefore, we switch the focus to control and transparency in unsupervised ML, which supports more subjective tasks, complex models, and different cases of explanations. In particular, we study user interaction with unsupervised ML using the case of interactive topic modeling. While many different topic visualizations exist, there are limited studies on their effectiveness, specifically which best supports users in understanding topics. Therefore, in Chapter 5, we evaluate different topic visualizations (or explanations) for interpretability. We then build a novel interactive topic modeling system based on user-preferred explanation (visualization) and feedback mechanisms and evaluate it with end users in Chapters 6 and 7. Second, while this chapter explored control in terms of whether and how it was provided, in Chapters 6 and 7 we explore different dimensions of control—instability and adherence—IML characteristics that might be exposed when users interact with transparent systems.

Chapter 5: Optimal Topic Visualizations for Interpretability: a Novel Visualization and Comparative Study[1]

Chapter 4 explored the interactions of explanations (control) and feedback (transparency) for a simple, supervised ML case. In this and the following chapters, we switch the focus to unsupervised ML to explore these interactions in more detail and under different settings, such as more subjective tasks and complex models. In particular, we explore a particular case of unsupervised ML: topic modeling, which is a common technique for organizing and understanding large text corpora.

As introduced in Chapter 3.2.1, probabilistic topic models are important tools for indexing, summarizing, and analyzing large document collections by their themes. However, promoting end-user understanding of topics remains an open research problem. In this chapter, we implement a novel topic visualization technique: topic-in-a-box, and compare labels generated by users given three other topic representations—word lists, word lists with bars, and word clouds—against each other and against automatically generated labels (Figures 5.1 and 5.2). Our basis of compari-

[1]The work in this chapter was published as "Alison Smith, Tak Yeon Lee, Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Niklas Elmqvist, and Leah Findlater. Evaluating Visual Representations for Topic Understanding and Their Effects on Manually Generated Labels. In Transactions of the Association for Computational Linguistics (TACL), 2017 (Smith et al., 2017)" and "Alison Smith, Jason Chuang, Yuening Hu, Jordan Boyd-Graber, and Leah Findlater. Concurrent Visualization of Relationships Between Words and Topics in Topic Models. In ACL workshop on Interactive Language Learning, Visualization, and Interfaces, 2014 (Smith et al., 2014a)."

son is participants' ratings of how well labels describe documents from the topic. Our study has two phases: a labeling phase where participants labelled visualized topics and a validation phase where different participants selected which labels best described the topics' documents. Although all visualizations produced similar quality labels, simple visualizations such as word lists allowed participants to quickly understand topics, while complex visualizations took longer but exposed multi-word expressions that simpler visualizations obscured. Automatic labels lagged behind user-created labels, but our dataset of manually labeled topics highlights linguistic patterns (e.g., hypernyms, phrases) that can be used to improve automatic topic labeling algorithms.

5.1 Background: Topic Representations

This chapter compares four topic representations (or visualizations) for end user interpretability. Although every word has some probability for every topic, $P(w|t)$, visualizations typically display only the top n words. The cardinality may interact with the effectiveness of the different visualization techniques (e.g., more complicated visualizations may degrade with more words). We used $n \in \{5, 10, 20\}$ for the study in this chapter. Figures 5.1 and 5.2 show each topic visualization for the three evaluated cardinalities (or number of words displayed) for the same topic (5, 10, and 15).

Word list

The most straightforward topic representation is a list of the top n words in the topic, ranked by their probability. In practice, topic word lists have many variations. They can be represented horizontally (Gardner et al., 2010; Malik et al., 2013) or vertically (Chaney and Blei, 2012; Eisenstein et al., 2012), with or without commas separating the individual words, or using set notation (Chaney and Blei, 2012). Nguyen et al. (2013) added the weights to the word list by sizing

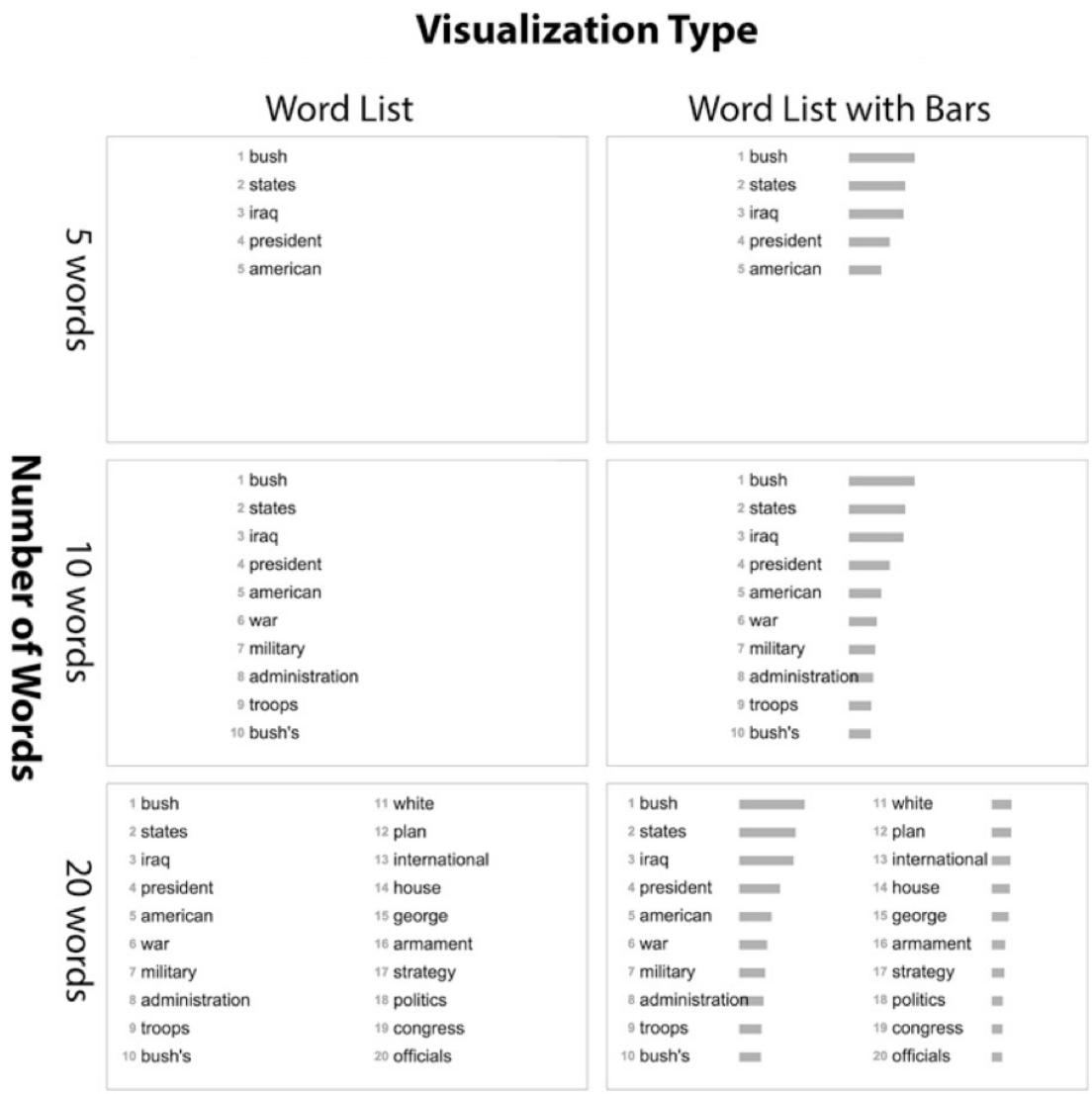


Figure 5.1: Examples of the six of the twelve experimental conditions, each a different visualization of the same topic about the George W. Bush presidential administration and the Iraq War. Rows represent cardinality, or number of topic words shown (five, ten, twenty). Columns represent visualization techniques. For **word list** and **word list with bars**, topic words are ordered by their probability for the topic. **Word list with bars** also includes horizontal bars to represent topic-term probabilities.

the words based on their probability for the topic, which blurs the boundary with word clouds; however, this approach is not common. We used a horizontal list of equally sized words ordered by the probability $P(w|t)$ for the word w in the topic t. For space efficiency, we organized our word list in two columns and added item numbers to make the ordering explicit.

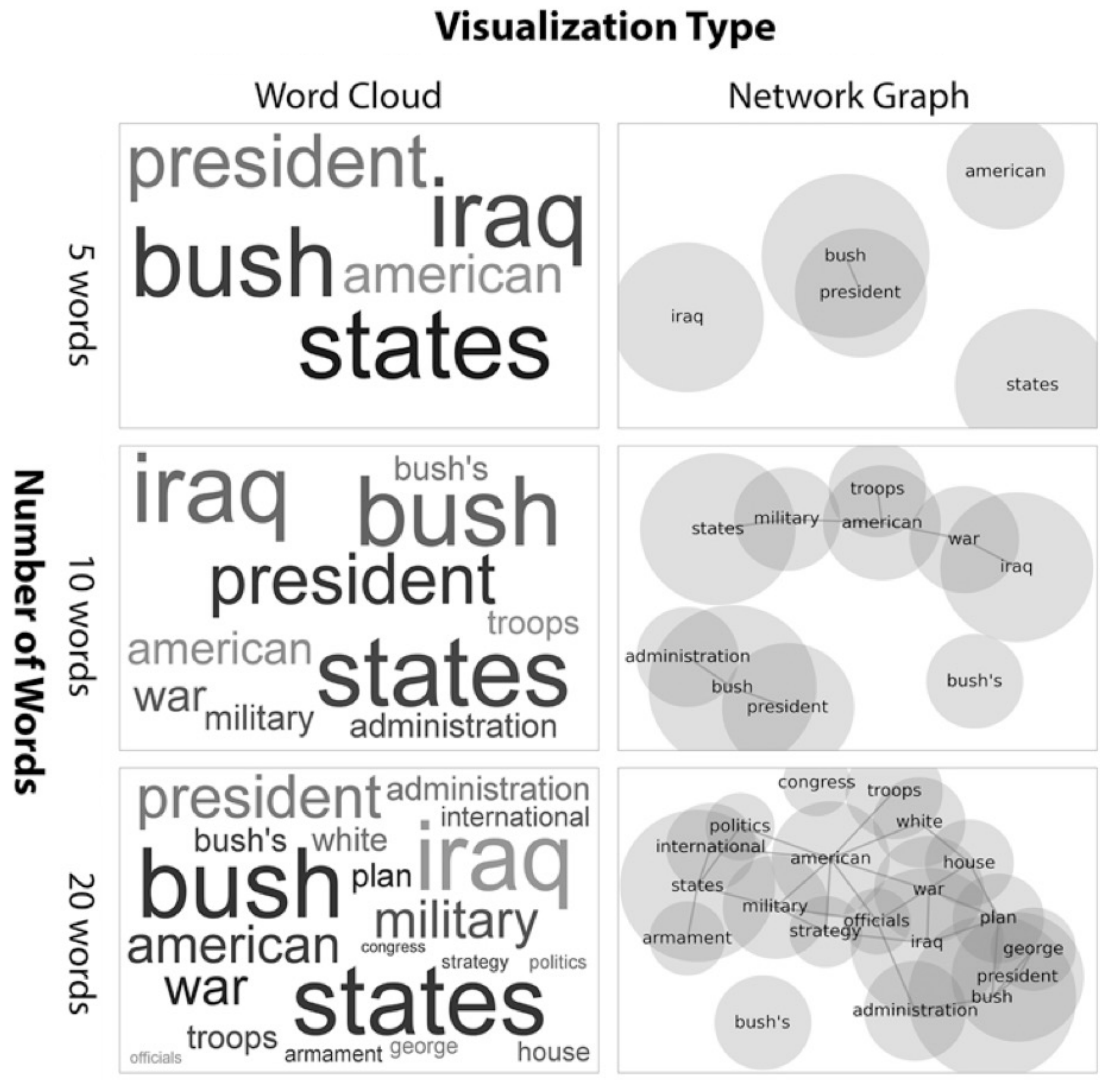


Figure 5.2: Examples of the six of the twelve experimental conditions, each a different visualization of the same topic about the George W. Bush presidential administration and the Iraq War. Rows represent cardinality, or number of topic words shown (five, ten, twenty). Columns represent visualization techniques. In the **word cloud**, words are randomly placed but are sized according to topic-term probabilities. The **network graph** uses a force-directed layout algorithm to co-locate words that frequently appear together in the corpus.

Word list with bars

Combining bar graphs with word lists yields a visual representation that not only conveys the ordering but also the absolute value of the weights associated with the words. We used a similar implementation as that of our prior work (Malik et al., 2013) to add horizontal bars to the word

list for a topic t where the length of each bar represents the probability $P(w|t)$ for each word w.

Word cloud

The word cloud (or *tag cloud*) is one of the most popular and well-known text visualization techniques and is a common visualization for topics. Many options exist for word cloud layout, color scheme, and font size (Mueller, 2012). Prior work on word cloud layouts is split between those that size words by their frequency or probability for the topic (Ramage et al., 2010) and those that size by the rank order of the word (Barth et al., 2014). We used a combination of these techniques where the word's font size is initially set proportional to its probability in a topic $P(w|t)$. However, when the word is too large to fit in the canvas, the size is gradually decreased (Barth et al., 2014). We used a gray scale to visually distinguish words and display all words horizontally to improve readability.

5.2 A Novel Topic Representation: Topic-in-a-Box

While word lists, word lists with bars, and word clouds can be found in many common topic model visualizations (Chaney and Blei, 2012; Eisenstein et al., 2012; Gardner et al., 2010), network graph-based topic visualizations are not as commonplace. However, network graphs support encoding additional relationship information between nodes, which may be useful for enhancing topic understanding. To this end, we implemented a *relationship-enriched* topic visualization, topic-in-a-box (TIB), to help users explore topic models through word and topic correlations.

TIB uses the group-in-a-box (GIB) layout (Rodrigues et al., 2011), which is a network graph-based visualization that represents clusters with emphasis on the edges within and between clusters. TIB uses this layout to visually separate topics of the model as groups.

In TIB, each topic is represented as a force-directed network graph (Fruchterman and Reingold, 1991) where the nodes of the graph are the top words of the topic. We draw edges between two words if they are commonly found next to each other in the corpus; specifically, an edge exists between two topic words, w_1 and w_2, based on bigram co-ocurrence, specifically if $\log(count(w_1, w_2)) > k$, with $k = 0.1$.[2] Similarly, edges are drawn between related topics. Topic relatedness is measured using a topic covariance metric, which measures topic overlap in the document set. Finally, the GIB layout optimizes the visualization such that related topic clusters are placed together spatially. The result is a topic-in-a-box visualization where related words are clustered within the topics and related topics are clustered within the overall layout, as shown in Figure 5.3.

We studied TIB's underlying network graph representation for individual topics alongside word lists, word lists with bars, and word clouds in the user study described in this chapter. Edge width and color were applied uniformly to further reduce complexity in the graph.

5.3 Method

We conducted a controlled study to compare four topic representations: word list, word list with bars, word cloud, and network graph (from TIB). We also compared effectiveness with the number of topic words shown, that is, the *cardinality* of the visualization: five, ten or twenty topic words. To produce a meaningful comparison, the space given to each visualization was held constant: 400 × 250 pixels.

[2] From $k \in \{0.01, 0.05, 0.1, 0.5\}$, we chose $k = 0.1$ as the best trade-off between complexity and provided information.

Figure 5.3: The TIB visualization uses a GIB-inspired layout to represent the topic model as a nested network graph.

5.3.1 Data and Automatic Labels

We selected a corpus that did not assume domain expertise: 7156 *New York Times* articles from January 2007 (Sandhaus, 2008). Ad described in Chapter 3.2.1, we modeled the corpus using an LDA (Blei et al., 2003) implementation in Mallet (Yao et al., 2009) with domain-specific stop-

words and standard hyperparameter settings.[3] Our simple setup was by design: our goal was to emulate the "off the shelf" behavior of conventional topic modeling tools used by novice users. Instead of improving the quality of the model using asymmetric priors (Wallach et al., 2009a) or bigrams (Boyd-Graber et al., 2014), our topic model has topics of variable quality, allowing us to explore the relationship between topic quality and our task measures.

Automatic labels were generated from representative Wikipedia article titles using a technique similar to Lau et al. (2011), following the approach described in Chapter 3.2.3. We first indexed Wikipedia using Apache Lucene.[4] To label a topic, we queried Wikipedia with the top twenty topic words to retrieve fifty articles. These articles' titles comprised our candidate set of labels. We then represented each article using its TF-IDF vector and calculated the centroid (average TF-IDF) of the retrieved articles. To rank and choose the most representative of the set, we calculated the cosine similarity between the centroid TF-IDF vector and the TF-IDF vector of each of the articles. We chose the title of the article with the maximum cosine similarity to the centroid. Unlike Lau et al. (2011), we did not include the topic words or Wikipedia title n-grams derived from our label set, as these labels are typically not the best candidates. Although other automatic labeling techniques exist, we chose this one as it is representative of general techniques.

5.3.2 Task and Procedure

The study included two phases with different users. In **Labeling** (Phase I), users described a topic given a specific visualization, and we measured speed and self-reported confidence in completing the task. In **Validation** (Phase II), users selected the best and worst among a set of Phase I descriptions and an automatically generated description for how well they represented the original

[3] n=50, α=0.1, β=0.01

[4] `http://lucene.apache.org/`

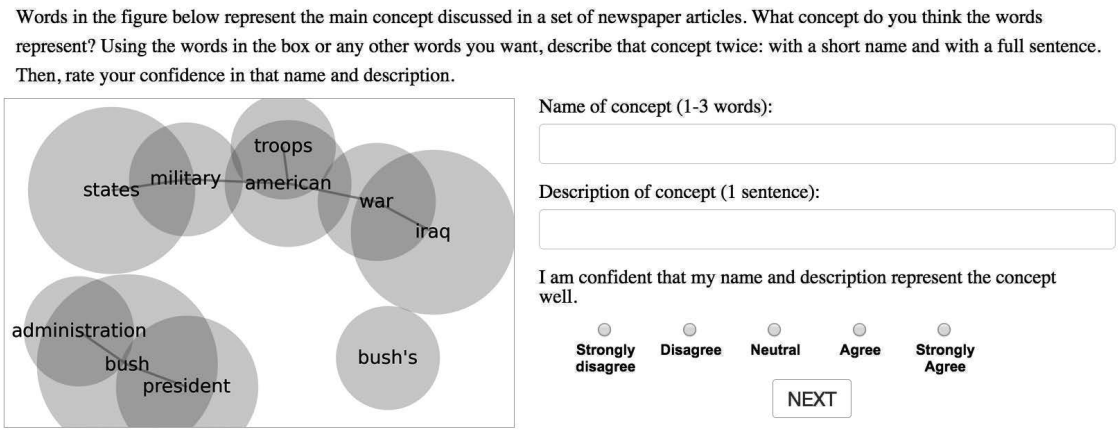


Figure 5.4: The labeling task for the network graph and ten words. Users created a short label and full sentence describing the topic and rated their confidence that the label and sentence represent the topic well.

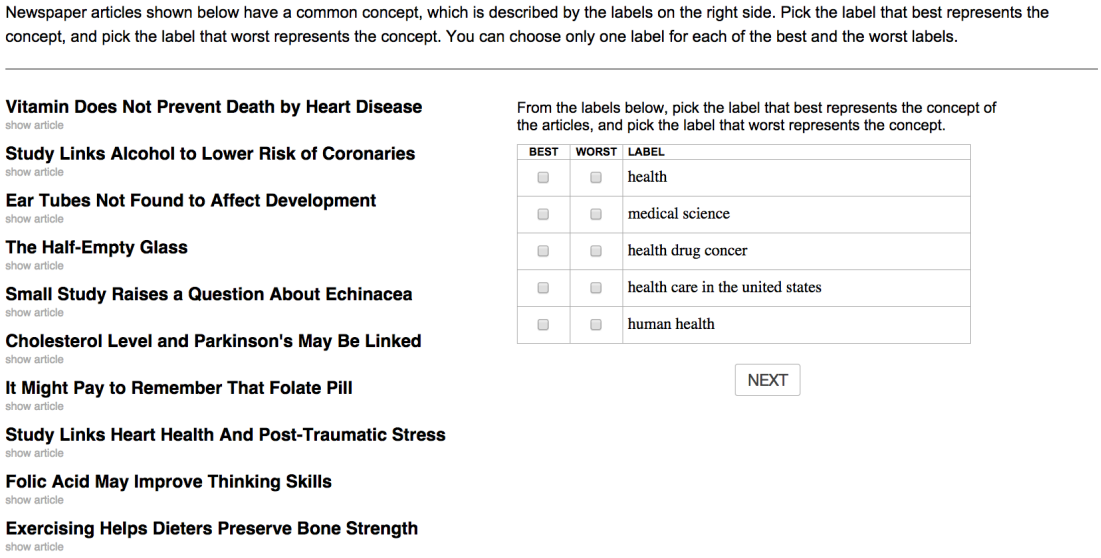


Figure 5.5: During the validation task, users saw the titles of the top ten documents and five potential labels for a topic. Users were asked to pick the best and worst labels. Four labels were created by Phase I users after viewing different visualizations of the topic, while the fifth was generated by the algorithm. The labels were shown in random order.

topics' documents.

Phase I: Labeling

For each labeling task, users saw a topic visualization, provided a short *label* (up to three words), then gave a longer *sentence* to describe the topic, and finally rated their confidence that the label and sentence represented the topic well on a five-point Likert scale. We also tracked the time to perform the task. Figure 5.4 shows an example of a labeling task using the network graph visualization technique with ten words.

Labeling tasks were randomly grouped into human intelligence tasks (HIT) on Mechanical Turk[5] such that each HIT included five tasks from the same visualization technique.[6]

Phase II: Validation

In the validation phase, a new set of users assessed the quality of the labels and sentences created in Phase I by evaluating them against documents associated with the given topic. It is important to evaluate the topic labels in *context*; a label that superficially looks good is useless if it is not representative of the underlying documents in the corpus. Algorithmically generated labels (not sentences) were also included. Figure 5.5 shows an example of a validation task.

The user-generated labels and sentences were evaluated separately. For each task, the user saw the titles of the top ten documents associated with a topic and a randomized set of labels or sentences, one elicited from each of the four visualization techniques within a given cardinality. The set of labels also included an algorithmically generated label. We asked the user to select the "best" and "worst" of the labels or sentences based on how well they described the documents. Documents were associated to topics based on the probability of the topic, t, given the document, d, $P(t|d)$.

[5] All users were in the US or Canada, had more than fifty previously approved HITS, and had an approval rating greater than 90%.

[6] We did not restrict users from performing multiple HITS, which may have exposed them to multiple visualization techniques. Users completed on average 1.5 HITS.

Only the title of each document was initially shown to the user with an option to "show article" (or view the first 400 characters of the document).

All labels were lowercased to enforce uniformity. We merged identical labels so users did not see duplicates. If a merged label received a "best" or "worst" vote, the vote was split equally across all of the original instances (i.e., across multiple visualization techniques with that label). Finally, we tracked task completion time.

Each user completed four randomly selected validation tasks as part of a HIT, with the constraint that each task must be from a different topic. We also used ground truth seeding for quality control: each HIT included one additional test task that had a purposefully bad label generated by concatenating three random dictionary words. If the user did not pick the bad label as the "worst," we discarded all data in that HIT.

