Design and Empirical Evaluation of Interactive and Interpretable Machine Learning

by

Forough Poursabzi-Sangdeh

B.S., University of Tehran, 2012

M.S., University of Colorado Boulder, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2018

This thesis entitled:
Design and Empirical Evaluation of Interactive and Interpretable Machine Learning
written by Forough Poursabzi-Sangdeh
has been approved for the Department of Computer Science

Prof. Michael J. Paul

Prof. Jordan Boyd-Graber

Prof. Leah Findlater

Date ____________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Poursabzi-Sangdeh, Forough (Ph.D., Computer Science)

Design and Empirical Evaluation of Interactive and Interpretable Machine Learning

Thesis directed by Prof. Jordan Boyd-Graber (2013-2017) and Prof. Michael J. Paul (2017-2018)

Machine learning is ubiquitous in making predictions that affect people's decisions. While most of the research in machine learning focuses on improving the performance of the models on held-out data sets, this is not enough to convince end-users that these models are trustworthy or reliable in the wild. To address this problem, a new line of research has emerged that focuses on developing interpretable machine learning methods and helping end-users make informed decisions.

Despite the growing body of research in developing interpretable models, there is still no consensus on the definition and quantification of interpretability. We argue that to understand interpretability, we need to bring humans in the loop and run human-subject experiments to understand the effect of interpretability on human behavior. This thesis approaches the problem of interpretability from an interdisciplinary perspective which builds on decades of research in psychology, cognitive science, and social science to understand human behavior and trust. Through controlled user experiments, we manipulate various design factors in supervised models that are commonly thought to make models more or less interpretable and measure their influence on user behavior, performance, and trust. Additionally, we develop interpretable and interactive machine learning based systems that exploit unsupervised machine learning models to bring humans in the loop and help them in completing real-world tasks. By bringing humans and machines together, we can empower humans to understand and organize large document collections better and faster. Our findings and insights from these experiments can guide the development of next-generation machine learning models that can be used effectively and trusted by humans.

Dedication

To

Maman, Shahin Heidarpour-Tabrizi

and

Baba, Ali Poursabzi-Sangdeh

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Jordan Boyd-Graber. Jordan's invaluable support made my doctorate studies a delightful experience. I appreciate that he made sure I enjoyed what I was doing and provided support, academically and emotionally, all these years. I am grateful for everything I have learned from him.

I am further thankful to the committee/advisory members. Prof. Leah Findlater, Prof. James H. Martin, Prof. Martha Palmer, Prof. Michael J. Paul, and Prof. Chenhao Tan for their help and feedback throughout my studies and research work.

I was extremely fortunate to collaborate with the most brilliant and amazing group of people during my graduate studies. Dr. Niklas Elmqvist, Dr. Daniel Goldstein, Dr. Jake Hofman, Dr. Pallika Kanani, Dr. Kevin Seppi, Dr. Jennifer Wortman Vaughan, Dr. Hanna Wallach, thank you for your mentorship. I cannot imagine where I would be without your advice, guidance, and support. Tak Yeon Lee, You Lou, Thang Nguyen, and Alison Smith, thanks for all the extra fun you brought to my Ph.D. studies by amazing collaborations.

I would like to thank my previous advisors, Prof. Ananth Kalyanaraman and Prof. Debra Goldberg, whose insights motivated me at the very beginning of this journey.

Former/current members of our lab at CU made my time in the lab more fun (and of course, they were always there for discussions and my endless pilot studies!). Alvin Grissom, Fenfei Guo, Shudong Hao, Pedro Rodriguez, and Davis Yoshida, Samantha Molnar, Allison Morgan, and Nikki Sanderson, I cannot thank you enough for always being there for great discussions, snacks, and tea!

I am grateful to my friends Al, Amir, Arash, Azadeh, Farhad, Ghazaleh, Hamid, Homa, Hooman, Liam, Mahdi, Mahnaz, Mahshab, Masoud, Mohammad, Neda, Paria, Reza, Reihaneh, Romik, Saman, Sanaz, Sepideh, Sina, and Sorayya for all the wonderful memories we made in Boulder. I will miss you! I would like to extend special thanks to Niloo, Maryam, Goli, Zeinab, Ghazal, and Sepideh for always listening to me and supporting me from miles away.

My parents have made far too many sacrifices for my education and well-being. They always challenged me with math problems since I was a little kid and always inspired me to follow my dreams, even in the hardest times, even when that meant not being able to see each other for the next five or six years. Words cannot express how much I have learned from them and how grateful I am for them. I additionally thank my sister, Farzaneh, and my brother-in-law, Hadi, for their endless love, encouragement, and support. Last but not least, I want to especially thank my partner, Hadi, for his constant love and support. The best thing about finishing this dissertation is that we are going to face what's next together.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

Chapter 1

Introduction

Machine learning is ubiquitous and continuous advancements improve the performance of machine learning techniques in social science, healthcare, and criminal justice. Machine learning models are usually evaluated and tested based on their predictive power on a held-out data set before being deployed in the real world. However, good performance on a held-out data set is seldom sufficient for practitioners to trust these models. As a result, hesitation in deploying these models is common among stakeholders, especially in critical areas such as healthcare and criminal justice.