5.3.3 Study Design and Data Collection

For Phase I, we used a factorial design with factors of *Visualization* (levels: word list, word list with bars, word cloud, and network graph) and *Cardinality* (levels: 5, 10, and 20), yielding twelve conditions. For each of the fifty topics in the model and each of the twelve conditions, at least five users performed the labeling task. These users each described the topic with a label and sentence, resulting in a minimum of 3000 label and sentence pairs. Each HIT included five of these labeling tasks, for a minimum of 600 HITS. The users were paid $0.30 per HIT.

For Phase II, we compared descriptions across the four visualization techniques (and automatically generated labels), but only *within* a given cardinality level rather than *across* cardinalities. We collected 3212 label and sentence pairs from 589 users during Phase I. For validation in Phase II, we used the first five labels and sentences collected for each condition for a total of $3,000$ labels

and sentences. These were shown in sets of four (labels or sentences) during Phase II, yielding a total of 1500 (3000/4 + 3000/4) tasks. Each HIT contained four validation tasks and one ground truth seeding task, for a total of 375 HITs. To increase robustness, we validated twice for a total of 750 HITs, without allowing any two labels or sentences to be compared twice. The users were paid $0.50 per HIT.

5.4 Findings

We report on labeling time and self-reported confidence for the labeling task (Phase I) before reporting on the label quality assessments (Phase II). We then report on linguistic qualities of the labels, which should motivate future work in automatic label generation.

We first provide an example of user-generated labels and sentences: the user labels for the topic shown in Figures 5.1 and 5.2 include government, iraq war, politics, bush administration, and war on terror. Examples of sentences include "President Bush's military plan in Iraq" and "World news involving the US president and Iraq".[7]

To interpret the results, it is useful to also understand the quality of the generated topics, which varied throughout the model and may impact a user's ability to generate good labels. We measured topic quality using *topic coherence*. As introduced in Chapter 3.2.2, topic coherence is an automatic measure that correlates with how much sense a topic makes to a user (Lau et al., 2014).[8] The average topic coherence for the model was 0.09 (σ = 0.05). Figure 5.6 shows the three best (top) and three worst topics (bottom) according to their observed coherence: the coherence metric distinguishes obvious topics from inscrutable ones.

[7]The complete set of labels and sentences are available at `https://github.com/alisonmsmith/Papers/tree/master/TopicRepresentations`.

[8]For this study, we used a reference corpus of 23 million Wikipedia articles for computing NPMI-based topic coherence.

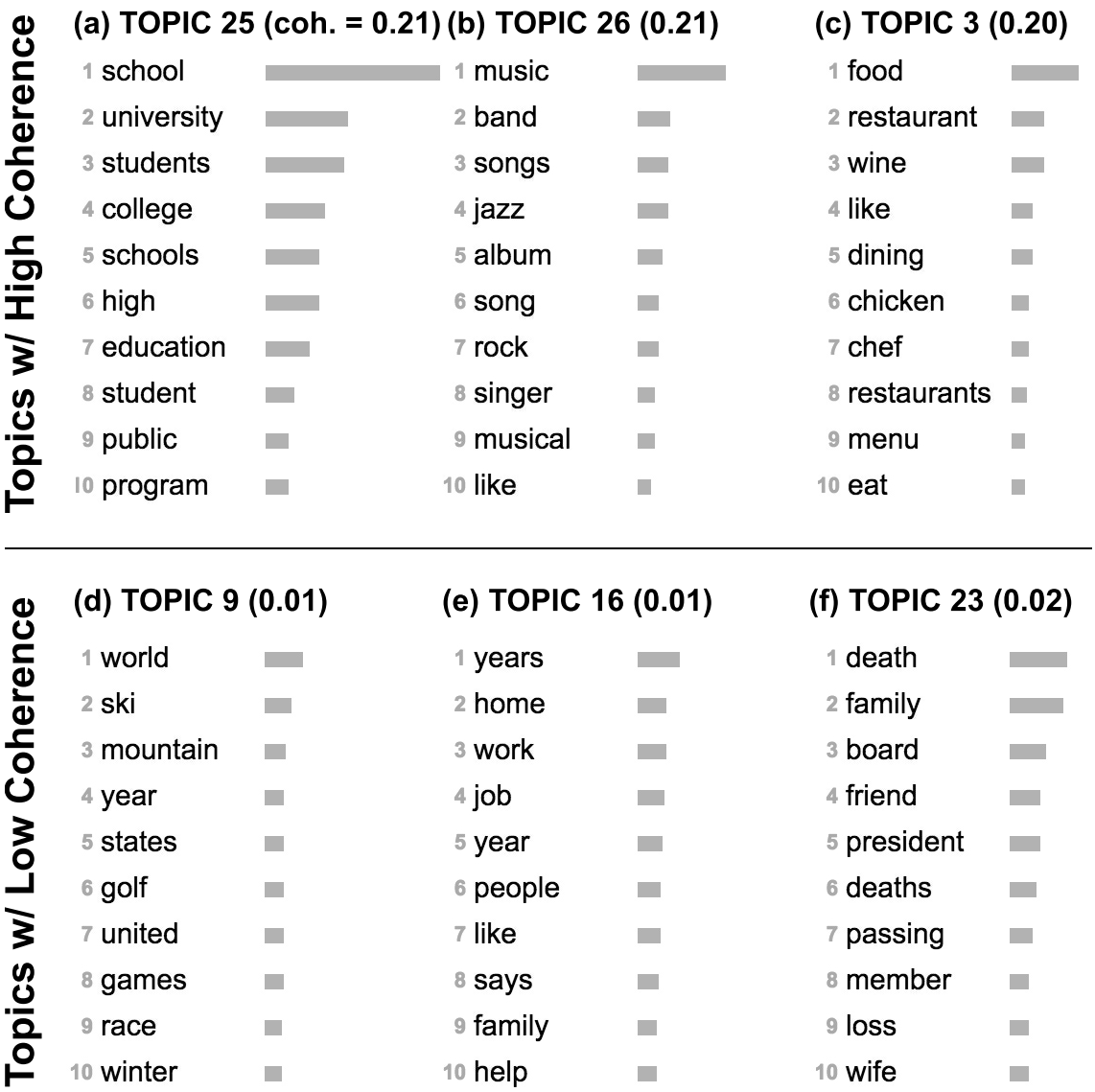


Figure 5.6: Word list with bar visualizations of the three best (top) and worst (bottom) topics according to their coherence score, which is shown to the right of the topic number. The average topic coherence was 0.09 (σ = 0.05).

Labeling Time

More complex visualization techniques took longer to label (Table 5.1 and Figure 5.7). The labeling tasks took on average 57.9 seconds (σ = 58.5) to complete and a two-way ANOVA (visualization technique × cardinality) reveals significant main effects for both the visualization technique[9] and

[9] $F_{(3,3199)} = 10.58$, $p < .001$, $\eta_p^2 = .01$

Table 5.1: Overview of the labeling phase: number of tasks completed, the average and standard deviation (in parentheses) for time spent per task in seconds, and the average and standard deviation for self-reported confidence on a 5-point Likert scale for each of the twelve conditions.

Viz	**Word List**			**Word List w/ Bars**			**Word Cloud**			**Network Graph**		
Card	5	10	20	5	10	20	5	10	20	5	10	20
# tasks	264	268	268	264	280	260	268	268	268	267	274	263
time	53.0	53.2	52.1	58.4	58.7	60.7	52.7	49.4	68.4	55.0	55.6	77.9
(σ)	(44.3)	(46.6)	(53.3)	(75.1)	(51.1)	(57.9)	(47.4)	(37.4)	(85.4)	(50.7)	(56.0)	(71.9)
conf	3.7	3.7	3.6	3.6	3.6	3.7	3.5	3.6	3.6	3.4	3.6	3.7
(σ)	(0.9)	(0.9)	(0.9)	(0.9)	(0.8)	(0.8)	(1.0)	(0.9)	(0.9)	(1.1)	(0.8)	(0.8)

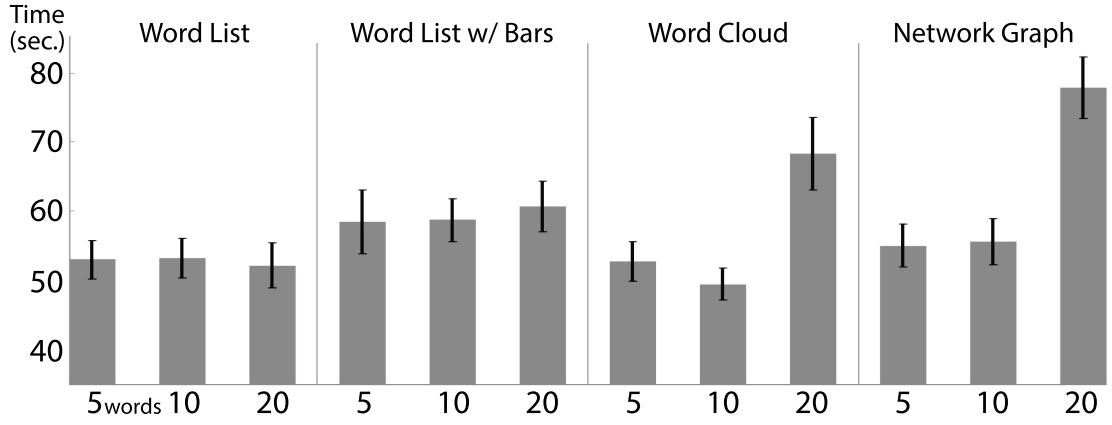


Figure 5.7: Average time for the labeling task, across visualizations and cardinalities, ordered from left to right by visual complexity. For 20 words, network graph was significantly slower and word list was significantly faster than the other visualization techniques. Error bars show standard error.

the cardinality,[10] as well as a significant interaction effect.[11]

For lower cardinality, the labeling time across visualization techniques was similar, but there were notable differences for higher cardinality. Posthoc pairwise comparisons based on the interaction effect (with Bonferroni adjustment) found no significant differences between visualizations with five words and only one significant difference for ten words (word list with bars was slower than word cloud, $p < .05$). For twenty words, however, the network graph was significantly slower at an average of 77.9s ($\sigma = 72.0$) than the other three visualizations ($p < .05$). This effect was likely due

[10] $F_{(2,3199)} = 14.60$, $p < .001$, $\eta_p^2 = .01$
[11] $F_{(6,3199)} = 4.59$, $p < .001$, $\eta_p^2 = .01$

to the network graph becoming increasingly dense with more nodes (Figure 5.2, *bottom right*). In contrast, the relatively simple word list visualization was significantly faster with twenty words than the three other visualizations ($p < .05$), taking only 52.1 seconds on average ($\sigma = 53.4$). Word list with bars and word cloud were not significantly different from each other.

As a secondary analysis, we examined the relationship between elapsed time and the observed coherence for each topic. Topics with high coherence scores, for example, may be faster to label, because they are easier to interpret. However, the small negative correlation between time and coherence (Figure 5.8, *top*) was not significant ($r_{48} = -.13$, $p = .364$).

Self-Reported Labeling Confidence

For each labeling task, users rated their confidence that their labels and sentences described the topic well on a scale from 1 (least confident) to 5 (most confident). The average confidence across all conditions was 3.6 ($\sigma = 0.9$). Kruskal-Wallis tests showed a significant impact of visualization technique on confidence with five and ten words, but not twenty.[12] While average confidence ratings across all conditions only ranged from 3.4 to 3.7, perceived confidence with network graph suffered when the visualization had too few words (Table 5.1).

As a secondary analysis, we compared the self-reported confidence with observed coherence for each topic (Figure 5.8, *bottom*). Increased user confidence with more coherent topics was supported by a moderate positive correlation between topic coherence and confidence ($r_{48} = .32$, $p = .026$). This result provides further evidence that topic coherence is an effective measurement of topic interpretability.

[12]Five words: $\chi^2_3 = 12.62$, $p = .006$. Ten words: $\chi^2_3 = 7.94$, $p = .047$. We used nonparametric tests because the data was ordinal and we could not guarantee that all differences between points on the scale were equal.

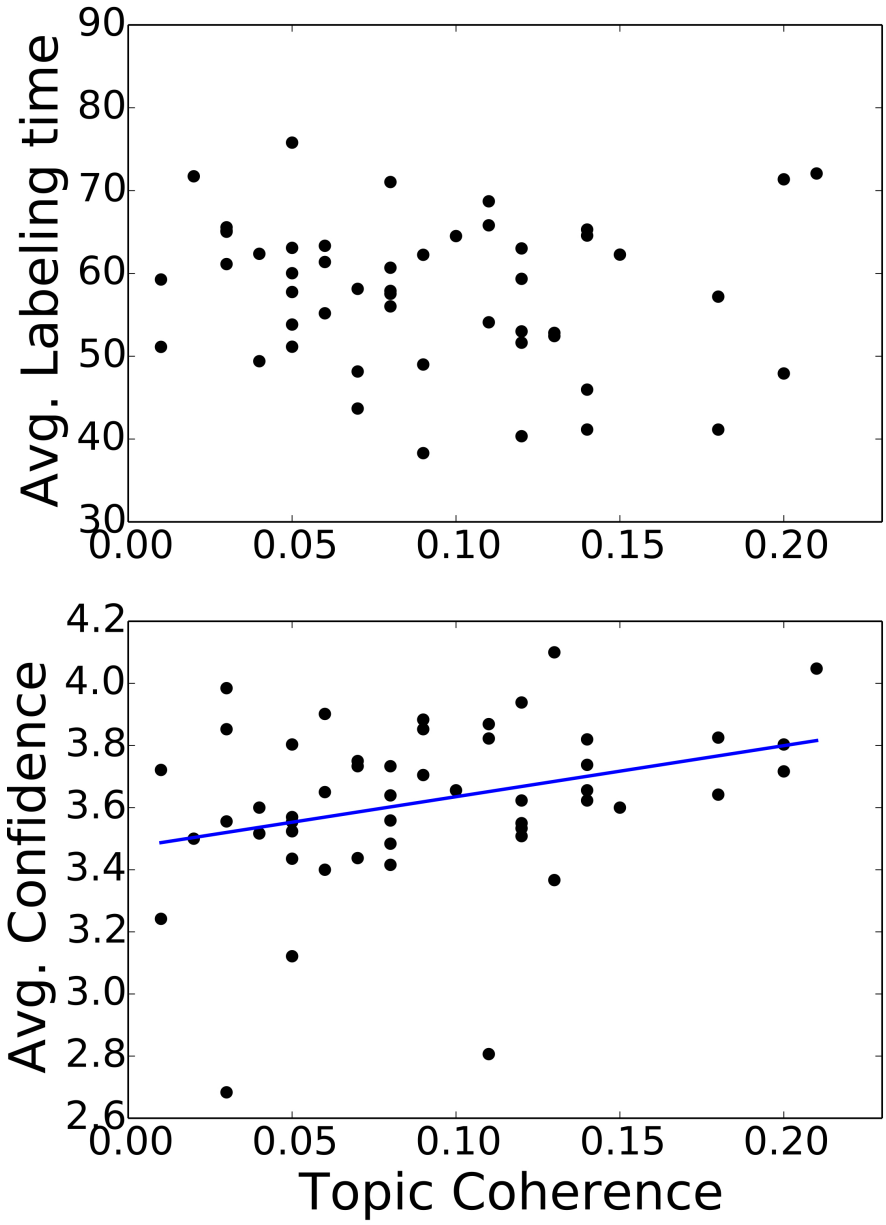
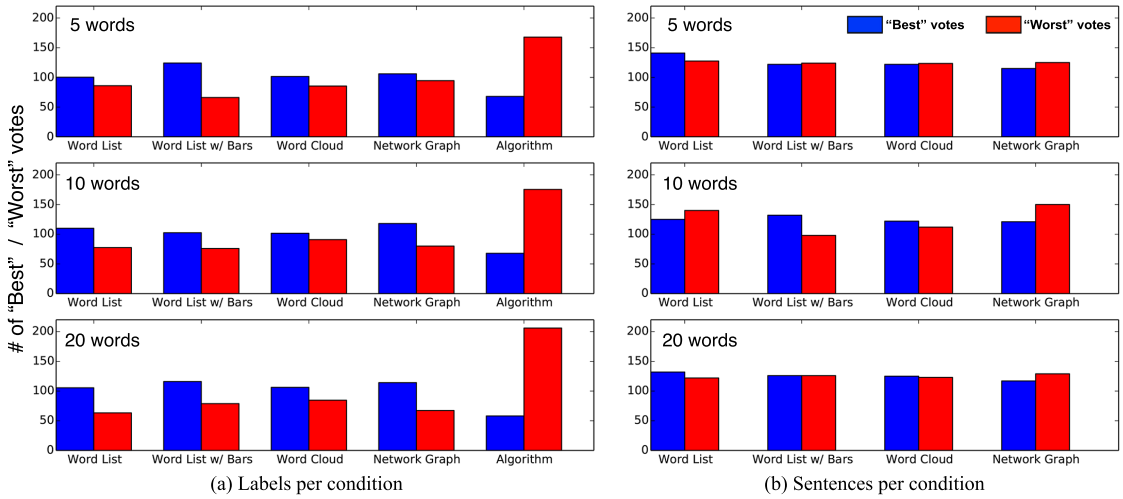


Figure 5.8: Relationship between observed coherence and labeling time (top) and observed coherence and self-reported confidence (bottom) for each topic. The positive correlation (Slope = 1.64 and R^2 = 0.10) for confidence was significant.

Other Users' Rating of Label Quality

Other users' perceived quality of topic labels is the best real-world measure of quality (as described in Chapter 5.3.2). Overall, the visualization techniques had similar quality labels, but



(a) Labels per condition

(b) Sentences per condition

Figure 5.9: The "best" and "worst" votes for labels and sentences for each condition. The automatically generated labels received more "worst" votes and fewer "best" votes compared to the user-created labels.

automatically generated labels did not fare well. Automatic labels got far fewer "best" votes and far more "worst" votes than user-generated labels produced from any of the four visualization techniques (Figure 5.9). Chi-square tests on the distribution of "best" votes for labels for each cardinality showed that the visualization matters.[13] Posthoc analysis using pairwise Chi-square tests with Bonferroni correction showed that automatic labels were significantly worse than user-generated labels from each of the visualization techniques (all comparisons $p < .05$). No other pairwise comparisons were significant.

For sentences, no visualization technique emerged as better than the others. Additionally, there was no existing automatic approach to compare against. The distribution of "best" counts here was relatively uniform. Separate Kruskal-Wallis tests for each cardinality to examine the impact of the visualization techniques on "best" counts did not reveal any significant results.

As a secondary qualitative analysis, we examined the relationship between topic coherence and the assessed quality of the labels. The automatic algorithm tended to produce better labels for the

[13] Five words: $\chi^2_{4,N=500} = 16.47$, $p = .002$. Ten words: $\chi^2_{4,N=500} = 14.62$, $p = .006$. Twenty words: $\chi^2_{4,N=500} = 22.83$, $p < .001$.

coherent topics than for the incoherent topics. For example, Topic 26 (Figure 5.6, b)—{music, band, songs}—and Topic 31 (Figure 5.6, c)—{food, restaurant, wine}—were two of the most coherent topics. The automatic algorithm labeled Topic 26 as music and Topic 31 as food. For both of these coherent topics, the labels generated by the automatic algorithm secured the most "best" votes and no "worst" votes. In contrast, Topic 16 (Figure 5.6, e)—{years, home, work}—and Topic 23 (Figure 5.6, f)—{death, family, board}—were two of the least coherent topics. The automatic labels refusal of work and death of michael jackson yielded the most "worst" votes and fewest "best" votes.

To further demonstrate this relationship, we extracted from the 50 topics the top and bottom *quartiles* of 13 topics each[14] based on their observed coherence scores. Figure 5.10 shows a comparison of the "best" and "worst" votes for the topic labels for these quartiles, including user-generated and automatically generated labels. For the top quartile, the number of "best" votes per technique ranged from 61 for automatic labels to 96 for the network graph visualization. The range for the bottom quartile was larger, from only 45 "best" votes for automatic labels to 99 for word list with bars. The automatic labels, in particular, received a large relative increase in "best" votes when comparing the bottom quartile to the top quartile (increase of 37%).

Additionally, the word list, word cloud, and network graph visualizations all led to labels with similar "best" and "worst" votes for both the top and bottom quartiles. However, the word list with bars representation showed both a large relative increase for the best votes (increase of 19%) and relative decrease for the "worst" votes (decrease of 23%) when comparing the top to the bottom quartile. These results suggest that adding numeric word probability information highlighted by the bars may help users understand poor quality topics.

[14]We could not get exact quartiles, because we have 50 topics, so we rounded up to include 13 topics in each quartile.

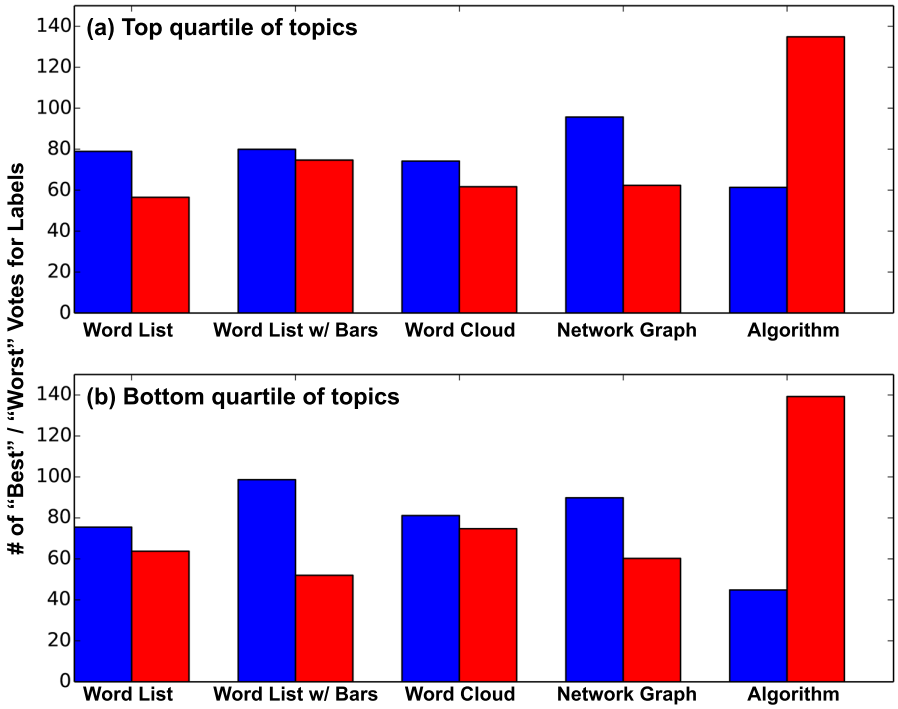


Figure 5.10: Comparison of the "best" and "worst" votes for labels generated using the different visualization techniques (and the automatically generated labels) for the top quartile of topics (top) and bottom quartile of topics (bottom) by topic coherence. The automatically generated labels receive far more "best" votes for the coherent topics.

Label Analysis

The results of Phase I provided a large manually generated label set. Exploratory analysis of these labels revealed linguistic features users tended to incorporate when labeling topics. We discuss implications for automatic labeling in Chapter 5.5. In particular, users prefer shorter labels, labels that include topic words and phrases, and abstraction in topic labeling.

Length

The manually generated labels use 2.01 words ($\sigma = 0.95$), and the algorithmically generated labels use 3.16 words ($\sigma = 2.05$). Interestingly, the labels voted as "best" were shorter on average than those voted "worst," regardless of whether algorithmically generated labels are included in the analysis. With algorithmically generated labels included, the average lengths are 2.04 ($\sigma = 1.16$)

words for "best" labels and 2.83 (σ = 1.79) words for "worst" labels,[15] but even without the algorithmically generated labels, the "best" labels are shorter (M = 1.96, σ = .87) than the "worst" labels (M = 2.09, σ = 1.01).

Shared topic words

Of the 3,212 labels, 2,278, or 71%, contained at least one word taken directly from the topic words—that is, the five, ten, or twenty words shown in the visualization; however, there were no notable differences between the visualization techniques. Additionally, the number of topic words included on average was similar across all three cardinalities, suggesting that users often used the same number of topic words regardless of how many were shown in the visualization.

We further examined the relationship between a topic word's rank and whether the word was selected for inclusion in the labels. Figure 5.11 shows the average probability of a topic word being used in a label by the topic word's rank. More highly ranked words were included more frequently in labels. As cardinality increased, the highest ranked words were also less likely to be employed, as users had more words available to them.

Phrases

Although LDA makes a "bag of words" assumption when generating topics, users can reconstruct relevant phrases from the unique words. For Topic 26, for example, all visualizations included the same topic terms. However, the network graph visualization highlighted the phrases "jazz singer" and "rock band" by linking their words as commonly co-occurring terms in the corpus. These phrases were not as easily discernible in the word cloud visualization (Figure 5.12). We computed a set of common phrases by taking all bigrams and trigrams that occurred more than fifty and

[15]The "best" label set includes all labels voted at least once as "best," and similarly the "worst" label set includes all labels voted at least once as "worst."

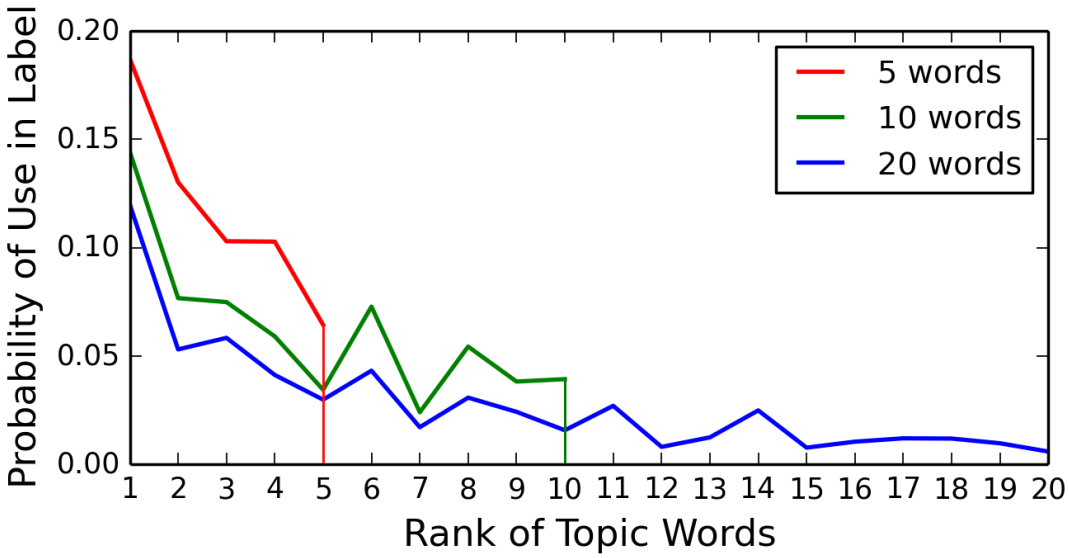


Figure 5.11: Relationship between rank of topic words and the average probability of occurrences in labels. The three lines—red, green, and blue—represent cardinality of five, ten, and twenty, respectively. The higher-ranked words were used more frequently.

twenty times, respectively, in the NYT corpus. Of the 3212 labels, 575 contained one of these common phrases, but those generated by users with the network graph visualization contained the most phrases. Labels generated in the word list (22% of the labels), word list with bars (25%), and word cloud (24%) conditions contained fewer phrases than the labels generated in the network graph condition (29%). Although it is not surprising that the network graph visualization better communicates common phrases in the corpus as edges are drawn between these phrases, this finding suggests other approaches to drawing edges. Edges drawn based on sentence or document-based co-occurrence, for example, could instead uncover longer-distance dependencies between words, potentially identifying distinct sub-topics with a topic.

Hyponymy

Users often preferred more general terms for labels than the words in the topic (Newman et al., 2010b). To measure this, we looked for the set of unique hyponyms and hypernyms of the topic

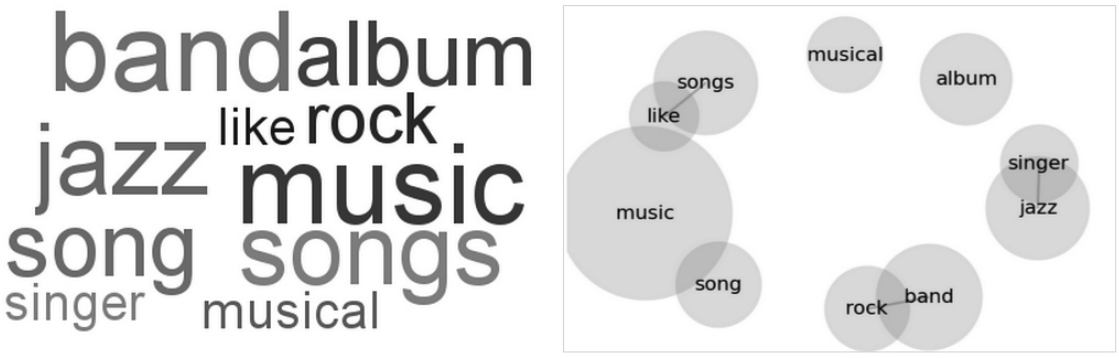


Figure 5.12: Word cloud and network graph visualizations of Topic 26. Phrases such as "jazz singer" and "rock band" were obscured in the word cloud but were shown in the network graph as connected nodes.

words, or those that were not themselves a topic word, that appeared in the manually generated labels. We used the super-subordinate relation, which represents hypernymy and hyponymy, from WordNet (Miller, 1995). Of the 3,212 labels, 235 included a unique hypernym and 152 included a unique hyponym of the associated topic words found using WordNet, confirming that users were significantly more likely to produce a more generic description of the topic ($\chi^2_{1,N=387} = 17.38$, $p < .001$). For the 235 more generic labels, fewer of these came from word list (22%) and more from the network graph (30%) than the other visualization techniques—word list with bars (24%) and word cloud (24%). This may mean that the network graph helps users to better understand the topic words as a group and therefore label them using a hypernym. We also compared hypernym inclusion for "best" and "worst" labels: 63 (5%) of the "best" labels included a hypernym while only 44 (3%) of the "worst" labels included a hypernym. Each of the visualization techniques led to approximately the same percentage of the 152 total more specific labels.

5.5 Discussion

Although the four visualization techniques yielded similar quality labels, our crowdsourced study highlighted the strengths and weaknesses of the techniques. It also revealed some preferred lin-

guistic features of user-generated labels and how these differ from automatically generated labels.

The trade-offs among the visualization techniques show that context matters. If efficiency is paramount, then word lists—both simple and fast—are likely best. For a cardinality of twenty words, for example, users presented with the simple word list were significantly faster at labeling than those shown the network graph visualization. At the same time, more complex visualizations exposed users to multi-word expressions that the simpler visualization techniques may have obscured (Chapter 5.4). Future work should investigate for what types of user tasks this information is most useful. There is also potential for misinterpretation of topic meaning when cardinality is low. Users can misunderstand the topic based on the small set of words, or adjacent words can inadvertently appear to form a meaningful phrase, which may be particularly an issue for the word cloud.

Our crowdsourced study identified the "best" and "worst" labels for the topic's documents. An additional qualitative coding phase could evaluate each "worst" label to determine why, whether due to misinterpretation, spelling or grammatical errors, length, or something else.

Surprisingly, we found no relationship between topic coherence and labeling time (Chapter 5.4). This is perhaps because not only were users quick to label topics they understand, but they also quickly gave up when they had no idea what a topic was about. We did, however, find a relationship between coherence and confidence (Chapter 5.4). This positive correlation supports topic coherence as an effective measure for human interpretability.

Automatically generated labels were consistently chosen as the "worst" labels, although they were competitive with the user-generated labels for highly coherent topics (Chapter 5.4). Future automatic labeling algorithms should still be robust to poor topics. Algorithmically generated labels were longer and more specific than the user-generated labels. It is unsurprising that these au-

tomatic labels were consistently deemed the worst. Users preferred shorter labels with more general words (e.g., hypernyms, Chapter 5.4). We show specific examples of this phenomenon from Topic 14 and Topic 48. For Topic 14—{health, drug, medical, research, conditions}—the algorithm generated the label health care in the united states, but users preferred the less specific labels: health and medical research. Similarly, for Topic 48—{league, team, baseball, players, contract}—the algorithm generated the label major league baseball on fox; users preferred simpler labels, such as baseball. Automatic labeling algorithms thus can be improved to focus on general, shorter labels. Interestingly, simple textual labels have previously been shown to be more efficient but less effective than topic keywords (i.e., word lists) for an automatic document retrieval task (Aletras et al., 2014), highlighting the extra information present in the word lists. Our findings showed that users were also able to effectively interpret the word list information, as that visualization was both efficient and effective for the task of topic labeling compared to the other more complex visualizations.

Although we used WordNet to verify that users preferred more general labels, this is not a panacea, because WordNet does not capture all of the generalization users want in labels. In many cases, users used terms that synthesize relationships beyond trivial WordNet relationships, such as locations or entities. For example, Topic 18—{san, los, angeles, terms, francisco}—was consistently labeled as the location California, and Topic 38—{open, second, final, won, williams}—which almost all users labeled as tennis, required a knowledge of the entities Serena Williams and the U.S. Open.

5.6 Conclusion

In this chapter, we presented a crowdsourced user study to compare four topic visualization techniques—a simple ranked word list, a ranked word list with bars representing word probability, a word cloud, and a network graph—based on how they impact the user's understanding of a topic. The four visualization techniques led to similar quality labels as rated by end users. However, users labelled more quickly with the simple word list, yet tended to incorporate phrases and more generic terminology when using the more complex network graph. Additionally, users felt more confident labeling coherent topics, and manual labels far outperformed the automatically generated labels against which they were evaluated.

Automatic labeling can benefit from this research in two ways: by suggesting when to apply automatic labeling and by providing training data for improving automatic labeling. While automatic labels faltered compared to human labels in general, they did quite well when the underlying topics were of high quality. Thus, one reasonable strategy would be to use automatic labels for a portion of topics, but to use human validation to either first improve the remainder of the topics (Hu et al., 2014) or to provide labels (as in this study) for lower quality topics. Moreover, our labels provide training data that may be useful for automatic labeling techniques using feature-based models (Charniak, 2000)—combining information from Wikipedia, WordNet, syntax, and the underlying topics—to reproduce the types of labels and sentences created (and favored) by users.

This chapter explored different topic representations for end user understanding (interpretability), and found that simple ranked word lists were sufficient for supporting users in quickly understanding topics. Based on these findings, we use a simple ranked word list for the topic overviews in

the interactive topic modeling tool, which we develop and evaluate in Chapters 6 and 7.

Chapter 6: User Experience and Perceptions when Controlling Transparent Systems: a Novel Interactive Topic Modeling System and Interview Study[1]

Chapter 4 exposed important interactions between transparency and control, in particular, the importance of supporting both to mitigate frustration and instill trust. In Chapter 4, control referred simply to whether or how it was provided, but did not explore whether users felt confident providing feedback or their reactions when it was not applied *predictably*, or as expected—a case that is exposed through system transparency. In this chapter, we examine users' experience and perceptions when interacting with transparent unsupervised ML systems—where controls are easier to validate—specifically, interactive topic modeling.

Interactive topic modeling allows users to guide the creation of topic models and to improve model quality without having to be experts in topic modeling algorithms (see Chapter 3.2.4). Prior work developing interactive topic modeling has focused either on algorithmic implementation without understanding how users actually wish to improve models or on user needs but without the context of a fully interactive system. To address this disconnect,we implemented a novel interactive topic

[1]The work in this chapter was published as "Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. Closing the Loop: User-Centered Design and Evaluation of a Human-in-the-Loop Topic Modeling System. In International Conference on Intelligent User Interfaces, 2018 (Smith et al., 2018)."

modeling system based on our previously identified optimal topic representations (from Chapter 5) and with a set of model refinements requested by users in our prior work (Lee et al., 2017). We then conducted a formative study of this system with twelve non-expert participants to examine how end users are affected by issues that arise when interacting with a complex, transparent ML system. We found that although users experience unpredictability, their reactions vary from positive to negative, and, surprisingly, we did not find any cases of distrust, but instead noted instances where users perhaps trusted the system too much or had too little confidence in themselves.

6.1 A "Human-Centered" Interactive Topic Modeling System

Prior work (Lee et al., 2017; Musialek et al., 2016) identified refinements that users expected to be able to use in an interactive topic modeling system. As there was no existing implementation for this broad set of user preferred refinements, in our prior work Lee et al. (2017), we simulated refinements using a Wizard-of-Oz method. To truly evaluate user experience with a fully functional interactive topic modeling system, we implemented seven refinements requested by users: **add word**, **remove word**, **change word order**, **remove document**, **split topic**, **merge topic**, and **add to stop words**.

These refinements included the six top refinements identified, but not implemented, in our prior work (Lee et al., 2017)), except for **merge words**. **Merge words** was suggested by users in our prior study as a means for organizing topic words in the interface rather than a deeper specification that should be implemented in the model. We also included two refinements that were not suggested by users in our prior study, perhaps due to that study's method: **merge topics** did not arise because users only refined individual topics and **add to stop words** may have been overlooked because that study used a generic corpus with a well-curated stop words list.

6.1.1 Refinement Implementation

Our interactive topic modeling implementation is based on LDA with Gibbs sampling for inference (see Chapter 3.2.1). When a user provides feedback to a topic model, we view this as correcting an error the model made.

As discussed in Chapter 3.2.4, we can divide this feedback into two broad classes: *forgetting* bad things the model learned and *injecting* new knowledge into the model. Forgetting is accomplished by "strategic unassignment," or invalidating the topic-word assignments (i.e., setting them to -1) and decrementing any associated counts with those tokens ($n_{w,t}$ and $n_{d,t}$ in Equation 3.4). The result of this process is equivalent to the model seeing that word for the very first time, allowing it to make better decisions. In tandem with forgetting, injecting provides hints that encourage the algorithm to make better decisions going forward.

Recall from Chapter 3.2.1, that the multinomial distributions θ and ϕ are drawn from Dirichlet distributions, where α and β are the Dirichlet priors over θ and ϕ, respectively. Injecting information happens through modifying the Dirichlet parameters for each document, α, and each topic, β.

To implement these refinement operations, we make use of the vector interpretation (rather than scalar) of these priors. Thus, α_d is a K dimensional vector for each document d and β_t is a V dimensional vector for each topic t, where K is the number of topics and V is the size of the vocabulary. Recall from Chapter 3.2.1 that Gibbs sampling iteratively samples a topic assignment, $z = t$ given an observed token w in document d and all other topic assignments, $z\_$, with conditional probability (treating α and β as vectors),

$$P(z = t | z\_, w) \propto (n_{d,t} + \alpha_{d,t}) \frac{n_{w,t} + \beta_{w,t}}{n_t + V\beta_t} \tag{6.1}$$

This conditional probability has two parts: how much a document likes a topic—$(n_{d,t} + \alpha_{d,t})$—and how much a topic likes a word—$(n_{w,t} + \beta_{w,t})$. The priors ($\alpha_{d,t}$ and $\beta_{w,t}$) are added to the topic assignment counts; thanks to the conjugacy of multinomial and Dirichlet distributions, these priors are sometimes called "pseudocounts." Our interactive topic modeling system takes advantage of this by creating pseudocounts to encourage the changes users want to see in the topics. We use initial, default prior values of $\alpha_{d,t} = 1.0/K$, and $\beta_{w,t} = 0.01$ for all refinements in this study.

The refinement operations are:

1. **Add word**: to add the word w to topic t, we forget w from all other topics by forgetting the word's tokens' topic assignments: for each word token w_i, we get its topic assignment t_i. If t not equal to t_i, we decrement the associated topic counts (n_{w,t_i} and n_{d,t_i}) and we assign w_i to an invalid topic (i.e., -1). We then encourage the Gibbs sampler to assign topic t for all of the word's tokens, w_i, by increasing the prior of w in t ($\beta_{w,t}$) by the difference between the topic-word counts of w and the topic-word counts of the topic's top word w' in topic t (i.e., $n_{w',t} - n_{w,t}$). The updated prior for token t is $\beta_{w,t} + n_{w',t} - n_{w,t}$. This large prior makes it more likely that the Gibbs sampler will choose topic t for w.
2. **Remove word**: to remove the word w from topic t, we first forget all the word's tokens w_i from t (like in **Add word**). We then discourage the sampler from reassigning w to t by assigning a very small prior,[2] ϵ, to w in t. The updated prior for w is $\beta_{w,t} = \epsilon$. This small prior makes it less likely that the Gibbs sampler will choose topic t for w.
3. **Change word order**: to reorder word w_2 to appear before word w_1 in topic t, we need to ensure that w_2 is ranked higher than w_1 in topic t. To enforce this, we increase the prior of w_2 in t by the difference between the topic-word counts. Specifically, $\beta_{w_2,t} = \beta_{w_2,t} + n_{w_1,t} - n_{w_2,t}$. For example, if the count of w_1 in t is 10 and the count of w_2 in t is 6, then we increase the

[2] We use $\epsilon = 0.000001$ for our experiments.

prior of w_2, $\beta_{w_2,t}$ from .01 to 4.01. This large prior aims to push w_2 ahead of w_1 in topic t. Intuitively, this operation resembles providing supplemental counts to w_2, so that it ranks higher than w_1 in the topic.

4. **Remove document**: in LDA, each document can be represented as a probability distribution over topics (θ). In the Gibbs sampler, a document's affinity to a particular topic is governed by the term $n_{d,t} + \alpha_{d,t}$. To remove the document d from topic t, we forget the topic assignment for all words in the document d by assigning all w in d to an invalid topic (i.e., -1) and decrementing the associated counts ($n_{w,t}$ and $n_{d,t}$); $n_{d,t}$ is thus 0. We then discourage the sampler from reassigning t for the document by assigning a very small prior,[2] ϵ, to the topic t for d. Specifically, $\alpha_{d,t} = \epsilon$.
5. **Merge topic**: merging topics t_1 and t_2 intends for the model to have a combined topic that represents both t_1 and t_2. We assign t_1 to all tokens that were previously assigned to t_2, simultaneously decrementing the associated counts for t_1 and incrementing those for t_2. This effectively deletes t_2 from the model and decrements the number of topics.
6. **Split topic**: to split topic t, the user provides a subset of the topic's words, or seed words, which need to be moved from the original topic, t, to a new topic, t_n. To implement this, assign all seed words to an invalid topic (and decrement the associated counts), create a new topic by incrementing the number of topics, and assign a large prior[3] for each of the seed words, w_s, in the new topic t_n (β_{w_s,t_n}). The Gibbs sampler's job is to sort which words land in which of the new child topics.
7. **Add to stop words**: adding the word w to global stop words removes w from *all* topics. We exclude that w from the vocabulary V. This ensures that the Gibbs sampler will ignore all occurrences of w in the corpus.

[3]Following Fan et al. (2017), we use 100 as the large prior.

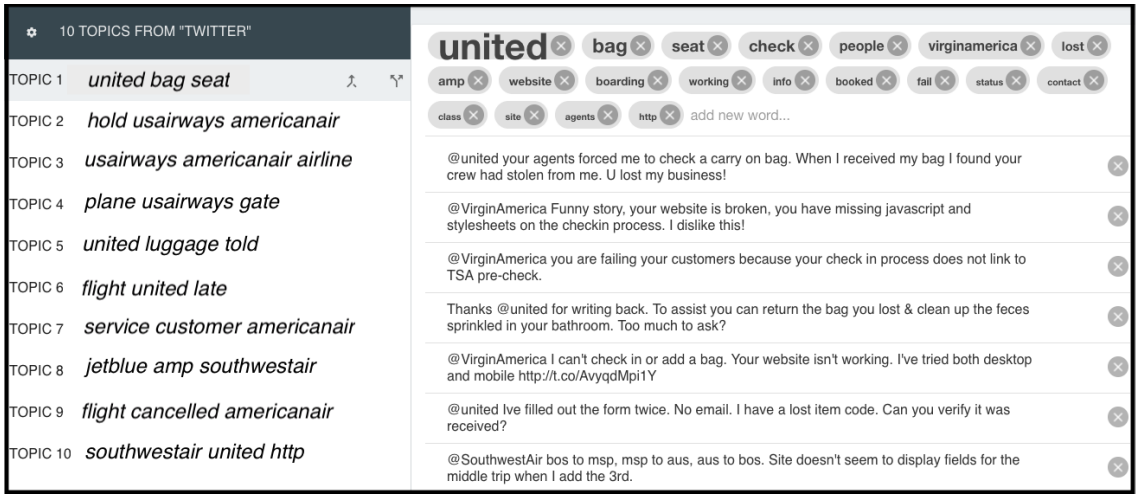


Figure 6.1: User interface for the interactive topic modeling system. A list of topics (left) are represented by topics' first three topic words. Selecting a topic reveals more detail (right): the top 20 words and top 40 documents. Hovering or clicking on a word highlights it within the documents. Users can refine the model using simple mechanisms: click "x" next to words or documents to remove them, select and drag words to re-order them, type new words from the vocabulary into the input box and press "enter" to add them, select a word and click the trash can to add it to the stop words list, or click "split" and "merge" (to the right of the topic words) to enter into split and merge modes.

We contrast this refinement implementation to that of Hu et al. (2014): their proposed framework models user feedback as word correlation knowledge, which does not clearly extend to all desired types of feedback. For example, it is not obvious how to implement **remove document** using word correlations. Alternatively, our proposed asymmetric prior-based framework allows us to incorporate diverse feedback types in the form of simple prior manipulation operations.

6.1.2 Interactive Topic Modeling System Interface

The interactive topic modeling user interface (Figure 6.1) represents a topic model as a list of topics on the left panel, each displayed as their first three words. Selecting any topic in the list shows the full topic view in the right panel, which consists of the top 20 topic words and snippets of the top 40 topic documents. Documents are ordered by their probability for the topic t given the document d, or $P(t|d)$. Each word, w, is ordered and sized by its probability for the topic t, or

$P(w|t)$; this simple word list representation provides users a quick topic understanding (Alexander and Gleicher, 2016; Smith et al., 2017). Hovering or clicking on topic words highlights the word in the displayed document snippets.

Users refine the topic model using simple interactive mechanisms. In this system, we require users to click "save" to incorporate their specified refinements instead of applying them immediately because the system does not support reverting the model after an update (we discuss batch vs. immediate refinements in Chapter 6.4). Instead, the interface displays intermediate feedback, such as bold and italicized words, representing users' specified refinements before saving, and any or all of the outstanding refinements can be undone. When users press "save," their specifications are incorporated into the model (Chapter 6.1.1).

6.2 Method

Our fully interactive user-centered interactive topic modeling system focuses on topic model novices. Participants explored and refined a model built from a Twitter corpus of complaints about airlines, followed by a semi-structured interview. The study focused on a broad set of operations in a fully interactive system (compared to our prior work (Lee et al., 2017)), as well as understanding how interactive machine learning challenges–predictability, complexity, and latency–complicate topic modeling. For refinements in our system, we explore predictability in terms of *control adherence* and *stability*, where *control adherence* is how much the user's refinement is reflected after the model updates (e.g., a specified word is added to the topic), and *stability* is how many other changes not specified by the user appear in the model (e.g., other unspecified words are added). Instability, in particular, is a concern with interactive topic modeling: small changes to the model can propagate in unexpected ways.