This skepticism is caused by a gap between model developers and end users. While machine learning has been extremely successful in automatizing many tasks and decision-making procedures, human involvement is still necessary for several aspects: humans should **annotate** and **collect** the data that is required for models, **develop, tune**, and **debug** models, and **evaluate** models and **trust** them. Despite the inevitable involvement of humans, their behavior is rarely studied and their goals are rarely considered. Additionally, machine learning models are often treated as black boxes, which makes it hard for humans to get involved and interact with them.

Part of the reason for the gap between model developers and end users is the difference in both expectations and areas of expertise. While the developers and designers of machine learning models are usually experts in computational and statistical methods, the end users are usually **domain** experts (e.g., doctors, lawyers, social scientists). Domain experts need to trust these models before deploying them and since human trust and goals are not purely computational, they are usually not a direct factor in model design and evaluation pipelines.

This thesis bridges the gap between humans and machine learning models and empirically evaluates model-based systems that humans can interact with. To achieve this goal, we take an interdisciplinary approach. We build on decades of research from psychology and cognitive science to design and run experiments and understand human behavior when they are exposed to machine learning models. Furthermore, we design and evaluate frameworks that bring machine learning models and humans together to empower end users to complete tasks and achieve their goals better and faster.

This chapter provides a summary of key innovations, approaches, and results presented in this thesis. We start by motivating human-in-the-loop design for interactive and interpretable machine learning.

1.1 Motivation

Machine learning is commonly used to make predictions that affect people and their decisions: recommending the next movie to watch in online streaming applications such as Netflix (Pilászy and Tikk, 2009; Christakou et al., 2007), detecting spam emails (Rajan et al., 2006; Jindal and Liu, 2007), and predicting the risk of a specific disease for a patient in healthcare (Cooper et al., 1997; Caruana et al., 2015) and predicting the risk of lending in finance (Wang et al., 2005; Yu et al., 2008). Although there have been huge advances to improve the performance of machine learning models, these models are often treated as black boxes. Interpreting them and interacting with them remains a persistent problem.

While computer scientists might not see this as a problem, professionals in other communities who are end users of machine learning models do. For example, models that are hard to interpret are considered “risky” to be deployed for making decisions about the hospitalization of patients by healthcare professionals (Caruana et al., 2015), and new regulations in the European Union require that algorithmic decisions should be **explainable** to the affected individuals (Goodman and Flaxman, 2016).

A related problematic aspect of black-box machine learning appears in the scenarios where end users are interested in **understanding** and **explaining** some phenomena. For example, social scientists commonly use machine learning to model and understand people’s actions and interactions (Iyyer et al., 2014; Nguyen et al., 2014a; Guo et al., 2015). Contrary to computer scientists who are more interested in **prediction** (and thus evaluate models based on their predictive performance), social scientists are usually interested in

characterizing the data and finding plausible **explanations** from the data (Hofman et al., 2017). Similarly, professionals in healthcare usually want to understand **how** and **why** a machine learning prediction has been made so that they can make informed decisions (Cooper et al., 1997).

These differences in the goals of computer scientists and end users are constantly overlooked. Developers of machine learning models commonly ignore the goals of end users and evaluate models solely based on their predictive performance on a held-out data set. However, with the ubiquity of machine learning in critical areas, good performance on a held-out data set is seldom enough to convince professionals to trust and deploy these models.

To address this problem and fill the gap between computer scientists and end users, a new line of research has emerged that focuses on developing **interpretable** machine learning models. The Fairness, Accountability, and Transparency in Machine Learning (FATML) community,[1] brings together researchers and practitioners concerned with fairness, accountability, and transparency in machine learning. A subsection of the work in this community focuses on developing interpretable machine learning models (Lakkaraju et al., 2017; Bastani et al., 2017). Interpretability in machine learning is inspired by the need for "simplicity" and "explainability" for gaining end users' **trust**. The goal is to design models that humans can understand, debug, interact with, and make informed decision with.

Despite the growing body of research on interpretability, there is still no consensus on how to define or measure interpretability (Lipton, 2016; Doshi-Velez and Kim, 2017). In this thesis, we argue that bringing humans in the loop is necessary to define and measure interpretability, gain users' trust and convince users that a machine learning model is reliable in the wild. As a result, we need to take a task-driven and human-in-the-loop approach and examine model interpretability, usability, and trustworthiness based on **humans'** abilities and behavior.