The study protocol included a training task to familiarize participants with topic modeling, a test task to refine a topic model, and a semi-structured interview on the experience.

6.2.1 Participants

We recruited twelve participants (five male, seven female) from campus e-mail lists. They were on average 30.5 years old (σ = 10.3) and fluent English speakers. Educational backgrounds included human-computer Interaction (5), information management (2), education (1), mechanical engineering (1), computer science (1), psychology (1), and international government (1). Experience with topic modeling varied (nine with no experience, three with limited) as did experience with data science or machine learning (seven with no experience, three limited, two significant). Each participant got a $15 Amazon gift card. We refer to participants as P1–P12.

6.2.2 Dataset and Topic Model

We used a separate dataset and model for the training and test tasks. For *training* we generated a model with 10 topics from a dataset of 2,225 BBC news articles corresponding to stories in five topical areas (business, entertainment, politics, sports, tech) from 2004 – 2005 (Greene and Cunningham, 2006). For the *test* we used the Twitter US Airline Sentiment dataset from Kaggle,[4] which includes 14,485 total tweets from February 2015 directed to six popular airlines (American, Delta, Southwest, United, US Airways, Virgin America). The dataset includes manually applied labels organizing the tweets into "positive" (2,363 tweets), "neutral" (3,099 tweets), and "negative" (9,178 tweets) sentiment categories. We modeled the 9,178 negative sentiment tweets with 10 topics using a standard stop words list[5] and 300 Gibbs sampling iterations. For each subsequent

[4]https://www.kaggle.com/crowdflower/twitter-airline-sentiment
[5]https://raw.githubusercontent.com/mimno/Mallet/master/stoplists/en.txt

Table 6.1: Initial topic model of 10 topics generated for the negative tweets from the airline Twitter corpus. Topics are represented by their top words. Observed topic coherence calculated by NPMI, which deems topics to be of higher quality if they contain words that appear more frequently together than apart in a reference corpus.

ID	NPMI	Topic Words
T1	.031	hold, usairways, americanair, call, back, phone, hours, wait, change, minutes
T2	.014	southwestair, virginamerica, ticket, united, amp, fly, website, boarding, time, guys
T3	.024	flight, usairways, delayed, hrs, hours, late, miss, made, delay, connection
T4	.045	united, bag, bags, luggage, lost, baggage, check, find, airport, time
T5	.015	jetblue, http, time, united, email, long, jfk, give, amp, guys
T6	.029	americanair, usairways, people, weather, due, day, airport, hotel, issue, issues
T7	.022	united, plane, gate, waiting, hour, seat, sitting, crew, delay, min
T8	.009	usairways, americanair, make, problems, days, travel, refund, miles, told, booking
T9	.030	service, customer, united, usairways, worst, airline, experience, agents, staff, flying
T10	.025	flight, cancelled, southwestair, flightled, americanair, flights, today, flighted, late, tomorrow

update during the task, 30 Gibbs sampling iterations were run. Table 6.1 shows the initial set of topics (henceforth T1–T10). We automatically computed topic quality for each topic using a topic coherence metric based on Normalized Pointwise Mutual Information (Bouma, 2009, NPMI) with Wikipedia as the reference corpus (Lau et al., 2014).

6.2.3 Procedure

Sessions were designed to take one hour, but in practice took up to 90 minutes, and they were conducted remotely with audio and screen-capture recording. We introduced participants to topic modeling and to the interactive topic modeling system using the *training topic model*. The inter-

viewer described each refinement operation and asked the participant to practice sample operations.

Participants then reviewed the raw tweets of the *test* dataset in a csv file and were told to imagine they had been asked to organize these tweets to identify different classes of airline complaints. They then opened the system with the *test topic model* (Figure 6.1) and were instructed that an initial model of 10 topics had been generated to help summarize the tweets, but that they may notice flaws and may need to refine the model. The interviewer asked a few introductory questions about the model and the system, then instructed participants to think aloud while refining the model using the system until they felt it best categorized the tweets into types of complaints. Participants were given a maximum of 20 minutes for the task, and afterwards they answered semi-structured interview questions about the task, model, and system.

6.2.4 Data and Analysis

We logged user interaction with the system, including the state of the model at each iteration, when the user pressed "save," and refinement usage. The task audio was also transcribed and coded along with the responses for the post-task interview. Coding followed a thematic analysis approach (Braun and Clarke, 2006) to uncover the overarching themes represented by more specific codes within the data. The codebook was organized into five themes containing a total of 40 codes: challenges (10 codes), system requests (10), refinement requests (8), save strategies (6), and refinement strategies (6). To determine agreement, two researchers independently coded transcripts for a random participant. Of 21 instances, the researchers agreed on the codes for 12 and disagreed on nine. Disagreements were resolved and codes clarified through discussion, and a second round of coding on transcripts for a different random participant achieved better agreement

(researchers agreed on codes for 14 of 15 instances). One researcher then coded the remaining transcripts.

6.3 Findings

We discuss findings related to refinement and save strategies, ability to improve the topic model, and challenges faced in using a fully functional interactive topic modeling system.

Participants preferred simple refinements

Like Lee et al. (2017), simple refinements, such as **remove word**, **change word order**, and **add word to stop words** were the most commonly used. While perceived utility aligned with usage in Lee et al. (2017), which is not surprising as refinements did not affect the model, there were two misaligned cases in our study: **change word order** and **add word** (Table 6.2). **Change word order** was the second most common refinement, yet only two of the 10 participants who used it in the task thought it was one of the most useful; alternatively, **add word** was only the fourth most common refinement, yet all six participants who used it thought it one of the most useful. These refinements provide varied control; we discuss this discrepancy in Chapter 6.4.

Detailed refinements usage and strategies

We recorded which refinements participants used. The most common refinement, **remove word**, was used by 11 participants a total of 270 times, followed by **change word order** (10 participants, 136 times), **add to stop words** (seven participants, 90 times), and **add word** (six participants, 41 times). Other refinement operations were used by only three or fewer participants (Table 6.2).

When we asked participants the strategies they used, we got similar answers: “remove irrelevant

words" (9 participants), "remove typos" (2), "skip bad topics" (2), "group common words" (2), "change word order to name" (2), "move irrelevant words to the end of the list" (1) , and "pinpoint refine" (1). To remove irrelevant words, participants were not consistent, instead employing both **remove word** and **add to stop words**. For example, P6 described that he would, *"first remove all similar words (e.g., make/makes) in each topic and then put all generic words in the stop words list."* Two participants described using **change word order** not only to fix the relative importance of words, but to name a topic, which they did by dragging three descriptive words to the front of the word list (each topic was represented by its top three words in the topic list on the left of the interface). A more expected usage of **change word order** came from P4, who said, *"I reordered the airline names to go to the end as I was not interested in what airlines attracted complaints"*. For dealing with poor quality topics, two participants described their strategy to ignore bad topics, while one participant described a pinpoint refinement strategy in which she would choose a single topic word from a seemingly random topic and then use **add word** and **remove word** to make the topic more about that single word. Finally, we also noted cases of participants using refinements to explore the model. For example, P10 used the **add word** refinement to see if words showed up in the topic's documents, by first adding a word and then hovering over it to see it highlighted in the documents. P10 would then undo the added word if it did not appear in any of the top documents.

When and why did participants choose to save their changes?

Participants refined the topic model by applying refinements and then separately clicking "save." Before saving, participants could undo some or all of their changes. To understand when participants choose to save and because the interactive topic modeling system does not support undo after saving, the system did not enforce a particular save strategy, such as after every refinement or a set number of refinements. Instead, participants could specify a series of local refinements,

Table 6.2: List of refinements ordered by in-task usage with count of participants that selected the specified refinement as one of the most useful or least useful refinements. Simple, word-level refinements were both the most commonly used and judged to be most useful (except for change word order: only two of the 10 participants who used it found it to be most useful).

Refinement	Most Useful	Least Useful	Used By	Total
Remove word	5	1	11	270
Change word order	2	1	10	136
Add to stop words	3	0	7	90
Add word	6	1	6	41
Remove document	0	3	3	20
Merge topic	2	3	2	5
Split topic	1	5	1	1

but these would only be applied to the model once they clicked "save," which they could do at any time. Save usage varied substantially ($min = 0$, $max = 42$, $avg = 14$, $\sigma = 12$); see Table 6.3.

Users were asked about their strategies for when to click "save": "after each refinement" (4 participants), "after each topic modified" (2), "after a batch of refinements" (2), "when sure" (2). These varied strategies suggest that interactive topic modeling systems should allow users to choose when to save their refinements. Additionally, two participants "forgot to save", and another was "afraid to save", which suggests that systems should remind users to save and support undo. "Save" counts and strategy feedback are shown in Table 6.3.

P8 saved the most frequently (42 times) and described his strategy as saving after each refinement, saying, *"I always press the save button when I make any refinements."* P9 saved 28 times, saying, *"only when I am very sure about the result, I would press the save button."* In contrast, P6 and P1 reported that they forgot to save, and four other participants stated that remembering to save was one of the main challenges of using the system. P12 wondered, *"if moving the [save] button over from the side would have helped me remember [to save]."* Finally, P3 was afraid to save, saying,

Table 6.3: Save strategies described by participants and the number of times each participant saved during the task, ordered from most to least iterations. There was no dominant strategy: save usage and strategy varied across participants.

Participant	Iterations	Save Strategy
P8	42	After each refinement
P9	28	When sure
P12	19	After a batch of refinements
P2	19	After each topic modified
P7	18	After a batch of refinements
P10	16	After each refinement
P11	15	When sure
P4	9	After each topic modified
P5	8	After each refinement
P1	3	Forgot to save
P6	1	Forgot to save
P3	0	Afraid to save

"I didn't want to start from scratch." She suggested that having a history of refinements that could have been rolled back might mitigate timidity.

Did participants improve the model?

To determine if participants improved the initial topic model using the interactive topic modeling system, we measured the quality of the initial topic model and the final topic models using qualitative and quantitative methods.

All participants started with the same model. We computed topic quality for the initial model and final models using a topic coherence metric based on NPMI (Lau et al., 2014) (see Chapter 3.2.2). The average topic coherence for the 10 topics of the initial model was .024 (min = .01, max = .04, σ = .01) (per-topic coherence shown in Table 6.1). The average topic coherence for the final model for each participant ranged from .021 to .037 (M = .027, σ = .005), which a paired t-test showed a significant improvement from the refinement process, $t(10) = 2.89$, $p = .037$.

Participants gave their satisfaction with the topic model before and after the task on a scale from one to seven, with one being not at all satisfied and seven being very satisfied. The average subjective model satisfaction increased from 4.7 ($\sigma = 1.3$) before the task to 5.2 ($\sigma = 0.8$) after the task. While this increase was not statistically significant by a Wilcoxen signed rank test ($Z = -1.04$, $p = .15$), six of the 12 participants commented unprompted after the task that their final model provided a good organization of the complaints. For example, P5 said, *"I'm overall happy with the [final] model and I like that I can use the system to make the changes that I want."*

Participants gave the best and worst topics in the initial model (Table 6.1). Most participants agreed the best topics were T4 (4 participants), T3 (3), T1 (3), and T9 (3) and the worst topics were T5 (8), T6 (3), and T8 (2), which correlates with the observed topic coherence. The three best topics by NPMI are T4 (NPMI=.045), T1 (.031), and T9 (.030), while the three worst topics are T8 (.009), T2 (.014), and T5 (.015).

What challenges did participants face?

A primary goal of this study is to understand how interactive machine learning system characteristics (e.g., latency, control, predictability) affect users of interactive topic modeling. We discussed these characteristics in detail in Chapter 2.2.

To this end, we coded four common characteristics—tracking complex changes, instability, lack of control, latency—and identified other challenges with our system. Participants also stated which challenges were most and least frustrating during the task. Of the four common challenges, tracking complex changes was the most frustrating, followed by instability, lack of control, and latency was the least frustrating challenge.

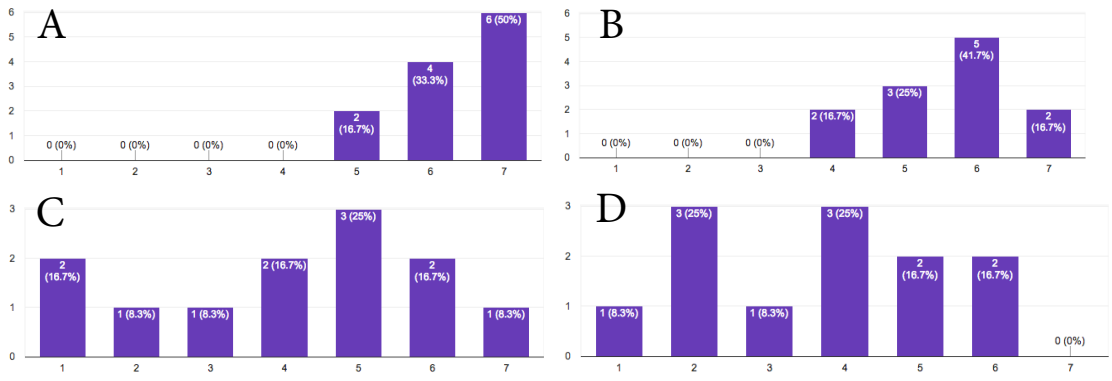


Figure 6.2: Counts for responses on a scale from one to seven for participants' agreement with statements related to latency (A), lack of control (B), instability (C), and tracking complex changes (D), with seven meaning they did not experience it and one that they did. Most participants found that the system updated quickly and refinements were applied as expected, while there was substantial variance for if participants could remember what the model looked like before updating or if they felt the updated model included other changes than those specified.

Tracking complex changes

When participants clicked "save," the algorithm updated the model, and the resulting model may have had substantial changes. To explore whether participants could track these changes, they rated their agreement with the statement, *"I was able to remember what the model looked like before my updates"* on a scale from one, meaning no agreement, to seven, meaning complete agreement (Figure 6.2, D) and discussed how this affected them. The average response was 3.7 out of 7 ($\sigma = 1.7$), and four of the 12 participants said this was the most frustrating challenge while one said it was the least.

Five participants said not being able to remember what the model looked like hurt their performance. For example, P9 said, *"a moment ago, I was satisfied with this topic, but now it's gone, and I don't think I am, but I can't remember,"* and P3 and P8 felt the lack of "undo" intensified this challenge. P3 said, *"I think this is a big issue–I'd like to know if I'm capturing the true data–and be able to step back to early versions of the model before saving,"* and P8 said, *"I don't know what I have done sometimes, and there are no ways to go back . . . "*. Four participants mentioned a

similar challenge, that it was hard to tell what changed in the model after an update, such as P10, who *"had to brush through all the words to confirm if [his specified] change occurred,"* and P5, who *"did not understand it at first, that the model actually changes, as there was no feedback or indication."* Finally, three participants requested a long-term history view of the model, such as P3, who suggested *"having a history of refinements."*

Stability

We asked if participants agreed with the statement, *"no changes other than the refinements I made occurred when I clicked update"* on a scale from one, or no agreement, to seven, or complete agreement (Figure 6.2, C). The average response was 4.1 out of seven (σ = 2.0), and three of 12 participants said instability was the most frustrating challenge while no participants said it was the least.

There was a large variance for not only whether users perceived instability but also their reactions to it. After the task, eight of 12 participants mentioned they had perceived instability. Of those, two participants found this to be positive. For example, P6 observed an unspecified change when *"new words were added on to the list to replace the ones I removed. It made the model better."* P2 noted that after removing some words from a topic there was *"some slight surprise at seeing words that I had not chosen show up, but I was pretty satisfied on looking at the results."* Three participants felt neutral about the instability. For example, P7 said, *"[instability] did not impact"* his ability to perform the task, and P4 said, *"when I removed some keywords, other keywords came up. I wasn't paying enough attention to this to determine if it helped or harmed."* Finally, three participants had negative reactions, such as P9, who was unsure of what had changed in the model after an update, but stated, *"... but I remember being happy with the topic and when that changed it made me unhappy."* This participant also requested the ability to *freeze a topic*, meaning it would

not be changed as other refinements were made.

Control

To explore whether participants felt in control of the system, they stated on a seven-point scale whether they agreed with the statement, *"the refinements I made were applied as expected when I clicked update"* from not agreeing at all to completely agreeing (Figure 6.2, B) and discussed how this affected their task. The average response was 5.6 out of seven ($\sigma = 1.0$), meaning overall users found the system to be fairly controllable. One of the 12 participants said lack of control was the most frustrating challenge and one said it was the least.

However, during the task seven participants noted frustration with the lack of control with the interface, and five participants specifically observed that **change word order** was uncontrollable. P4 tried to drag important words to the front of the topic list and stated that, *"the reordering didn't always get accepted,"* and P8 tried to drag unimportant words to the end of the list and said, *"I tried to move this word and it just goes back up."*

Latency

Our refinement implementation is efficient by design, and the data set used in this user study was relatively small (both in document size and length), therefore the algorithm updated almost instantaneously during the task (.09 – .63 seconds). No participant said that latency was the most frustrating challenge while two participants said it was the least frustrating, and the average response was 6.3 out of seven ($\sigma = 0.8$) for participants agreement with the statement, *"after clicking the update button, the model updated quickly"* (Figure 6.2, A).

However, for a more realistic corpus size or alternative refinement implementation, latency becomes a challenge. We asked participants to describe how their ability to perform the task would

be affected had the wait time been 10 seconds, 30 seconds, two minutes, or 10 minutes. Most participants felt 10 seconds would be an acceptable time to wait: five participants felt that waiting 10 seconds would have no effect on the task and two participants felt that this longer wait time would have a positive effect, for example, P5 stated that waiting longer *"would be better for me to realize that the tweets have changed."* For a 30 second wait time, two participants felt this would be an acceptable wait time without any changes to the interface, whereas four participants said that changes to the interface would be required for this longer wait time. P7 worried this wait would further hinder the ability to remember what the model looked like before updating, and P3 thought this would further affect save strategy, suggesting that it would instead be *"better to not 'save' changes, but to have highlights to show what it 'might' look like once saved."* Most participants felt that both two minutes and 10 minutes would be unacceptable wait times.

Trust and confidence

Trust is a primary design goal for intelligent systems (Höök, 2000; Norman, 1994). Surprisingly, we did not see participants mistrusting the interactive topic modeling system. Unlike in our experiments in Chapter 4, where users distrusted the low accuracy classification system that produced easily identifiable, incorrect classifications, topic models have less obvious *incorrect* answers.

However, participants sometimes put too much trust in the system or lacked self-confidence. For example, P10 was confused about a topic word, saying, *"if the system coughed it up, there must be a reason for it, right?"* Some participants lacked confidence in their refinements: P7 said that **remove document** is the least useful refinement, because, *"I don't feel comfortable removing a document."* And when P5 added words to a topic, she said, *"it's putting my words on top . . . I've added too many words, which have gone to the top of the list, so either the algorithm thinks it's important or it's because I've added them,"* followed by, *"I don't think that it should always give*

more importance [to my added words], because I could be wrong!" This challenge has a direct connection to the issues of instability and lack of control, which we discuss in more detail in Chapter 6.4.

What other requests did participants have for the system?

Many participants wanted a better understanding of the model and the data. For example, two participants requested a better model overview, such as P7, who wanted to *"see the entire list of the top 20 words for each topic on one screen to allow for making bulk, faster changes."* Additionally, three participants wanted to view words or documents across topics, such as P12 who suggested, *"a note or color to indicate that a certain term appears only in this topic and not in the others."* Two participants requested enhancing the word in context feature, such as by scrolling to the selected word or filtering to only documents containing the word. Three participants wanted to view more documents than the 40 shown, and two participants wanted to view the total number of documents for a topic.

Similar to the **merge word** operation identified by Lee et al. (2017), six participants requested a refinement to add phrases (instead of just single words), and four participants requested a refinement to group synonyms and plurals. As anticipated, participants used the **add to stop words** refinement, and two participants requested an enhancement to the stop words functionality, such as being able to view the stop words list and remove words that have been added to it. However, seven participants noted confusion between the **add to stop words** and **remove word** refinements, which should be clarified in future interface design. For example, P5 said, *"removing a word feature is similar to the delete feature, which got me a bit confused,"* and P9 said, *"I got confused between removing keywords from a particular [topic] and the overall [topics], so I made mistakes in the beginning."* To help better organize the view, three participants wanted to name topics,

noting that it would be a useful way to remember what the topics are about, and two participants wanted to reorder topics in the list. Finally, two participants wanted to delete a topic if it was particularly bad.

6.3.1 Summary

Participants were frustrated by their inability to track how the model changed throughout the refinement process. While participants perceived system instability, they had varied reactions (positive and negative). On the other hand, users did not experience substantial latency or lack of control. We did not find any cases where users distrusted the system, but users perhaps trusted the system too much or had too little confidence in themselves. Participants specifically requested the ability to undo changes after saving and to curate the topic model view, such as by re-ordering the topic list, removing poor quality topics, and naming topics. Participants also requested multi-word refinements, such as adding phrases and grouping synonyms.

6.4 Discussion

In this section, we outline implications for future interactive topic modeling system design, discuss open questions related to interactive machine learning, and provide a reflection on our interactive topic modeling implementation.

6.4.1 Design Recommendations

From our findings and those of related work, we distilled design recommendations, which we detail in the following.

Provide richer history

Participants voiced concerns with their inability to remember the history of the model, and four of 12 participants said they were unable to tell how the model has changed after an update. Interactive topic modeling interfaces should strive to support visualization of short term and long term model changes; users want to track how the model changed throughout the refinement process. This was the most consistent and most frustrating issue in the study.

Support undo

In our studies, participants noted that the lack of undo meant they were afraid to save during the task and some specifically requested an undo functionality for the tool. Similarly, user interface design guidelines highlight the importance of "undo" for removing anxiety and encouraging exploration (Shneiderman, 1996). Therefore, when possible, interactive topic modeling should support reverting to prior states of the model.

Allow users to choose when to save, but remind them to do so

We had anticipated needing a separate save action to allow users to confirm refinements (lacking undo) and to counteract latency, but we also noted users who created refinements as a data exploration system without the intent of having them update the model. Thus, interactive topic modeling systems should allow users to choose when to save their refinements to the model instead of forcing a save. However, because users forget to save, additional information should be provided in the interface to remind users, such as a more prominent count of outstanding refinement operations or a visual cue that displays if they have not saved recently.

Freeze topics to protect from instability

Users complained of instability when topics that were once high quality or about a particular thing had changed. A process, such as freezing a topic, suggested by one participant as a mechanism to hold a particular topic constant during subsequent updates, is a promising solution to this problem and should be incorporated in future design.

Support multi-word refinements

Participants requested the ability to add phrases and group synonyms. Group synonyms could be implemented as the merge word refinement discussed in Lee et al. (2017), not as an update to the underlying model, but as a way of organizing words in the interface. On the other hand, add phrases should be implemented in the interface as an extension to **add word** (as requested by participants), but would likely require a more complex modeling approach that supports *n*-grams as opposed to single tokens.

Clarify difference between adding a word to stop words and removing it from a single topic

Future design should explicitly delineate between removing a word from all topics (and the modeling process entirely), **add to stop words**, and removing a word from a single topic, **remove word**, as many participants confused the two operations during the task.

Support user-curated model view

Three participants requested named topics. Two other participants used **change word order** for *ad hoc* topic naming. As this operation is not always applied as expected, providing a control-

lable topic naming functionality will improve user experience. Participants also requested other techniques for curating their model view, which should be incorporated in the design of future systems, such as the ability to re-order the topic list and to remove poor quality topics entirely.

6.4.2 Open Questions

This is the first system to efficiently implement a full suite of refinements desired by users in prior work (Lee et al., 2017; Musialek et al., 2016), enabling the study of true human-in-the-loop interactions of a comprehensive interactive topic modeling system. We enumerate open questions about interactive topic modeling design that follow from our findings.

Trust vs. instability and control

Users were not bothered by instability or lack of control either because they trusted the system or had little confidence in themselves. Specifically, users with limited confidence blamed *themselves* for creating poor refinements (i.e., when the change did not happen as anticipated). If system builders do not want novice users to feel like the "junior partner" in the human-machine collaboration, future work should explore whether ensuring users understand the teaming aspect of these systems can improve their experience and make unpredictability more acceptable (and sometimes welcome, as it can drive discovery).

Trust, control, and refinement

Lee et al. (2017) studied refinement usage without a refinement implementation, meaning users did not see the full effect of their refinements on the model. In that study, **remove document** was a commonly used refinement, however, that is not the case in our study. Before the study participants

worried that it may take too long to determine which documents to remove, while afterwards noted they lacked confidence to remove a document. Although Lee et al. (2017) considered that refinements that take too long would hurt usage, lack of trust or confidence in interactive topic modeling is a new challenge to consider.

Change word order was commonly used, but frustrating to users, while **add word** was used less, yet all participants who used it thought it was useful. This discrepancy highlights the difference in control of the two refinements: **change word order** was unpredictable and thus frustrating, but **add word** always worked on the first try.

Save strategy and instability

When users save after a batch of refinements (as opposed to a single requirement) their intentions are clearer. This in turn minimizes instability as the system has more information to incorporate into the model. On the other hand, each refinement may have cascading effects, and a batch of refinements could therefore appear to be more unstable than a single refinement. We did not find a relationship between users' described save strategies and their perceived system instability. Future work should explore the relationship with a specific focus on how much information users provide to the system and whether this information affects the system's stability and how users react.

6.4.3 Algorithm Reflection

This chapter proposes an asymmetric prior-based interactive topic modeling implementation. We implemented seven refinement operations using the proposed algorithm, which can be easily extended to other refinements, such as creating a new topic using seed words or deleting a topic. One limitation of this algorithm is the difficulty to specify word order constraints. For example,

if a user wants to change a word's position from rank eight to two in the word list, the algorithm cannot reliably maintain the exact user provided word order. We argue that topic models are probabilistic models and during parameter estimation they can ignore user provided feedback if the underlying data does not support the user's hypothesis. For example, if a user wants to add a word to a topic that only shows up a few times in the corpus, the model might not put that word in the list of top ranked words for that topic. Another limitation of our algorithm is with the **split topic** refinement: our proposed implementation cannot reliably generate a good quality topic if the user provides only very few or unrelated seed words.

6.5 Conclusion

Prior work in interactive topic modeling either implemented refinement operations without first understanding the needs of end users (Choo et al., 2013; Hoque and Carenini, 2015; Hu et al., 2014) or identified the refinement operations that users wish to do (Lee et al., 2017), but did not implement them. The work in this chapter was the first to examine user experience with a fully-functional interactive topic modeling system that contains the refinements users want. Specifically, we validated prior results, such as refinement usage and effectiveness, and explored how these and user experience are affected by previously hidden system characteristics, such as instability and lack of adherence. We also presented suggestions, such as the need to visualize complex model changes and support undo. Non-expert end users used the system to refine a topic model and we explored how these users perceived and were affected interactive machine learning system characteristics, such as control, unpredictability, and latency. Participants improved a topic model using the system and identified additional refinement and system suggestions that should guide future interactive topic modeling development.

In this chapter, we demonstrated that when users *controlled* transparent systems, in this case interactive topic modeling, they perceived unpredictability; however, their reactions varied from positive to negative. And, surprisingly, we found that overall users trusted our system and in some cases perhaps even trusted it too much or had too little confidence in themselves. In Chapter 7, we build on these findings and further examine control in transparent, interactive systems. In particular, we present a study comparing three distinct interactive topic modeling implementation variants, which result in varied system characteristics—adherence, stability, update speeds, and model quality, to determine whether users perceive differences between the systems and if their differences affect user experience. Based on the findings presented in this chapter, we also enhanced our initial interactive topic modeling system with additional refinements: create topic and delete topic, support for renaming topics, support for undo, and we better distinguished between removing words from single topics opposed to all topics (stop words) in the tutorial.

Chapter 7: Predictable Control in Transparent Systems: a Comparative Study[1]

Human-in-the-loop techniques allow users to guide unsupervised algorithms by exposing and supporting interaction with underlying model representations, increasing transparency and promising fine-grained control. However, these models must balance user input and the underlying data, meaning they sometimes update slowly, poorly, or unpredictably—either by not incorporating user input as expected (*adherence*) or by making other unexpected changes (*instability*).

Building on the exploratory study in Chapter 6, this chapter explores user perceptions of control and instability with transparent systems—where controls are easy to validate—using a study where 100 participants performed a document organization task with one of three distinct interactive topic modeling approaches. These approaches incorporate input differently, resulting in varied adherence, stability, update speeds, and model quality. Participants disliked slow updates most, followed by lack of adherence. Instability was polarizing: some participants liked it when it surfaced interesting information, while others did not. Across modeling approaches, participants differed only in whether they noticed adherence.

[1]The work in this chapter was performed in collaboration with Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater and was accepted to the International Conference on Intelligent User Interfaces (IUI) 2020; the algorithm details, particularly the modeling approaches and refinement implementations, were published as "Varun Kumar, Alison Smith-Renner, Kevin Seppi, Leah Findlater, and Jordan Boyd-Graber. Why Didn't You Listen to Me: Comparing User Control of Human-in-the-Loop Topic Models. In Association for Computational Linguistics (ACL), 2019 (Kumar et al., 2019)"

7.1 Method

For this study, crowd workers interacted with a topic model to organize documents using one of three contrasting interactive topic modeling approaches, which support the same set of nine refinement operations (e.g., merging topics and removing words or documents from topics), and differed only in implementation details, as these criteria affect model attributes, such as adherence, instability, quality, and latency.

This study used a between-subjects experimental design with a single factor, *Modeling Approach*, with three conditions: informed priors using Gibbs sampling (*info-gibbs*), informed priors using variational inference (*info-vb*), and constraints using Gibbs sampling (*const-gibbs*).

The goal of this study was to explore how users perceive and interact with transparent systems with varied characteristics: adherence, instability, latency, and quality. This study explored specifically: (RQ1) How do users perceive instability and adherence across the three modeling approaches? (RQ2) How does user experience vary given these differing characteristics? (RQ3) How do users behave with the three modeling approaches?

7.1.1 Modeling approaches

We implemented three distinct interactive topic modeling systems, based on LDA, following three modeling approaches. Recall from Chapter 3.2.4 that interactive topic modeling approaches incorporate user feedback by first *forgetting* what the model learned before, by unassigning words from topics (Hu et al., 2014), and then *injecting* new information based on user feedback into the model.

Our three modeling approaches differ in how user input (e.g., added words) is injected to the

model—informed priors (Smith et al., 2018) or constraints (Yang et al., 2015)—and how inference (and forgetting) is performed—variational inference (Blei et al., 2003) (Equation 3.5) or Gibbs sampling (Griffiths and Steyvers, 2004) (Equation 3.4).

We compare two existing techniques for *injecting* new information: (1) informed priors (see Chapter 6), which are used extensively for injecting knowledge into topic models (Pleplé, 2013; Wang et al., 2019; Zhai et al., 2012) by modifying Dirichlet parameters, α and β, and (2) constraints (Yang et al., 2015), in which a knowledge source m is incorporated as a potential function $f_m(z, m, d)$ of the hidden topic z of word type w in document d.

We also compare two inference techniques for topic models (detailed in Chapter 3.2.1): (1) Gibbs sampling and (2) variational EM. Recall from Chapter 6, in Gibbs sampling information is forgotten by invalidating topic-word assignments (i.e., setting them to -1) and adjusting associated counts. Here we discuss how information can be forgotten in Variational EM. First, recall from Chapter 3.2.1 that Variational EM defines a mean field variational distribution,

$$q(z, \phi, \theta | \lambda, \gamma, \pi) = \prod_{k=1}^{K} q(\phi_k | \lambda_k) \prod_{d=1}^{D} q(\theta_d | \gamma_d) \prod_{n=1}^{N_d} q(z_{dn} | \pi_{dn}) \tag{7.1}$$

where γ_d, π_d are local variational parameters of the distribution q for document d, and λ is a global variational parameter. Inference minimizes the KL divergence between the variational distribution and true posterior in the following EM algorithm (Geigle, 2016):

E-step: Minimize KL divergence from p to q for each document d by performing the following updates until convergence:

$$\pi_{d,n,i} \propto \lambda_{i,w_{d,n}} exp\left(\psi(\gamma_{d,i}) - \psi\left(\sum_{k=1}^{K} \gamma_{d,k}\right)\right) \tag{7.2}$$

$$\gamma_{d,i} = \alpha_i + \sum_{n=1}^{N_d} \pi_{d,n,i} \tag{7.3}$$

Where $\psi(\bullet)$ is the "digamma" function; q is now a good approximation to the posterior distribution p.

M-step: Using q, re-estimate λ. Specifically, since $\pi_{d,n,i}$ represents the probability that word $w_{d,n}$ was assigned to topic i, we compute and re-normalize expected counts:

$$\lambda_{i,v} = \beta_v + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \pi_{d,n,v} I(w_{d,n} = v) \tag{7.4}$$

Where $I(\bullet)$ is the indicator function that takes value 1 if the condition is true and value 0 otherwise.

The parameter $\lambda_{t,w}$ encodes how closely the word w is related to topic t. Since, λ is a Dirichlet, $\lambda_{t,w}$ can be viewed as pseudocount that represents the number of times word w was assigned to topic t. Essentially, in the *E-step*, the model assigns latent topics based on the current value of λ (Equations 7.2 and 7.3), and in the *M-step*, the model updates λ using the current topic assignments (Equation 7.4). Because the model relies on a fixed λ for topic assignment—essentially a memory of topic-word information, information for a word w in a topic t can be forgotten by resetting $\lambda_{t,w}$ to the default prior $\beta_{t,w}$. Therefore, to forget what is known about word w in topic t, $\lambda_{t,w} = \beta_{t,w}$.

For both Gibbs sampling and variational EM, we make use of the vector interpretation (rather than scalar) for the priors, α and β. Thus, α_d is a K dimensional vector for each document d and β_t is a V dimensional vector for each topic t, where K represents the number of topics and V is the size

of the vocabulary.

Together, combinations of these injection (priors and constraints) and inference (i.e., forgetting) techniques (Gibbs and variational) result in three modeling approaches:

Informed priors using Gibbs sampling (*info-gibbs*) forgets topic-word assignments for a word w in topic t by assigning an invalid topic (-1) for w and updating associated counts. This approach injects new information by modifying Dirichlet parameters, α and β. This implementation mirrors that of Chapter 6.

Informed priors using variational inference (*info-vb*) forgets topic-word assignments for a word w in topic t by resetting the value of $\lambda_{t,w}$ to the default prior, $\beta_{t,w}$. For injecting new information, like in *info-gibbs*, this approach manipulates priors, α and β.

Constraints using Gibbs sampling (*const-gibbs*) forgets topic-word assignments like in *info-gibbs*, but instead of manipulating the priors (α and β), injects new information into the model using potential functions, $f_m(z, m, d)$, as demonstrated by Yang et al. (2015).

While other topic modeling approaches exist (Hofmann, 1999; Larochelle and Lauly, 2012), we chose these LDA-based variants because they support the same user-preferred refinement set. For example, "anchor words"-variants (Lund et al., 2017) also generate topics, but cannot support word-level operations like adding words. Also, these chosen approaches may differ by the attributes we are interested in examining. For example, prior work asserts that informed priors better *adhere* to refinement operations (Kumar et al., 2019), and Gibbs sampling-based methods can yield more coherent topics than variational inference (Nguyen et al., 2015). Also, Gibbs sampling and variational inference have different convergence rates (Asuncion et al., 2009). While Gibbs sampling is often preferred for small datasets and interactive settings because of its low latency, variational inference can scale to millions of documents (Hoffman et al., 2010; Zhai et al.,

2012). Our setting allows a focused, task-center comparison (Chapter 7.2.1).

For the Gibbs sampling conditions, *info-gibbs* and *const-gibbs*, we trained initial LDA models with 300 Gibbs sampling iterations and default Mallet toolkit[2] hyperparameters ($\alpha = 0.1; \beta = 0.01$) and, for the variational inference condition, *info-vb*, we ran 30 EM iterations. For each subsequent update during the task, we applied the refinement and ran inference.

7.1.2 Refinement implementations

For each of the three modeling approaches, we implemented nine refinement operations previously requested by users (Lee et al., 2017; Musialek et al., 2016; Smith et al., 2018). These refinements are the same from the study in Chapter 6 with the addition of **create topic** and **delete topic**. Specifically, the refinement set includes four topic-level refinements: **add word**, **change word order**, **remove word**, **remove document** and five model-level refinements: **merge topics**, **split topic**, **create topic**, **delete topic**, **add to stop words**.

In what follows, we provide detailed refinement implementation details for the three modeling approaches. Keep in mind that the implementation for *info-gibbs* is the same as in Chapter 6.

- **Add word**: to add word w to topic t, for all three approaches, we first forget all w's tokens w_i from all other topics except t. Specifically, for *info-gibbs* and *const-gibbs*, we get w_i's topic assignment t_i, and if it is not equal to t, we decrement the associated topic counts (n_{w,t_i} and n_{d,t_i}) and assign w_i to an invalid topic (-1). For *info-vb*, we forget by setting λ_{t_i,w_i} to the default prior β_{t_i,w_i} (i.e., 0.01). Then, to inject information about the added word, for *info-gibbs* and *info-vb*, we increase the prior of w in t ($\beta_{w,t}$) by the difference between the topic-word counts of w and topic-word counts of the topic's top word w' in t (i.e., $n_{w',t} - n_{w,t}$).

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

For *const-gibbs*, we add a constraint $f_m(z, w, d)$, such that $f_m(z, w, d) = 0$ if $z = t$ and $w = x$, else assign $log(\epsilon)$.

- **Remove word**: to remove word w from topic t, for all three approaches, we first forget all w's tokens w_i from t. For *info-gibbs* and *const-gibbs*, we forget by finding all w's tokens assigned to t and assigning them to an invalid topic (-1), while simultaneously decrementing associated counts. For *info-vb*, we forget by resetting $\lambda_{t,w}$ to the default prior $\beta_{t,w}$ (i.e., 0.01) for all w tokens assigned to t. Then, for *info-gibbs* and *info-vb*, we assign a very small prior[3] ϵ to w in t ($\beta_{w,t}$). For *const-gibbs*, we add a constraint[4] $f_m(z, w, d)$, such that $f_m(z, w, d) = log(\epsilon)$ if $z = t$ and $w = x$, else assign 0.
- **Change word order**: to ensure w_2 is ranked higher than w_1 in t, in *info-gibbs*, we increase the prior of w_2 in t ($\beta_{w_2,t}$) by the topic word counts' difference $n_{w_{1,t}} - n_{w_{2,t}}$. In *info-vb*, we increase $\beta_{w_2,t}$ by $\lambda_{t,w_1} - \lambda_{t,w_2}$. Finally, for *const-gibbs*, we compute the ratio r between the topic word counts' difference $n_{w_{1,t}} - n_{w_{2,t}}$ and the counts of word w_2, which have any topic except t, $n_{w_{2,x},x\neq t}$. Then, add a constraint $f_m(z, w, d)$, such that $f_m(z, w, d) = 0$ if $z = t$ and $w = w_2$, else assign δ where $\delta = \log(\epsilon)$ if $r > 1$ else $\delta = 1.0 - r$.
- **Remove document**: to remove document d from topic t, for all three approaches, we first forget the topic assignment for all word tokens in the document d. For *info-gibbs* and *const-gibbs*, we assign all words in the document to an invalid topic (-1) and decrement $n_{w,t}$. We also set $n_{d,t} = 0$. For *info-vb*, we reset $\lambda_{t,w}$ to $\beta_{t,w}$ for all words in d. Then, for *info-gibbs* and *info-vb*, we inject information about the removed document by assigning a very small prior,[3] ϵ, to t in α_d. For *const-gibbs*, add a constraint $f_m(z, w, d)$, such that $f_m(z, w, d) = log(\epsilon)$ if $z = t$ and $d = x$, else assign 0.
- **Merge topics**: to merge topics t_1 and t_2 into a single topic, t_1, for *info-gibbs* and *const-gibbs*,

[3] We use $\epsilon = 10^{-8}$

[4] We use $log(\epsilon)$ to make it a soft constraint. Replacing it with -∞ will make it a hard constraint.

we assign t_1 to all tokens previously assigned to t_2 and update the associated counts. This effectively removes t_2 and updates t_1, which should represent both t_1 and t_2. For *info-vb*, we add the Dirichlet parameter (or pseudocount) λ_{t_2} to λ_{t_1} and remove the row from λ that corresponds to t_2.

- **Split topic**: to split topic t given seed words into two topics, t_n, containing the seed words, and t, without the seed words. For each vocabulary word, we move a fraction of probability mass from t to t_n as proposed by Pleplé (2013). Then, for *info-gibbs* and *info-vb*, we assign a large prior[5] for all seed words in t_n. For *const-gibbs*, to push the seed words s to t_n, we add a constraint $f_m(z, w, d)$, such that $f_m(z, w, d) = 0$ if $z = t_n$ and $w = w_i \in s$, else assign $log(\epsilon)$.
- **Create topic**: to create a topic t_n given seed words, we first forget all previous topic assignment for all of the seed words' tokens (as in **add word** and **remove word**). Then, for *info-gibbs* and *info-vb*, we assign a large prior[5] to the seed words for t_n. For *const-gibbs*, to assign the seed words s to t_n, we add a constraint $f_m(z, w, d)$, such that $f_m(z, w, d) = 0$ if $z = t_n$ and $w = w_i \in s$, else assign $log(\epsilon)$.
- **Delete topic**: to delete a specified topic t, in all three approaches, we first forget all word-topic assignments which were assigned to t, and then we decrement the number of topics by (i.e., reduce by 1).
- **Add to stop words**: to add a word w to the stop words list, we exclude w from the vocabulary.

When participants use these refinements we apply them and run inference for fixed N iterations to limit latency (rather than running inference until convergence). Moreover, all refinements have different levels of complexity, meaning the models converge faster for certain refinements than others. For example, **add to stop words** is a simpler refinement than **create topic**, and hence

[5]Following Fan et al. (2017), we use 100 as the large prior.

requires fewer iterations to converge. For each refinement, we empirically fine-tuned N on 9000 tweets randomly selected from a different dataset.[6] In particular, to fine-tune N for a refinement, we randomly applied a refinement multiple times and observed how fast the model converged. For *info-gibbs* and *const-gibbs*, N ranged from one for **add to stop words** to 20 for **create topic**. For *info-vb*, N varied from one for **add to stop words** to four for **create topic**.

7.1.3 Dataset

For this study, we used the Twitter Airline Sentiment Dataset, which includes tweets directed at various common airlines (e.g., United, Southwest Airlines, Jet Blue) and manually tagged by sentiment (positive, negative, neutral).[7] We produced initial topic models of 10 topics from only the 9,178 negative sentiment tweets, as these reflect a distinct set of complaints regarding air travel.

7.1.4 Task interface

We use a similar interactive topic modeling interface as that of the study in Chapter 6, but with some enhancements based on the findings of that study. The user interface was the same for all three modeling approaches (Figure 7.1). Like in the prior study, the topics are listed on the left, each initially represented by a generic topic label (e.g., "Topic 1") and the three most probable words for the topic. The currently selected topic is on the right, which displays the top 20 topic words and the top 20 topic documents. Documents are ordered by their probability for the topic t given the document d, or $P(t|d)$. Each word, w, is ordered and sized by its probability for the topic t, or $P(w|t)$; recall that this simple word list representation provides a quick understanding

[6] https://www.kaggle.com/kazanova/sentiment140/

[7] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

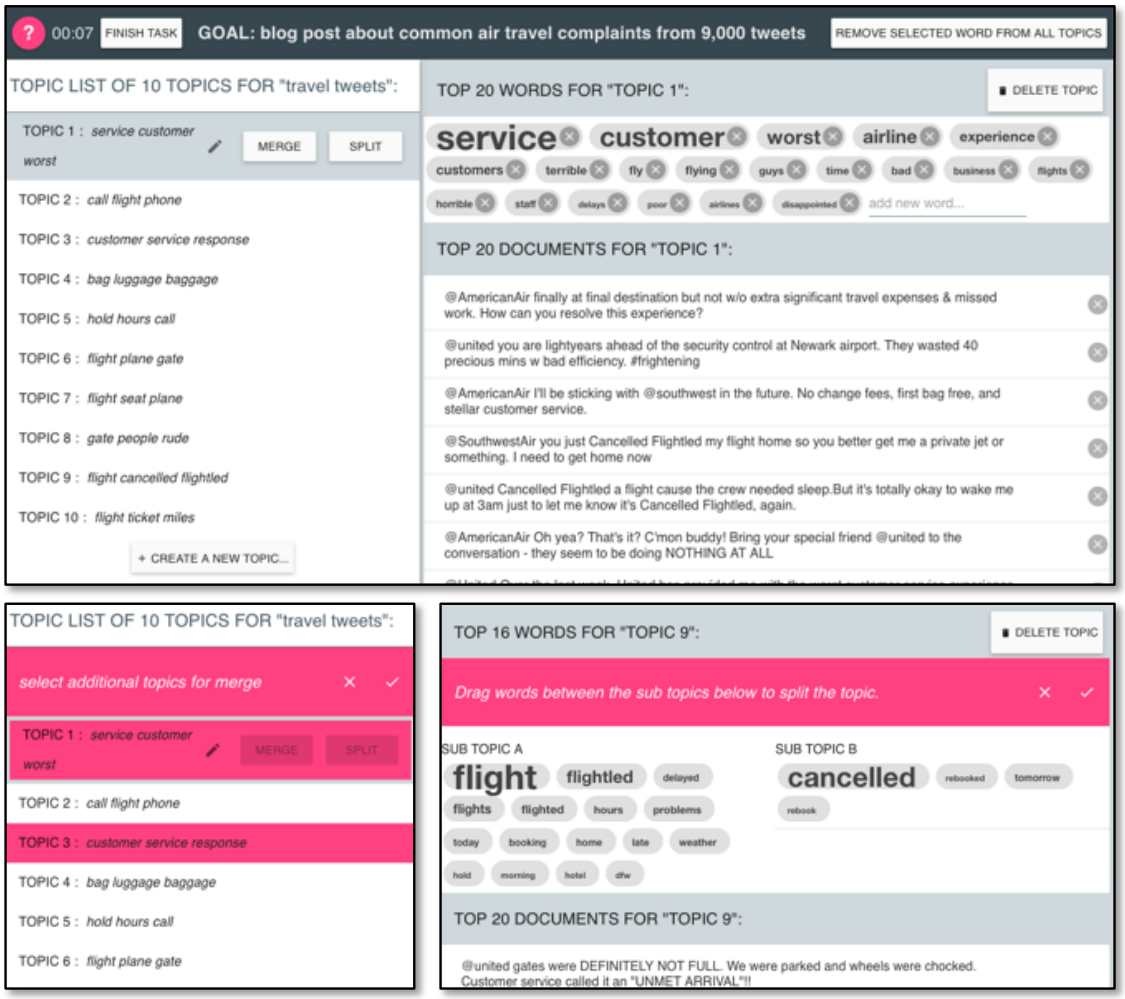


Figure 7.1: User interface for the interactive topic modeling systems. Initial model (top) represented as a list of topics, each displayed with topic name and three most probable words. Selecting a topic reveals more detail: the top 20 words and top 20 documents. Participants interacted with the model to refine it, including merging topics by clicking the "merge" button next to the topic and selecting additional topics with which to merge (bottom left), and splitting topics by clicking the "split" button next to the topic and dragging to separate words into sub topics (bottom right).

of the topic (Alexander and Gleicher, 2016; Smith et al., 2017). Hovering or clicking on topic words highlights the word in the displayed document snippets.

Participants in our previous study in Chapter 6 used **change word order** as a strategy for renaming topics in the topic list. This lead to frustration as **change word order** was a particularly unpredictable refinement. Therefore, we provide an explicit topic renaming mechanism in the updated interactive topic modeling system studied here; participants can click the pencil icon to rename

the topic labels to be more descriptive.

Participants can explore and update the model using the set of nine refinement operations: click "x" next to words or documents to remove them, select and drag words to re-order them, type new words into the input box and press "enter" to add them, select a word and click "remove selected word from all topics" to add it to the stop words list, click "delete topic" to remove the selected topic, or click "create a new topic," "split," or "merge" (in the topic list) to enter into create, split, or merge modes, respectively (Figure 7.1).

Unlike in the study in Chapter 6, each refinement is immediately saved and the model is updated; save after each refinement is required here as we compute per-refinement instability and adherence. Based on this change and on participants' feedback from the prior study, we provide undo support in our updated interactive topic modeling system; after updates, participants can *undo* to revert their models to prior states.

7.1.5 Participants

We recruited 100 participants (32 male and 68 female) on the Upwork platform.[8] Participants were required to have a 90% or higher job success score and be native or bilingual English speakers. The task was designed to take approximately 60 minutes, and participants were paid 20 USD. We used Upwork instead of other common crowdworker platforms (e.g., Mechanical Turk), to recruit more motivated participants; participants were paid a higher rate and are in contact with one of the researchers throughout their session in case of questions.

Participants varied in age (< 19: four, 20 – 29: 46, 30 – 39: 23, 40 – 49: 13, 50 – 59: seven, > 60: eight), education (college degree: 49, graduate degree: 29, some college: 17, high school or GED:

[8] https://www.upwork.com/

5), and background (most common include 12 participants with background in English or writing, seven in education, and five in business).

To understand participants' prior exposure to topic models and machine learning, as this could affect our results, study participants rated prior experience with statistical topic modeling and machine learning, respectively. Participants varied for prior experience (rated on a scale from one to five) with topic models ("none" (one): 44, two: 25, three: 18, four: seven, "significant" (five): six) and machine learning ("none" (one): 44, two: 19, three: 19, four: nine, "significant" (five): seven).

7.1.6 Procedure

Each participant was randomly assigned to one of the three modeling approaches and all used the same user interface (Figure 7.1). Each user got a unique starting model from a pool of 50 pre-trained initial LDA models with 10 topics for each of the three HL-TM modeling approaches. Given the assigned approach, we randomly selected an initial topic model from the pool of pre-trained models and then removed the selected model from the pool. The study began with a tutorial, which introduced participants to topic modeling, relevant terminology, and the task interface. The tutorial also required participants to experiment with each of the nine refinement operations. After the tutorial, participants were given the following task instructions:

> Imagine you have been asked to write a travel blog post about the common complaints that travelers have when flying. The system has gathered 9000 tweets of people complaining about their air travel experience directed at various popular airlines and has generated an initial set of 10 topics to organize these air travel complaint tweets. Use the tool to improve these topics, so that you can write a blog post about common air

Table 7.1: Seven-point rating scale statements for nine subjective measures. All are on a scale from "strongly disagree" to "strongly agree" aside from satisfaction, which is on a scale from "not at all" to "very" and improvement, which is on a scale from "much worse" to "much better."

Measure	Statement
frustration	"Using this tool to perform the task was frustrating"
trust	"I trusted that the tool would update the organization of tweets well"
task ease	"It was easy to use this tool to perform the task"
confidence	"I was confident in my specified changes to the tool"
final model satisfaction	"How satisfied are you with the final organization of the tweets into categories of air travel complaints?"
model improvement	"How do you think the final organization compares to the initial organization of tweets?"
low latency	"After my changes, the tool updated quickly"
adherence (overall)	"The tool made the changes I asked it to make"
instability	"The tool made unexpected changes beyond what I asked it to make"

travel complaints with a few example tweets from each. You do not need to write the actual blog post as part of this task.

Participants were then asked to spend 30 minutes interacting with the model, and to click the "finish task" button when they were happy with the organization they had achieved. The interface required that participants spend at least 20 minutes and no more than 45 on this task. The task goal and time elapsed were denoted in the task interface (Figure 7.1).

After the task, participants completed a survey containing closed- and open-ended questions on their perceptions and experience with the system (Table 7.1) and which refinements they felt were the most and least useful, with follow up "why" questions. Participants also responded to whether they noticed any unexpected behavior while using the tool and what they liked and did not like about using the tool for the task.

7.1.7 Measures

We report on nine overall subjective measures, collected using seven-point rating scales (Table 7.1): four *user experience* measures (frustration, trust, task ease, confidence) and five *user perception* measures perceived adherence, perceived instability, perceived latency, final model satisfaction, and perceived improvement. We also report on subjective per-refinement adherence, collected using seven-point rating scales (strongly disagree to strongly agree) for nine statements of the form, "the system incorporated the [refinement] operation as I asked it to." These statements also included a "did not use operation" option.

We also report on quantitative measures of the system characteristics: adherence, instability, latency, and quality (initial, final, and improved). To compute *adherence* for each of the nine refinements we used the metrics provided by Kumar et al. (2019):

- **add word**, **remove word**, and **change word order**: treat the topic as a ranked word list, and then take the ratio of the actual rank change (where the added, removed, or reordered word is in the updated model) and the expected rank change.
- **remove document**: compute similarly to **remove word**, except treat the topic as a ranked document list.
- **create topic**: compute the ratio of the number of seed words in the created topic out of the total number provided.
- **split topic**: compute the average adherence of the parent and child topic, using the adherence measure for **create topic**.
- **merge topics**: compute the ratio of the number of the words in the merged topic that came from either of the parent topics over the total number of words shown to the user.

- **add to stop words** and **delete topic**: these refinements are deterministic, and therefore always have a perfect adherence score.

Adherence is measured on a range from 0.0, meaning the system ignored the user's input, to 1.0, meaning the system did exactly as the user asks. The exception is adherence to **change word order**, which can range from $-\infty$ to ∞, where a negative adherence value meant the system did the opposite of what the user asked. For example, if a user moves a word up two positions, but it is instead moved down one, the adherence would be $-.5$. Overall adherence was computed as the average adherence score over all refinements applied by the user.

To estimate the *instability* caused by a refinement, we used a modified topic-term stability metric (Belford et al., 2018). We first computed the difference between each topic as 1.0 minus the overlap coefficient (M.K and K, 2016) between the top 20 words of the topic, before and after the refinement. Instability was then measured as the average difference between each topic excluding the refined topic(s). Put simply, we computed what percentage of topic words were removed after an update for the untouched topics. Instability was scored from 0.0 (all topics the same) to 1.0 (all topics completely different).

Latency is the time the model takes to incorporate each refinement. We also computed each participants' initial and final topic model quality as the models' average NPMI-based topic coherence (Lau et al., 2014); *quality* is thus the difference (i.e., improvement or degradation) from initial to final model quality.[9] We additionally logged all interactions with the system including how many and which refinements participants applied.

[9]Automatic coherence metrics require an external reference corpus for NPMI computation; as in prior work, we use Wikipedia. As the Twitter-based topics included many words not found in the Wikipedia reference corpus, their overall topic coherence scores were relatively low, but are still useful for *relative* comparison.

7.1.8 Data and analysis

We disqualified five of the 100 participants because they made an outlying number of survey response "mistakes" on per-refinement adherence statements. We considered a response to be a "mistake" if the participant said they had used a refinement for the task when they had not, or vice versa, and used an interquartile range (IQR) approach to determine outliers based on the count of mistakes (Tukey, 1977): the median number of mistakes was two, and the upper quartile bound for outliers ($Q3 + 1.5IQR$) was five (out of nine possible mistakes). Removing outliers above this bound resulted in 95 participants in our final dataset: 31 in the *info-gibbs* condition, 33 in the *const-gibbs* condition, and 31 in the *info-vb* condition.

For quantitative analysis, we used separate Kruskal Wallis tests to determine significance across the conditions for each of the subjective rating responses and the quantitative measures. For qualitative analysis, we followed a thematic approach (Braun and Clarke, 2006), and coded the open-ended responses related to what participants found unexpected, liked and did not like, and which refinements they found were most and least useful. Two annotators independently coded a random subset of 20 of the 95 responses for each of the statements regarding what was *unexpected*, what participants *liked*, and what they *did not like*; agreement was scored using Cohen's κ: $\kappa = .93$ for *unexpected* responses, $\kappa = .88$ for *liked* responses, and $\kappa = .89$ for *did not like* responses.

7.2 Findings

Each of the 95 participants started with a distinct initial random topic model and applied refinements with the goal of improving the model for their imagined travel blog.

In the following sections, we provide detailed results regarding computed model characteristics

followed by user perceptions, experience and behavior given those different characteristics, and with interactive topic models in general. We refer to participants throughout this section as P1-P95.

7.2.1 Computed Differences

The three modeling approaches differed significantly for four out of the five computed characteristics: adherence, instability, latency, and final model quality, but not model improvement (Table 7.2). The Gibbs sampling approaches (*const-gibbs* and *info-gibbs*) had higher final model quality than variational inference (*info-vb*), while variational inference was more stable than Gibbs. Informed priors with Gibbs sampling (*info-gibbs*) provided the fastest updates over *const-gibbs* and *info-vb*. Finally, informed priors (*info-gibbs* and *info-vb*) provided higher control than constraints (*const-gibbs*).

Analyzing adherence in more detail, Table 7.3 shows the average computed per-refinement adherence for each modeling approach. Computed adherence differed significantly across modeling approaches for four of the nine refinements: *const-gibbs* provided less control for **add word**, **change word order**, and **create topic** than the other approaches. For **split topic**, *info-vb* provided the most control followed by *const-gibbs*, and *info-gibbs* provided the least control.

7.2.2 User Perceptions

We analyzed participants' perceptions regarding adherence, instability, latency, and model performance through subjective responses (Figure 7.2 and Figure 7.3). While computed adherence, instability, latency, and final model quality differed across modeling approaches, for subjective measures, only adherence was significantly impacted by condition: participants in *const-gibbs*

Table 7.2: Computed measures for system characteristics: instability, adherence, latency (seconds), and performance—final model quality (coherence) and percent improvement. Coherence scores multiplied by 1000 for readability. Responses reported as "mean, σ." Kruskal-Wallis results reported as "$\chi^2(2)$, p." The modeling approaches differed significantly (bold) for all computed characteristics except improvement; cell shading for significantly different characteristics represents how that modeling approach compares to other approaches (darker is better).

	info-gibbs	*const-gibbs*	*info-vb*	Kruskal-Wallis
adherence	.84, .10	.70, .14	.82, .09	**20.8, p<.001**
stability	.12, .03	.12, .03	.03, .03	**1754.8, p<.001**
latency (s)	15.2, 6.2	19.3, 9.2	20.4, 5.9	**18.1, p<.001**
final quality	7.4, 3.5	.7.0, 1.9	.5.7, 1.5	**8.5, .014**
improvement	6%, 42%	4%, 34%	-7%, 30%	1.4, .489

perceived lower adherence than the other modeling approaches. It is important to note that we did not control for these characteristics nor for the magnitude of their differences, which may explain why users did not perceive differences in all dimensions.

Overall, participants thought the systems adhered to their input ($M = 5.3$ of 7, $\sigma = 1.8$), but were mixed on whether they observed instability ($M = 3.3$, $\sigma = 2.2$). Participants on average thought the final models showed improvement ($M = 5.8$, $\sigma = 1.1$) and they were satisfied with the quality ($M = 5.1$, $\sigma = 1.3$), but they thought the models updated slowly ($M = 2.7$, $\sigma = 1.6$).

Participants noticed when word-level refinements did not adhere

Adherence was lower for *const-gibbs* than other approaches (Table 7.2), particularly for three of the nine refinements: **add word**, **change word order**, and **create topic** (Table 7.3).

Participants thought that the system *adhered to* their input (Figure 7.2) more in the *info-vb* ($M = 5.7$, $\sigma = 1.6$) and *info-gibbs* approaches ($M = 5.5$, $\sigma = 1.5$) than *const-gibbs* ($M = 4.6$, $\sigma = 1.9$). These differences were significant ($\chi^2(2) = 6.3, p = .042$).

Perceived adherence was also significantly lower for *const-gibss* for two relatively easy-to-validate

Table 7.3: Computed per-refinement adherence measurements reported as "mean, σ". Kruskal-Wallis results reported as "$\chi^2(2)$, p." There were significant differences (bold) between modeling approaches for add word, change word order, create topic, and split topic; cell shading for these reflects how well that modeling approach adheres to that refinement compared to the other approaches (darker is better).

	info-gibbs	*const-gibbs*	*info-vb*	Kruskal-Wallis
add word	.99, .01	.62, .28	0.96, .04	**49.4, p<.001**
remove word	.91, .17	.97, .08	.99, .03	3.4, .180
remove doc	.78, .32	.88, .22	.69, .28	3.6, .160
change order	.67, .26	.06, .50	.53, .36	**29.7, p<.001**
create topic	1.0, 0	.53, .24	1.0, 0	**21.9, p<.001**
delete topic	1.0, 0	1.0, 0	1.0, 0	NA
merge topics	.82, .08	.79, .07	.83, .09	4.0, .130
stop word	1.0, 0	1.0, 0	1.0, 0	NA
split topic	.80, .27	.88, .08	.94, .12	**10.0, .007**

word-level refinements: **add word** ($\chi^2(2) = 10.1, p = .006$) and **change word order** ($\chi^2(2) = 11.5, p = .003$); as shown in Table 7.4. However, there was not a significant difference between the modeling approaches for perceived adherence of the **create topic** ($\chi^2(2) = .9, p = .62$) or **split topic** refinements ($\chi^2(2) = 3.6, p = .17$), even though these differed for computed adherence (Table 7.3). This is perhaps because it is harder for users to discern perfect refinements (all requested words appear in the new topic) from those that are "good enough."

Participants were mixed on whether they observed instability

The computed instability metric shows that the *info-vb* condition was significantly more stable than the other modeling approaches (Table 7.2). However, participants' responses for whether they observed instability had high variability, a pattern that was similar for all modeling approaches (Figure 7.2). While *info-vb* was perceived as the most stable ($M = 2.6$, $\sigma = 2.0$) compared to *info-gibbs* ($M = 3.5$, $\sigma = 2.3$) and *const-gibbs* ($M = 3.8$, $\sigma = 2.3$), these differences were not significant ($\chi^2(2) = 5.6, p = .105$).

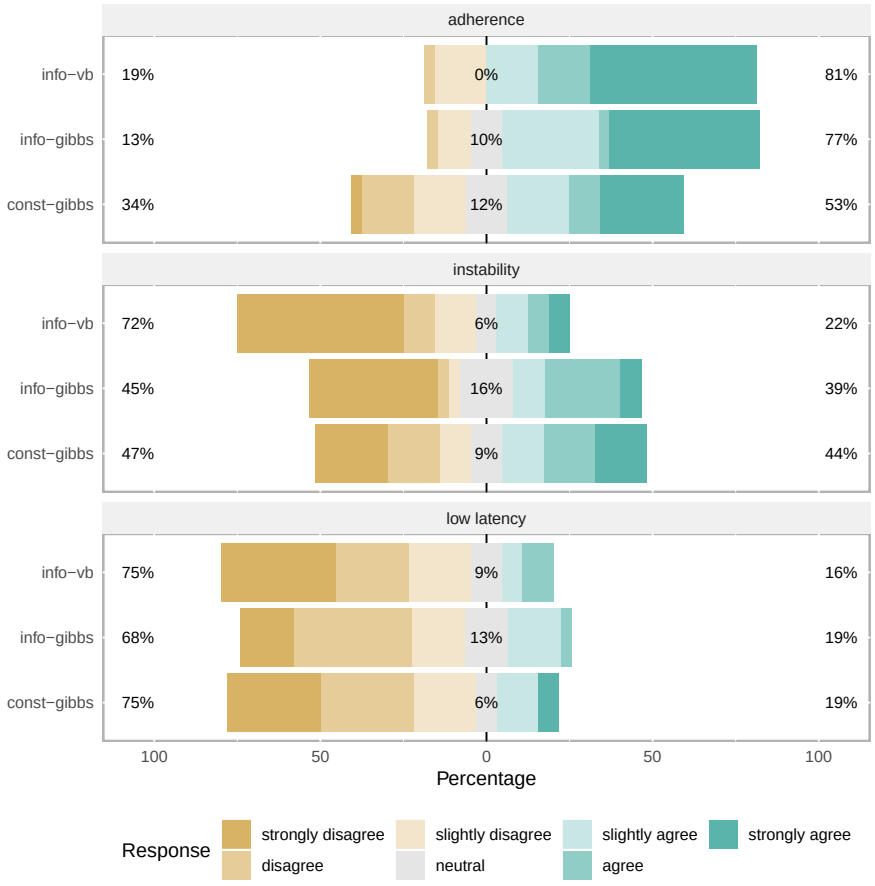


Figure 7.2: Seven-point rating scale responses by modeling approach for perceived adherence, instability, and low latency (quick updates), from "strongly disagree" to "strongly agree." Participants in general thought the systems adhered to their input, but updated slowly. There was high variability for whether participants perceived instability.

Participants thought they improved the models, but coherence scores did not reflect this

We measured model quality and improvement using qualitative—judged by the user—and quantitative—automatic topic coherence—methods (Figure 7.3 and Table 7.2, respectively). Confirming that our initial random model creation was effective; there were no significant differences between modeling approaches for the initial model quality ($\chi^2(2) = 4.1, p = .130$). Automatic coherence declined on average for models, most notably for *info-vb*, confirming previous reports that variational in-

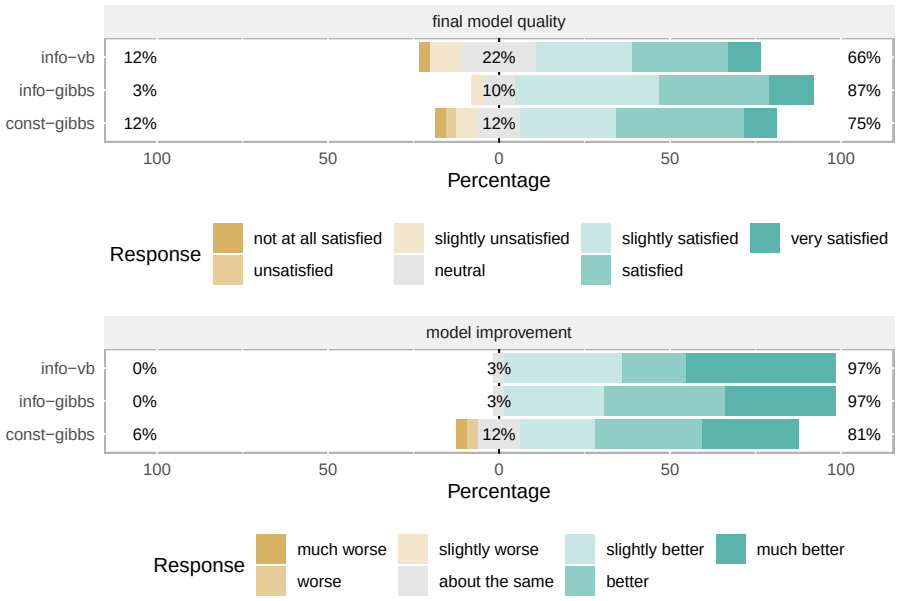


Figure 7.3: Seven-point rating scale responses for subjective model performance: final model satisfaction from "not at all satisfied" to "very satisfied" and model improvement from "much worse" to "much better," reported by modeling approach. Overall participants were satisfied with the final model quality and thought the models had improved from the initial models.

ference can produce less coherent topics than Gibbs sampling (Nguyen et al., 2015). In contrast, participants believed they improved the models: while only 42% of the 95 participants improved the model (as measured by NPMI), 98% thought the final model was better than the initial model (subjective response > 4 out of 7).

Topic coherence is intended to reflect human rating of individual topics (Chang et al., 2009), but our users *reduced* the overall model quality while feeling that they improved it. This discrepancy reflects the limited view of traditional topic coherence metrics: they examine each topic by only top words, and model-wide measures average over all topics; whereas participants typically care about the model as a whole or sometimes prefer a particular subset of topics. Future work should explore robust metrics that better capture how topics model all of the data or put weight on particular topics of interest. Also, topics should be evaluated as both their words and associated documents. Additionally, ideal metrics would be less dependent on the data being modeled.

Table 7.4: Likert scale responses for agreement with statements of the form "the system incorporated the [refinement] operation as I asked it to" for each of the nine refinements. Measurements reported as "mean, σ." Kruskal-Wallis results reported as "$\chi^2(2)$, p." Overall, change word order had low perceived adherence, and there were significant (bold) perceived adherence differences between modeling approaches for add word and change word order; cell shading for these reflects how well participants perceived that modeling approaches to adhere to that refinement compared to the other approaches (darker is better).

	info-gibbs	*const-gibbs*	*info-vb*	Kruskal-Wallis
add word	6.1, 1.5	4.6, 2.5	6.5, 1.4	**9.2, .010**
remove word	6.5, 1.1	5.9, 2.1	6.7, .6	.8, .660
remove doc	6.3, 1.5	6.8, .5	5.6, 2.1	5.0, .080
change order	4.9, 2.2	2.9, 2.5	5.2, 2.4	**11.5, .003**
create topic	6.0, 1.9	6.1, 1.4	6.3, 2.1	.9, .620
delete topic	6.8, .7	6.4, 1.3	6.9, .3	1.5, .470
merge topics	6.7, .8	6.8, .5	6.7, .7	.2, .900
stop word	6.0, 2.0	6.3, 1.4	6.6, .7	.3, .860
split topic	5.6, 2.2	5.9, 2.0	6.9, .3	3.6, .170

Participants thought all the systems were too slow

Objectively, the *info-gibbs* condition had significantly faster updates (Table 7.2). However, users thought all the systems were slow (Figure 7.2), and the perceived latency differences between modeling approaches were not significant ($\chi^2(2) = 1.0, p = .610$). This was likely a combination of participants wanting the systems to be faster and of unrealistic expectations for speed given participants' experiences in the tutorial. For example, P71 (*info-gibbs*) asked, *"is there any way to make it a bit faster?... It would be better if the tutorial wasn't so fast... so you don't have the expectation of speed with this tool."*

7.2.3 User Experience

To understand how variations in adherence, instability, latency, and model performance may affect user experience, participants responded to statements regarding frustration, trust, task ease, and

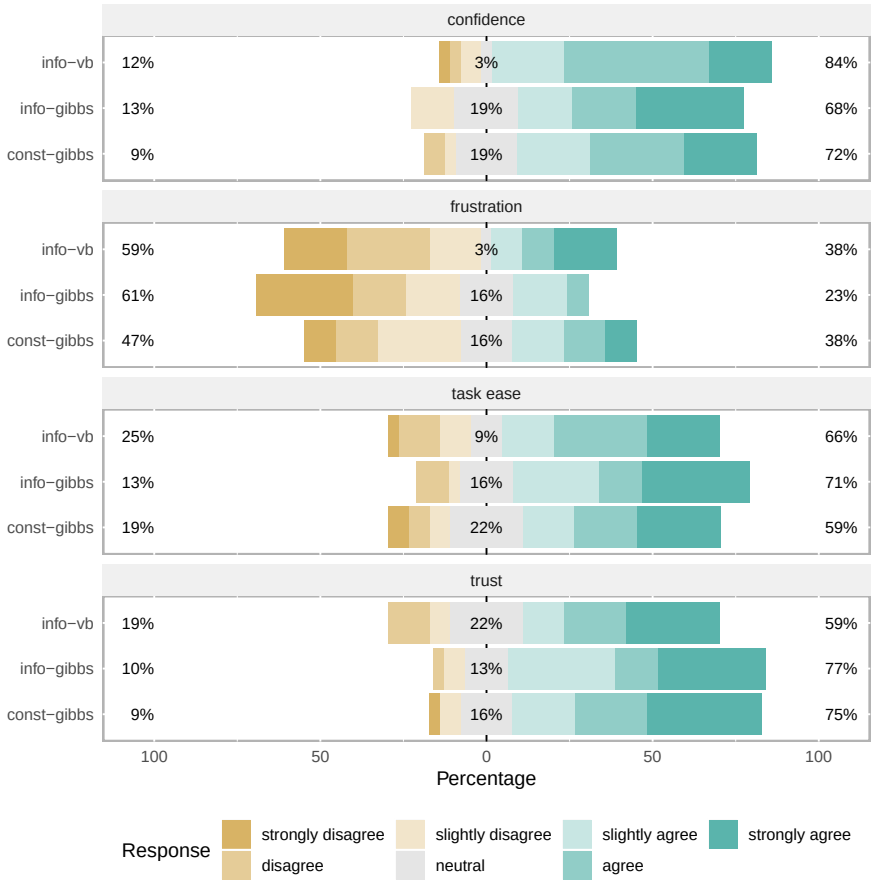


Figure 7.4: Seven-point rating scale responses for four subjective user experience measures from "strongly disagree" to "strongly agree," reported by condition. On average, participants were confident in their input, trusted the system, and thought the task was easy; frustration varied.

confidence (Figure 7.4). Participants were confident, found the task easy, and trusted the tool: mean response for these measures across all modeling approaches was 5.4, 5.0, and 5.3 out of 7, respectively. Participants were neutral regarding frustration, at 3.5 out of 7 for all models, with *info-gibbs* the least frustrating ($M = 2.9$, $\sigma = 1.7$) and *const-gibbs* ($M = 3.8$, $\sigma = 1.8$) and *info-vb* ($M = 3.7$, $\sigma = 2.2$) the most. There were no significant effects of modeling approach on these experience measures, but the open-ended responses provide additional insight into how adherence, instability, and so on affect user experience.

Open-ended responses regarding likes, dislikes, and unexpected behavior

Our coding of open-ended responses (Chapter 7.1.8) resulted in seven *disliked*, seven *liked*, and five *unexpected* codes.

Participants disliked "latency" the most (42 of 95) followed by "lack of control" (21 participants). Ten participants thought the systems were "missing functionality," requesting support for dragging documents between topics or comparing two topics at once. Eight participants thought the tool was "overwhelming", while five said there was "nothing" they did not like. Five disliked "model qualities," such as too many similar topics (P46, *const-gibbs*). Finally, two participants mentioned disliking "instability."

Participants liked that the systems were "useful" for organizing and filtering the documents (40 of 95) and that they were "intuitive" (28). Ten participants liked the "refinements," particularly when they worked as expected, such as P22 (*const-gibbs*), *"the removing of terms was neat and operated as expected"*, while three participants said they liked when the systems "worked as expected." Five participants liked the systems' "design," two participants said they liked "instability," and one liked that the tool was "fast."

Of the measured attributes, participants thought "lack of control," or adherence, (35 of 95) was most unexpected, such as P14 (*info-gibbs*) who said, *"once the change word order did not happen, even though I tried it three times,"* followed by "slowness" (22) and "instability" (12). Twenty participants said "nothing" was unexpected and six mentioned "other" things, like issues with the tutorial.

Instability was the most polarizing attribute. Not all noticed it, but those that did disagreed, confirming our findings in Chapter 6. While 12 of 95 participants said "instability" (as opposed to other attributes) was unexpected, some participants, such as P79 (*info-vb*) said, *"I didn't expect*

the word list to automatically update after adding a new word but I thought that was cool." While other participants said instability was negative, such as, *"I [removed a word] and saw it in a later topic . . . bad ML!"* (P20, *info-gibbs*). Also, two participants said they liked and two participants said they did not like instability.

7.2.4 User Behavior

In addition to measuring participants' subjective responses regarding whether they perceived differences in system attributes and how this affected their experience, we were also interested in understanding how users interact with these systems. On average, each participant used six (σ = 1.4) of the nine operations to make a total of 31.3 (σ = 16.1) changes to their model. In the following, we detail whether user behavior differed given the varied attributes and how users behaved with these systems.

Low adherence may have led participants to stop the task early

Table 7.5 shows the average time spent on the task and number of refinements performed for each condition. The *const-gibbs* modeling approach had significantly slower updates, so we might have expected those participants to spend the longest time on the task, but they did not: participants in the *const-gibbs* condition on average made fewer refinements (M = 27, σ = 13) and spent significantly less time on the task (M = 1859 seconds, σ = 352) than with the other modeling approaches. This might be explained by adherence: the *const-gibbs* modeling approach had significantly lower computed and perceived adherence (Table 7.2 and Figure 7.2), suggesting participants may have abandoned the task if they thought the system was ignoring their input.

Table 7.5: Task time (seconds) and number of refinements per condition. Responses reported as "mean, σ." Kruskal Wallis results reported as "$\chi^2(2)$, p." with significant results in bold.

	info-gibbs	*const-gibbs*	*info-vb*	Kruskal Wallis
Task Time (s)	1970, 356	1859, 352	2071, 352	**6.1, .048**
# Refinements	33, 18	27, 13	37, 18	3.8, .150

Participants used "undo" infrequently, but reverted delete and split topic the most

Participants used "undo" 58 times to revert after applying a refinement. Thirty six of the 95 participants used "undo" an average of 1.6 times ($min = 1$, $max = 5$). Figure 7.5 shows the distribution of refinements that preceded undo normalized by the usage of the refinement. The most frequently undone refinements were **delete topic**, which was undone 10% of the time, and **split topic**, which was undone 8% of the time.

The high frequency of undoing **delete topic** is unexpected. While we had anticipated that participants might undo if operations were not applied as expected, all systems perfectly adhered to the **delete topic** refinement; that is, in these cases, participants were likely exhibiting *experimentation* behavior (Amershi et al., 2010)—perhaps looking for instability to update other areas of the model and then undoing the change if they were not happy with it.

Participants attended to prominent and low quality topics

Figure 7.6 shows which topics were refined by participants based on their location in the topic list (left) and their relative coherence (right). All participants saw a random topic model with random topic ordering, yet participants focused their refinements on the topics at the top of the list ($corr = -0.98$) and on the topics that had the lowest coherence ($corr = -0.94$).

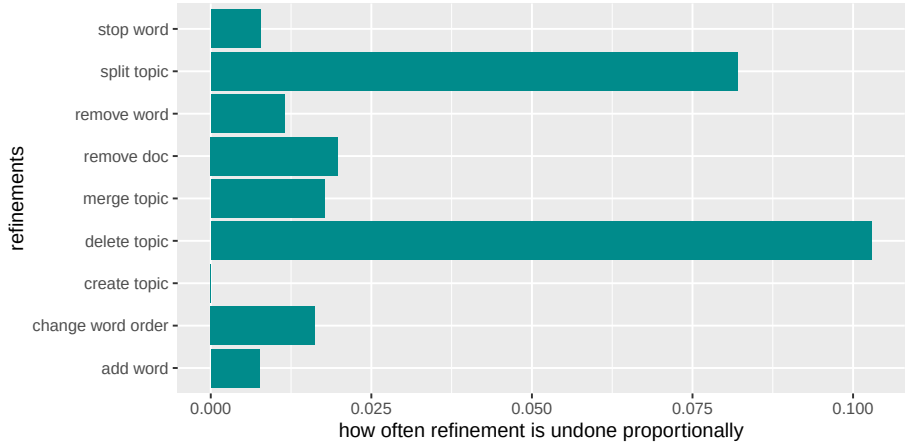


Figure 7.5: Proportion of refinement usage that is followed by undo. **Delete topic** and **split topic** are undone the most often, 10% and 8% of the times they are used, respectively.

Which refinement operations were used and preferred?

Participants refined models at the topic-level more often than at the model-level: **remove document** was used most (8.0 times per participant), followed by **remove word** (7.3), **change word order** (6.5), and **add word** (4.1). Of the topic-level refinements, the two least used (**add word** and **change word order**) were also those that had lower perceived adherence. The most common model-level refinement was **merge topics**, used 2.4 times per participant on average, followed by **add to stop words** (1.4), **delete topic** (0.7), **split topic** (0.6), and **create topic** (0.5).

Participants specified which refinements were most and least useful: **merge topics** was overwhelmingly favored (46 of 95 participants said it was most useful), while **change word order** was unpopular (25 of 95 participants thought it least useful). To better understand why, we look to the open-ended responses.

Participants may have disliked that change word order did not work as expected

Thirteen of the 25 participants who thought **change word order** was the least useful were in the *const-gibbs* condition, likely because this refinement had significantly lower computed and

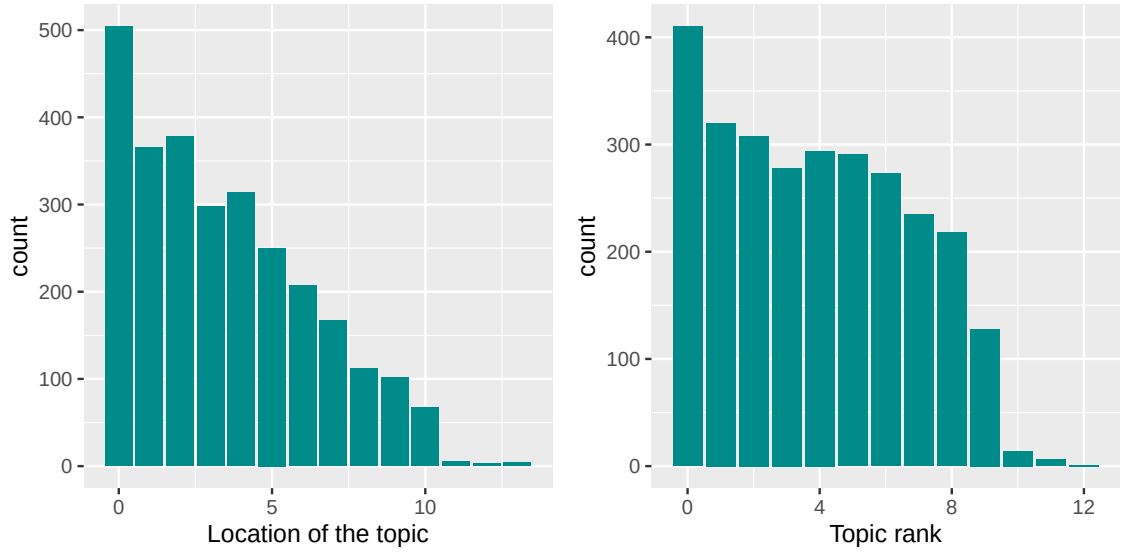


Figure 7.6: Distribution of refined topics by location in the topic list (left) and ranked NPMI quality (right). Participants refined low quality topics and topics at the top of the list.

perceived adherence than in other modeling approaches. Further, many of the participants who did not like **change word order** explained that it "did not work" or had no noticeable effect on the updated model. For example P98 (*const-gibbs*) said, *"for some reason, [change word order] would not work with me."*

Merge topic was a useful refinement for the data and task

Of 95 participants, 49 said that **merge topic** was the most useful refinement, while none thought it least useful. Many of these participants thought **merge topic** was especially useful for the task and model; for example, P82 (*const-gibbs*) said, *"there were multiple topics generated that meant the same thing as another. Putting them together made it more organized."*

7.3 Discussion and Future Work

This chapter explores users' perceptions, experience, and behavior with systems with easy-to-validate controls—in particular, those that provide varied levels of control for both adherence and instability. This section discusses implications and design recommendations for such systems as well as limitations of this study and suggestions for future work.

Users want to be heard

End users want to be in control (Kocielnik et al., 2019; Vaccaro et al., 2018), but what about when systems cannot respect user inputs? While users may expect that their input will be adhered to, as demonstrated by qualitative comments in our study, modeling approaches differ in how user input is incorporated, particularly when it conflicts with the underlying data. For example, suppose a user interacting with a property pricing tool tries to remove all weight from crucial features (e.g., house price and lot size); if the model follows this guidance, prediction quality will decrease. Or, suppose a user tries to add a word to a topic that does not appear in any of the documents; the model simply cannot add this word as it is out of vocabulary.

In our study, refinements that did not work as expected were less popular (e.g., change word order and add word), whereas users preferred refinements that reflected their intent well (e.g., merge topics). Adherence is thus an important quality for developers of human-in-the-loop systems to consider. To account for this, when user input cannot be adhered to, transparent systems could either *explain* why or provide superficial adherence (i.e., treating word-level refinements as modifications of the model *representation*, which do not impact the underlying model).

Users might be willing to share control if they have a helpful partner

Importantly, our study also showed that users think about instability differently than the related concept of adherence. Instability was a lower priority consideration, and not all participants perceived it. For those who did, it was polarizing: some preferred "help" from the system, while others disliked it, particularly when model updates reverted prior changes (e.g., reintroducing previously removed words) or changed topics that users thought were already high quality. Therefore, our recommendation is to (1) better inform users to how models might update and clarifying why models might make other unexpected changes (i.e. faithfully modeling all underlying data); and (2) provide mechanisms for users to *lock* portions of the model which should not be updated and easily revert low quality, unstable updates. These recommendations should promote a healthier human-machine collaboration in which users and models can share control.

Different users, different needs

Users do not have a homogeneous process for interacting with models. As human-in-the-loop systems become more ubiquitous, designers should ensure that models and interfaces are robust to innate user variation. For example, while we did not explore this in our study, different levels of expertise, both with ML and the domain, could impact use: ML experts or those using the system on their own data are more likely to perceive when models update in unexpected ways, and while ML experts might be understanding of this, domain experts (without ML background), are likely to become frustrated. Similarly, personality traits, such as confidence and locus of control, are likely to affect users' desire to be in control, and increase their frustration if systems limit control.

Need for speed—latency and granularity

Machine learning pipelines typically focus on *throughput* as the metric of choice (Gani et al., 2016; Landset et al., 2015). This is indeed important for sating data-hungry models, but humans typically inspect high-level summaries rather than minutiæ. Computational frameworks that can serve intermediate updates quickly would best address users' complaints about "slowness." Further, better management of latency expectations may have reduced frustration in our study; tutorials and initial introductions to ML tools should set expectations regarding latency, as well as other system attributes (e.g., instability and adherence).

7.3.1 Limitations

The study in this chapter used a simple, and fairly short document organization task. Had participants been working with their own data, or working with the systems for longer periods of time, they might have been more invested in model quality, which in turn might have affected their perceptions and experience. Similarly, while our study was aimed at understanding how non-ML experts are affected by unpredictable control in transparent systems, ML experts would likely have differing perceptions and experience.

7.4 Conclusion

In this chapter, we explored users' perceptions, experience, and behavior with easy-to-validate controls that vary in terms of control, particularly how well user input was *adhered to* and whether other changes occurred during model updates (*instability*), as well as how long updates took and model quality. We found that: (1) participants noticed, and in many cases disliked, when their

input was not adhered to, particularly for the easiest-to-validate refinements; (2) participants were polarized by instability, both in whether they noticed it and how they reacted to it: some participants liked it while others did not; (3) participants thought all the systems were slow, but good: participants were satisfied with the final models they generated and thought they showed improvement over their starting points; (4) user experience did not differ between the systems: participants on average were confident in their input, trusted the models to update effectively, and thought the task was easy, but some participants were frustrated, particularly by slow updates.

Chapter 8: Conclusion and Future Work

The goals of this dissertation were to determine effective mechanisms for control and transparency in IML and to provide a better understanding of how these constructs affect end users' experience, perceptions, and behavior, in supervised and unsupervised ML setting. In this chapter, we first briefly summarize the steps in this dissertation research before summarizing the resulting design guidelines and outlining directions for future work.

We studied the interaction between control and transparency for both a simple task and supervised ML technique (interactive text classification) and more subjective tasks with an unsupervised ML technique (interactive topic modeling). More specifically, to fulfill the dissertation goals, we conducted four user studies to (1) examine the interaction between explanations (transparency) and feedback (control) for a simple task and model, (2) determine optimal topic representations for end-user understanding, (3) explore user experience given transparency and control for a subjective task and complex model, and (4) explore user perceptions of control in more detail, specifically whether feedback is applied predictably. We also developed new mechanisms for transparency and control: a new visualization for topics and a new interactive topic modeling system based on users' desired refinement mechanisms, and we evaluated the effectiveness of these mechanisms with end users.

8.1 Designing for the Human in the Loop

In the following sections we summarize design guidelines, which follow from the studies in Chapters 4–7.

Users want to provide feedback and prefer to give detailed guidance

End users want to be in control when interacting with ML systems (Kocielnik et al., 2019; Vaccaro et al., 2018). Similarly, in our studies in Chapter 4, participants felt strongly that the opportunity to provide feedback to improve the model was important. And, our prior work in interactive topic modeling highlighted that ML models cannot simply be provided as "take-it-or-leave-it"; end users have domain expertise that should be incorporated into models, both to improve the model performance as well as user satisfaction (Hu et al., 2014). Moreover, the studies in Chapter 4 provided additional evidence for how different levels of feedback impact user behavior and subjective response, in particular, we confirmed the recommendation of Amershi et al. (2014) that "people naturally want to provide more than just data labels" to ML models.

Explanations and feedback complement each other

While algorithm transparency and interactive machine learning techniques can separately enhance user experience, we focus on their interactions; in particular, we explore the benefits of providing both transparency and control, specifically support for feedback, in ML. Explanations that expose model uncertainty negatively impact users' perceptions of ML models (Lim and Dey, 2011). We hypothesize that such cases are particularly frustrating when users are unable to provide feedback to fix exposed model issues.

For low quality models, explanations were frustrating, precisely because they exposed flaws, in-

cluding *uncertainty* in the model's reasoning (Chapter 4). As expected, providing explanations without support for feedback resulted in decreased users' satisfaction compared to when feedback was provided.

On the other hand, explanations can improve feedback quality (Kulesza et al., 2015). Similarly, asking users to provide feature-level feedback without providing explanations reduced trust compared to when explanations were provided (Chapter 4), suggesting users may not want to provide detailed feedback without understanding why it is needed or how best to help the model.

Users want to be heard, but shared control is needed for a human-machine collaboration

IML systems cannot fully relinquish control to end users so long as system performance is a consideration; this is because IML models must balance respecting user inputs and faithfully modeling the underlying data. Therefore, in some cases, users' input is not perfectly adhered to or could even be ignored. Vaccaro et al. (2018) explored user satisfaction with "difficult-to-validate" controls. For their opaque system, simply providing control mechanisms, whether or not they worked, increased satisfaction. However, what about when systems are transparent, and therefore, controls are "easy-to-validate"?

Our studies in Chapters 6 and 7 exposed how users reacted when transparent models provided varied levels of control. In particular, users reacted to instability differently than the related concept of adherence. Instability was a lower priority consideration, which not all participants perceived. And for those who did, it was polarizing: some participants preferred "help" from the system, while others disliked it. Therefore, systems might (1) better inform users to how models might update and clarify or *explain* why models might make other unexpected changes (i.e., faithfully

modeling all underlying data); (2) provide superficial adherence when desired by users; and (3) provide mechanisms for users to lock portions of the model which should not be updated and to easily revert after low-quality, unstable updates.

These recommendations should promote a healthier human-machine collaboration (or team) in which users and models can share control. We discuss *human-machine teaming* in more detail in Chapter 8.2.3.

Task context matters when choosing an explanation technique

Explanation or transparency techniques should be chosen based on the current task and needs. In Chapter 5, we determined that, in general, the word list allowed users to quickly and adequately understand topics. However, more complex visualizations, such as the topic-in-a-box, exposed users to multi-word expressions that the simpler visualizations obscured. These explanation types serve different purposes and, as such, task goal and user needs should be considered when choosing the appropriate explanation technique.

Need for speed in IML

ML pipelines typically focus on throughput as the metric of choice (Gani et al., 2016; Landset et al., 2015), but humans typically inspect high-level summaries; therefore, computation frameworks that can serve intermediate updates quickly would best address users' concerns about latency (Chapters 6 and 7). If slow updates are an unavoidable system characteristic, tutorials and initial introductions to ML tools should set expectations regarding latency.

Set users' expectations regarding model improvement

Whether from prior experience or general misunderstanding, users may have misconceptions about whether and how much models can improve. In our studies in Chapter 4, many participants expected the model to improve regardless of whether they provided feedback.

Interactive ML designers must ensure that these expectations are managed, such as by clarifying how model feedback is treated or what accuracy the model could reasonably achieve. Or, if feedback is not supported, designers should take special care to ensure users do not think they are in some way providing feedback to the model. We discuss design constructs that may yield feelings of providing feedback in the Chapter 8.2.

8.2 Future Work

In the following sections we outline directions for future work building on the research described in Chapters 4–7.

8.2.1 Further Work on the Interactions of Explanations and Feedback in ML

In Chapter 4, we studied user experience with a simple interactive text classification system, and we varied whether users received simple explanations (i.e., highlighting important words) or could provide feedback (i.e., correcting predictions or specifying important words). Providing explanations without means for feedback reduced user satisfaction, and overall, users expected model improvement (regardless of whether they provided feedback or saw an explanation). Future work should explore the effects of user experience and expectations of improvement given different,

more advanced, explanation types and feedback mechanisms. For example, we hypothesize that "human-like" explanations may increase expectations of improvement, as human-like characteristics in ML systems can cause users to believe systems will act rationally or take responsibility for their actions (Höök, 2000). Expectations may also be affected by when explanations are shown (always or only after erring) or how users attend to them ("dismissing" opposed to "accepting" or "rejecting").

Different feedback mechanisms, when feedback is requested, and task subjectivity or complexity would also likely affect users' desire to be in control and their overall experience. For example, users may prefer choosing when to provide feedback as opposed to the required feedback in our study (Chapter 4), particularly when designing for the stereotypical "lazy user." Similarly, in that study, both the task and feedback mechanisms were simple; rarely were users unsure about the correct classification. However, future work should explore cases where feedback is harder for the user to provide—due to both more complex feedback mechanisms and tasks—to understand how users respond to explanations with and without support for feedback in such cases.

Similarly, personality traits, such as confidence and locus of control, are likely to affect users' desire to be in control, and increase their frustration if systems limit control. Future work should explore how personality traits affect user experience of systems that provide explanations without feedback (as in Chapter 4) and of those that support feedback but have varied adherence to control (as in Chapter 7).

8.2.2 Further Work in Interactive Topic Modeling

Topic coherence and model improvement

In our study in Chapter 7, the majority of participants reduced the model quality (coherence), whereas nearly all felt they had improved the model. This suggests that topic coherence measures are not appropriate for measuring performance in interactive topic modeling. Topic coherence measures how *coherent* a topic's top words are given how often they appear together in a reference corpus. As is common, we use Wikipedia. However, a reference corpus that is more representative of the sample, may be required. For example, a social media reference corpus would better align with the Twitter data we used in our studies (Chapters 6 and 7).

Further, to compute a topic model's coherence, individual topic coherence is averaged over the number of topics. This provides an unfair comparison between models of different sizes, and also does not capture (1) how well the set of topics model the data or (2) how well documents align with topics. However, users tried to improve specifically these model "qualities" during the tweet organization tasks (Chapters 6 and 7). This discrepancy motivate the need for a hybrid metric for performance in interactive topic modeling, which should take into account the following criteria: coherence of topic words, alignment of topic words and associated documents, topic distinctness, and topic coverage of the data set.

Communicating complex model changes in interactive topic modeling

Our exploratory study with an interactive topic modeling system (Chapter 6) found that users had trouble identifying what had changed in the model after an update, a similar concept to model instability. As small changes to topic models can propagate in unexpected ways, it is important

that interactive topic modeling systems effectively communicate these changes and support comparison of the model before and after user refinement. Existing topic model visualizations support efficient word-level comparison of topics using a matrix (Chuang et al., 2012) and topic-level comparison of models using a Sankey Diagram (Malik et al., 2013). However, future work is necessary to determine whether such visualizations can be adapted to effectively visualize complex model changes in interactive topic modeling where topics may be split or merged, words may be added, removed, or reordered, and documents may be added or removed.

8.2.3 Further Work Exploring Human-Machine Teaming

While user interface design guidelines call for users to always be in control (Hoekman, 2007; Shneiderman et al., 2009), full user control is not always feasible in IML, nor is it optimal. For example, in interactive topic modeling, users perform complex, subjective tasks with the aid of IML systems. In this case, systems provide valuable input to assist with a task that the users cannot perform on their own (due to time, complexity, or other requirements). Here, full user control could limit the systems' utility—particularly limiting unpredictable, but useful updates. Users of such systems might consider their interactions as part of a human-machine team, where systems are given some leeway to best support them in complex tasks.

In our studies in Chapters 6 and 7, we observed that participants had varied reactions to adherence and instability (positive, neutral, and negative). In particular, users who trusted the system (or had little confidence in themselves) had more positive reactions to this unpredictability (Chapter 6). One explanation for this variance is the level of shared control the user expects (or desires) from the system. We hypothesize that such a *mindset* affects users' experience with IML systems. For example, users may be more understanding of latency and unpredictability if they consider their

interactions to be working with the system as a team towards some larger goal as opposed to if they feel they should be fully controlling the system. We refer to these as *teaming mindsets*, or to what extent users consider their relationship with an IML system as a human-machine team (e.g., controlling system, equal contributors to the team, being led by system).

We further hypothesize that certain system characteristics lend themselves to human-machine teaming: whether systems and users have complementary strengths, task complexity and subjectivity, and model transparency. We suggest future work exploring whether these or other system characteristics are required for teaming, what if anything affects teaming mindset (e.g., personality traits, expertise, interface elements), what system characteristics make for a good teammate, and how a users' teaming mindsets affect experience with IML systems.

Bibliography

J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Proceedings of Advances in Neural Information Processing Systems*, 2018.

J. W. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open user profiles for adaptive news systems: Help or harm? In *Proceedings of the World Wide Web Conference*, 2007.

N. Aletras, M. Stevenson, and R. Court. Labelling topics using unsupervised graph-based methods. In *Proceedings of the Association for Computational Linguistics*, 2014.

E. Alexander and M. Gleicher. Assessing topic representations for gist-forming. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2016.

D. Alvarez-Melis and T. S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Proceedings of Advances in Neural Information Processing Systems*, 2018.

S. Amershi, J. Fogarty, A. Kapoor, and D. Tan. Examining multiple potential models in end-user interactive concept learning. In *International Conference on Human Factors in Computing Systems*, 2010.

S. Amershi, J. Fogarty, and D. Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *International Conference on Human Factors in Computing Systems*, 2012.

S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning, 2014.

S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz. Guidelines for human-AI interaction. In *International Conference on Human Factors in Computing Systems*, 2019.

D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *Proceedings of the International Conference of Machine Learning*, 2009.

A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On Smoothing and inference for topic models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2009.

P. Awasthi, M. F. Balcan, and K. Voevodski. Local algorithms for interactive clustering. *Journal of Machine Learning Research*, 2017.

A. Bakharia, P. Bruza, J. Watters, B. Narayan, and L. Sitbon. Interactive topic modeling for aiding qualitative content analysis. In *Proceedings of the ACM Conference on Human Information Interaction and Retrieval*, 2016.

M.-F. Balcan and A. Blum. Clustering with interactive feedback. In *International Conference on Algorithmic Learning Theory*, 2008.

G. Bansal, B. Nushi, E. Kamar, D. S. Weld, W. S. Lasecki, and E. Horvitz. Updates in human-AI teams: Understanding and addressing the performance/compatibility tradeoff. *Association for the Advancement of Artificial Intelligence*, 2019.

L. Barth, S. G. Kobourov, and S. Pupyrev. Experimental comparison of semantic word clouds. In *International Symposium on Experimental Algorithms*, 2014.

M. Belford, B. Mac Namee, and D. Greene. Stability of topic modeling via matrix factorization. *Expert Systems with Applications*, 2018.

M. Bilgic and R. J. Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at IUI*, 2005.

O. Biran and K. McKeown. Human-centric justification of machine learning predictions. In *International Joint Conference on Artificial Intelligence*, 2017.

D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55:77–84, 4 2012.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3(1):993–1022, 2003.

G. Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*, 2009.

J. Boyd-Graber, D. M. Blei, and X. Zhu. A topic model for word sense disambiguation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2007.

J. Boyd-Graber, D. Mimno, and D. Newman. Care and feeding of topic models: Problems, diagnostics, and improvements. *Handbook of mixed membership models and their applications*, pages 225 – 254, 2014.

V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 1 2006.

A. Bunt, J. McGrenere, and C. Conati. Understanding the utility of rationale in a mixed-initiative system for GUI customization. In *International Conference on User Modeling*, 2007.

C. J. Cai, J. Jongejan, and J. Holbrook. The effects of example-based explanations in a machine learning interface. In *International Conference on Intelligent User Interfaces*, 2019.

O. M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom. E-SNLI: Natural language inference with natural language explanations. In *Proceedings of Advances in Neural Information Processing Systems*, 2018.

R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Knowledge Discovery and Data Mining*, 2015.

A. Chandrasekaran, V. Prabhu, D. Yadav, P. Chattopadhyay, and D. Parikh. Do explanations make VQA models more predictable to a human? *arXiv preprint arXiv:1810.12366*, 2018.

A. J.-B. Chaney and D. M. Blei. Visualizing topic models. In *Proceedings of the International Conference on Weblogs and Social Media*, 2012.

J. Chang, S. Gerrish, C. Wang, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2009.

E. Charniak. A maximum-entropy-inspired parser. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2000.

J. Choo, C. Lee, C. K. Reddy, and H. Park. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.

J. Chuang, C. D. Manning, and J. Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012.

J. Chuang, Y. Hu, A. Jin, J. D. Wilkerson, D. A. McFarland, C. D. Manning, and J. Heer. Document exploration with topic modeling: Designing interactive visualizations to support effective analysis workflows. In *Proceedings of Advances in Neural Information Processing Systems*, 2013.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 1994.

H. Cramer, V. Evers, S. Ramlal, M. Van Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 2008.

A. Culotta, T. Kristjansson, A. McCallum, and P. Viola. Corrective feedback and persistent learning for information extraction. *Artificial Intelligence*, 2006.

J. Dodge, Q. Vera Liao, Y. Zhang, R. K. Bellamy, and C. Dugan. Explaining models: An empirical study of how explanations impact fairness judgment. In *International Conference on Intelligent User Interfaces*, 2019.

F. Du, C. Plaisant, N. Spring, and B. Shneiderman. Finding similar people to guide life choices: Challenge, design, and evaluation. In *International Conference on Human Factors in Computing Systems*, 2017.

U. Ehsan, P. Tambwekar, L. Chan, B. Harrison, and M. O. Riedl. Automated rationale generation: A technique for explainable AI and its effects on human perceptions. In *International Conference on Intelligent User Interfaces*, 2019.

J. Eisenstein, D. H. Chau, A. Kittur, and E. Xing. TopicViz: Interactive topic exploration in document collections. In *International Conference on Human Factors in Computing Systems*, 2012.

J. A. Fails and D. R. Olsen. Interactive machine learning. In *International Conference on Intelligent User Interfaces*, 2003.

A. Fan, F. Doshi-Velez, and L. Miratrix. Prior matters: simple and general methods for evaluating and improving topic quality in topic modeling. *arXiv preprint arXiv:1701.03227*, 2017.

S. Feng and J. Boyd-Graber. What can AI do for me? Evaluating machine learning interpretations in cooperative play. In *International Conference on Intelligent User Interfaces*, 2019.

R. Fiebrink, D. Trueman, and P. R. Cook. A metainstrument for interactive, on-the-fly machine learning. In *Proceedings of New Interfaces for Musical Expression (NIME)*, 2009.

J. Fox. Effect displays for generalized linear models. *Sociological Methodology*, 1987.

J. Fox. Effect displays in R for generalized linear models. *Journal of Statistical Software*, 8(15): 1–27, 2003.

T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

K. Z. Gajos, K. Everitt, D. S. Tan, M. Czerwinski, and D. S. Weld. Predictability and accuracy in adaptive user interfaces. In *International Conference on Human Factors in Computing Systems*, 2008.

A. Gani, A. Siddiqa, S. Shamshirband, and F. Hanum. A survey on indexing techniques for big data: taxonomy and performance evaluation. *Knowledge and Information Systems*, 2016.

M. J. Gardner, J. Lutes, J. Lund, J. Hansen, D. Walker, E. Ringger, and K. Seppi. The Topic Browser: An interactive tool for browsing topic models. In *Proceedings of the Workshop on Challenges of Data Visualization, held in conjunction with the Annual Conference on Neural Information Processing Systems*, 2010.

C. Geigle. Inference methods for latent dirichlet allocation. 2016.

D. Gkatzia, O. Lemon, and V. Rieser. Natural language generation enhances human decision-making with uncertain information. In *Proceedings of the Association for Computational Linguistics*, 2016.

D. Głowacka, T. Ruotsalo, K. Konyushkova, K. Athukorala, S. Kaski, and G. Jacucci. Directing exploratory search: Reinforcement learning from user interactions with keywords. In *International Conference on Intelligent User Interfaces*, 2013.

B. Goodman and S. Flaxman. European union regulations on algorithmic decision making and a "right to explanation". *AI Magazine*, 2017.

D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the International Conference of Machine Learning*, 2006.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl 1):5228–5235, 2004.

D. Gunning. Explainable Artificial Intelligence (XAI), 2016.

J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Conference on Computer Supported Cooperative Work and Social Computing*, 2000.

R. Hoekman. *Designing the Obvious: A Common Sense Approach to Web Application Design*. New Riders Publishing, 2007.

M. Hoffman, D. M. Blei, and F. Bach. Online learning for latent Dirichlet allocation. In *Proceedings of Advances in Neural Information Processing Systems*, 2010.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

L. E. Holmqvist. Intelligence on tap: artificial intelligence as a new design material. *Interactions*, pages 28–33, 2017.

K. Höök. Steps to take before intelligent user interfaces become real. *Interacting With Computers*, 12(4):409–426, 2000.

E. Hoque and G. Carenini. ConVisIT: Interactive topic modeling for exploring asynchronous online conversations. In *International Conference on Intelligent User Interfaces*, 2015.

E. Horvitz. Principles of mixed-initiative user interfaces. In *International Conference on Human Factors in Computing Systems*, 1999.

T. Hospedales, S. Gong, and T. Xiang. A Markov clustering topic model for mining behaviour in video. In *International Conference on Computer Vision*, 2009.

Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith. Interactive Topic Modeling. *Machine Learning*, 95(3):423–469, 6 2014.

A. Huang. Similarity measures for text document clustering. In *New Zealand Computer Science Research Student Conference*, 2008.

A. Kangasrääsiö, D. Głowacka, and S. Kaski. Improving controllability and predictability of interactive recommendation interfaces for exploratory search. In *International Conference on Intelligent User Interfaces*, 2015.

L. F. Klein, J. Eisenstein, and I. Sun. Exploratory thematic analysis for digitized archival collections. *Digital Scholarship in the Humanities*, 2015.

W. B. Knox and P. Stone. Reinforcement learning from human reward: Discounting in episodic tasks. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, 2012.

R. Kocielnik, S. Amershi, and P. N. Bennett. Will you accept an imperfect AI? Exploring designs for adjusting end-user expectations of AI systems. In *International Conference on Human Factors in Computing Systems*, 2019.

T. Kulesza, S. Stumpf, M. Burnett, W. K. Wong, Y. Riche, T. Moore, I. Oberst, A. Shinsel, and K. McIntosh. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*, 2010.

T. Kulesza, S. Stumpf, M. Burnett, and I. Kwan. Tell me more? The effects of mental nodel soundness on personalizing an intelligent agent. In *International Conference on Human Factors in Computing Systems*, 2012.

T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W. K. Wong. Too Much, too little, or just right? Ways explanations impact end users' mental models. In *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing*, 2013.

T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *International Conference on Intelligent User Interfaces*, 2015.

V. Kumar, A. Smith, L. Findlater, K. Seppi, and J. Boyd-Graber. Why didn't you listen to me? comparing user control of human-in-the-loop topic models. In *Proceedings of the Association for Computational Linguistics*, 2019.

I. Lage, A. S. Ross, B. Kim, S. J. Gershman, and F. Doshi-Velez. Human-in-the-loop interpretability prior. In *Proceedings of Advances in Neural Information Processing Systems*, 2018.

H. Lakkaraju, R. Caruana, E. Kamar, and J. Leskovec. Faithful and customizable explanations of black box models. In *AAAI/ACM Conference on AI, Ethics, and Society (AEIS)*, 2019.

S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin. A survey of open source tools for machine learning with big data in the Hadoop ecosystem. *Journal of Big Data*, 2015.

K. Lang. NewsWeeder: Learning to filter Netnews. In *Machine Learning Proceedings*. 1995.

H. Larochelle and S. Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, 2012.

J. H. Lau, K. Grieser, D. Newman, and T. Baldwin. Automatic labelling of topic models. In *Proceedings of the Association for Computational Linguistics*, 2011.

J. H. Lau, D. Newman, and T. Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, 2014.

H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. iVisClustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31:1155–1164, 2012.

T. Y. Lee, A. Smith, K. Seppi, N. Elmqvist, J. Boyd-Graber, and L. Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human Computer Studies*, 105:28–42, 2017.

T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions. In *Proceedings of Empirical Methods in Natural Language Processing*, 2016.

D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of European Conference of Machine Learning*, 1998.

B. Lim, A. Dey, and D. Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *International Conference on Human Factors in Computing Systems*, 2009.

B. Y. Lim and A. K. Dey. Investigating intelligibility for uncertain context-aware applications. In *Proceedings of the international conference on Ubiquitous computing*, 2011.

C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2009.

J. Lund, C. Cook, K. Seppi, and J. Boyd-Graber. Tandem anchoring: a multiword anchor approach for interactive topic modeling. In *Proceedings of the Association for Computational Linguistics*, 2017.

S. Malik, A. Smith, T. Hawes, P. Papadatos, J. Li, C. Dunne, and B. Shneiderman. TopicFlow: Visualizing topic alignment of Twitter data over time. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, 2013.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

G. A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11): 39–41, 1995.

D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of Empirical Methods in Natural Language Processing*, 2011.

V. M.K and K. K. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 2016.

A. Mueller. Word cloud, 2012.

C. Musialek, P. Resnik, and A. S. Stavisky. Using text analytic techniques to create efficiencies in analyzing qualitative data: A comparison between traditional content analysis and a topic modeling approach. In *American Association for Public Opinion Research*, 2016.

M. Narayanan, E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*, 2018.

D. Newman, J. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2010a.

D. Newman, Y. Noh, E. Talley, S. Karimi, and T. Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries*, 2010b.

A. Y. Ng and M. I. Jordan. On discriminative vs. Generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, 2002.

D. Q. Nguyen, K. Sirts, and M. Johnson. Improving topic coherence with latent feature word representations in MAP estimation for topic modeling. In *Proceedings of the Australasian Language Technology Association Workshop*, 2015.

V.-A. Nguyen, Y. Hu, J. Boyd-Graber, and P. Resnik. Argviz: Interactive visualization of topic dynamics in multi-party conversations. In *Proceedings of the NAACL HLT Demonstration Session*, 2013.

D. A. Norman. How might people interact with agents. *Communications of the ACM*, 37(7): 68–71, 1994.

D. H. Park, L. A. Hendricks, Z. Akata, B. Schiele, T. Darrell, and M. Rohrbach. Attentive explanations: Justifying decisions and pointing to the evidence. *arXiv preprint arXiv:1612.04757*, 2016.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Q. Pleplé. Interactive topic modeling. Master's thesis, UC San Diego, 2013.

F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Vaughan, and H. Wallach. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*, 2018.

P. Pu and L. Chen. Trust building with explanation interfaces. In *International Conference on Intelligent User Interfaces*, 2006.

H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on both features and instances. *Journal of Machine Learning Research*, 2006.

D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *International Conference on Weblogs and Social Media*, 2010.

A. M. Rashid, K. Ling, R. D. Tassone, P. Resnick, R. Kraut, and J. Riedl. Motivating participation by displaying the value of contribution. In *International Conference on Human Factors in Computing Systems*, 2006.

A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment*, 2017.

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you? Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining*, 2016.

P. Riehmann, M. Hanfler, and B. Froehlich. Interactive Sankey diagrams. In *International Symposium on Information Visualization*, 2005.

E. M. Rodrigues, N. Milic-Frayling, M. Smith, B. Shneiderman, and D. Hansen. Group-in-a-box layout for multi-faceted analysis of communities. In *International Conference on Privacy, Security, Risk and Trust and International Conference on Social Computing*, 2011.

S. L. Rosenthal and A. K. Dey. Towards maximizing the accuracy of human-labeled sensor data. In *International Conference on Intelligent User Interfaces*, 2010.

Q. Roy, F. Zhang, and D. Vogel. Automation accuracy is good, but high controllability may be better. In *International Conference on Human Factors in Computing Systems*, 2019.

A. M. Saeidi, J. Hage, R. Khadka, and S. Jansen. ITMViz: Interactive topic modeling for source code analysis. In *International Conference on Program Comprehension*, 2015.

E. Sandhaus. *The New York Times Annotated Corpus*. Linguistic Data Consortium, Philadelphia, 2008.

W. Saunders, A. Stuhlmüller, G. Sastry, and O. Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2018.

P. Schmidt and F. Biessmann. Quantifying interpretability and trust in machine learning systems. *arXiv preprint arXiv:1901.08558*, 2019.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision*, 2017.

B. Settles. Active learning literature survey. *Machine Learning*, 15(2):201–221, 2010.

B. Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of Empirical Methods in Natural Language Processing*, 2011.

B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages*, 1996.

B. Shneiderman, C. Plaisant, M. Cohen, and S. Jacobs. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Fifth edition, 2009.

Z. Si and S. C. Zhu. Learning and-or templates for object recognition and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

R. S. Siegler et al. Microgenetic studies of self-explanation. *Microdevelopment: Transition processes in development and learning*, pages 31–58, 2002.

C. Sievert and K. Shirley. LDAvis: A method for visualizing and interpreting topics. *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, 2014.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, and D. Hassabis. Mastering the game of Go without human knowledge. *Nature*, 2017.

K. Simonyan. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations*, 2013.

R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *International Conference on Human Factors in Computing Systems*, 2002.

A. Smith, J. Chuang, Y. Hu, J. Boyd-Graber, and L. Findlater. Concurrent Visualization of Relationships between Words and Topics in Topic Models. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, 2014a.

A. Smith, T. Hawes, and M. Myers. Hi´erarchie: Interactive Visualization for Hierarchical Topic Models. In *Proceedings of the Workshop on Interactive Language Learning*, 2014b.

A. Smith, T. Yeon Lee, F. Poursabzi-Sangdeh, J. Boyd-Graber, N. Elmqvist, and L. Findlater. Evaluating visual representations for topic understanding and their effects on manually generated topic labels. *Transactions of the Association for Computational Linguistics*, 5:1–15, 2017.

A. Smith, V. Kumar, J. Boyd-Graber, K. Seppi, and L. Findlater. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *International Conference on Intelligent User Interfaces*, 2018.

J. Soo Yi, R. Melton, J. Stasko, and J. A. Jacko. Dust and magnet: Multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, 2005.

K. Stowers, N. Kasdaglis, M. Rupp, J. Chen, D. Barber, and M. Barnes. Insights into human-agent teaming: Intelligent agent transparency and uncertainty. In *Advances in Intelligent Systems and Computing*, 2017.

S. Stumpf. Explanations considered harmful? User interactions with machine learning systems. In *ACM SIGCHI Workshop on Human-Centered Machine Learning*, 2016.

J. W. Tukey. *Exploratory data analysis*. Reading, Mass., 1977.

K. Vaccaro, D. Huang, M. Eslami, C. Sandvig, K. Hamilton, and K. Karahalios. The illusion of control: Placebo effects of control settings. In *International Conference on Human Factors in Computing Systems*, 2018.

J. W. Vaughan. Making better use of the crowd: How crowdsourcing can advance machine learning research. *Journal of Machine Learning Research*, 2018.

E. Wall, S. Ghorashi, and G. Ramos. Using expert patterns in assisted interactive machine learning: A study in machine teaching. In *IFIP Conference on Human-Computer Interaction*, 2019.

H. M. Wallach, D. Mimno, and A. Mccallum. Rethinking LDA : Why priors matter. In *Proceedings of Advances in Neural Information Processing Systems*, 2009a.

H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the International Conference of Machine Learning*, 2009b.

J. Wang, C. Zhao, J. Xiang, and K. Uchino. Interactive topic model with enhanced interpretability. In *Proceedings of the Workshop on Explainable Smart Systems*, 2019.

J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The Aligned Rank Transform for nonparametric factorial analyses using only ANOVA procedures. In *International Conference on Human Factors in Computing Systems*, 2011.

T. Wu, D. S. Weld, and J. Heer. Local decision pitfalls in interactive machine learning: An investigation into feature selection in sentiment analysis. *ACM Transactions on Computer-Human Interaction*, 2019.

Y. Yang, D. Downey, J. Boyd-graber, and J. B. Graber. Efficient methods for incorporating knowledge into topic models. In *Proceedings of Empirical Methods in Natural Language Processing*, 2015.

L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *Knowledge Discovery and Data Mining*, 2009.

M. Yin, J. W. Vaughan, and H. Wallach. Understanding the effect of accuracy on trust in machine learning models. In *International Conference on Human Factors in Computing Systems*, 2019.

K. Zhai, J. Boyd-Graber, N. Asadi, and M. Alkhouja. Mr. LDA: A flexible large scale topic modeling package using variational inference in MapReduce. In *Proceedings of the World Wide Web Conference*, 2012.