1.2 Thesis Goals

At a high level, the goal of this thesis is to design and empirically evaluate interpretable models and systems that help humans complete real-world tasks better and faster and enable humans to effectively

[1] http://www.fatml.org/

interact with them. This goal requires an understanding of how humans perceive and react to models and how model "interpretability" affects their behavior. Therefore, we take a two-fold approach: (1) empirically studying humans when they are using machine learning models to complete tasks and measuring the effect of various model design factors on their behavior; (2) developing interpretable and interactive machine learning based systems that bring humans in the loop and helps them complete tasks better and faster.

More specifically, our goal is to answer the following questions:

(1) How do the factors that are generally thought to make supervised machine learning models more or less interpretable affect users' behavior such as their trust in models and their level of understanding of how models work?

  (a) How can we isolate these factors and study their effect on users' abilities and behavior?

  (b) Can these experiments provide insights on how to design models that users can trust?

(2) Can we use interpretable, interactive, and unsupervised machine learning to help users complete a task better and faster?

  (a) Do interpretability and interactivity lead to better performance faster?

  (b) Do empirical experiments with humans in the loop provide insights on efficiency of different approaches in different scenarios?

1.3 Thesis Approach and Overview

This thesis combines findings from human-computer interaction, psychology, cognitive science, and social science research to design human-in-the-loop methods along with human-subject experiments to evaluate these methods. Additionally, it provides insights in several aspects of machine learning with humans in the loop. Chapter 3 studies how users behave when they are exposed to a supervised machine learning model and how their abilities and behavior are affected with different factors in the model design. Chapter 4 and Chapter 5 present interactive frameworks that exploit unsupervised machine learning models to enable users to complete tasks with the help from machine learning and information retrieval methods.

Interacting with machine learning models empowers humans, and machine learning research can benefit from collaboration between humans and machines. Designing models that end users can trust and use as intended requires an understanding of users' abilities and behavior. The experiments and findings in this thesis should encourage machine learning researchers to collaborate with professionals from other fields to understand human behavior better, which will, in turn, lead to designing and evaluating models that humans can use effectively.

We now provide a description of approaches and contributions of the work in this thesis.

1.3.1 An Interdisciplinary Approach for Quantifying Supervised Model Interpretability

With the ubiquity of machine learning in several domains that directly or indirectly affect people, a new line of research has focused on creating **interpretable** machine learning methods and models. The general goal in this subfield of machine learning is to design models that humans can understand, debug, interact with, and make informed decisions with. Despite this growing body of research, there is still no consensus on the definition and quantification of interpretability; most of the approaches do not consider humans in evaluating their methods and there have been very few studies verifying whether interpretable methods achieve their intended effects on end users.

In Chapter 3, we take the perspective that part of the reason for the difficulty in defining and measuring interpretability is that it is not something that can be directly manipulated or measured. Rather, it is a latent property that is influenced by several possible manipulable factors, which are commonly thought to make models more or less understandable by humans (e.g., the number of features, model transparency, or the visualizations). People's abilities to understand, trust, or debug the model, directly or indirectly, depend on these factor.

While the manipulable factors are properties of the **model**, measurable outcomes (dependent variables) are properties of **human behavior** (e.g., users' trust in the model, users' abilities in debugging the model, users' abilities in making informed decisions with the model). In other words, it is the humans' behavior and abilities that is affected by the notion of model interpretability. Additionally, different end users with different expertise and goals might care about different outcomes to different extent in different scenarios. Therefore, we propose to measure interpretability by isolating the effect of each of the factors of interest and measuring the influence on any outcome of interest.

We take an interdisciplinary approach and build on decades of psychology and social science research on human trust in models. We bring humans in the loop, enable them to interact with a supervised model and complete a task with the help of the model. Designing and running controlled user studies lets us measure how different factors affect people's behavior and abilities in completing the task.

We present a framework for assessing the effects of model interpretability on users via pre-registered experiments in which participants are shown functionally identical supervised models that vary in factors, which are commonly thought to influence interpretability. Using this framework, we run a sequence of large-scale randomized experiments, varying two putative drivers of interpretability: the number of features and the model transparency (clear or black-box). We then explore how these factors interact with trust in the model's predictions, the ability to simulate the model, and the ability to detect the model's mistakes.

Our large-scale crowdsourced experiments show that participants who are shown a clear model with a small number of features are better able to simulate the model's predictions. However, there is no difference in multiple measures of trust. Additionally, clear models do not improve the ability to correct model's mistakes. Given that some of these results contradict common intuition that transparent and simple models lead to higher trust, interpretability research could benefit from more emphasis on empirically verifying that interpretable models achieve all their intended effects.

The experiments in Chapter 3 focus on the effect of interpretability on users behavior and performance in a predictive task with the help of a **supervised** model. Other interesting properties of interpretable models include users' abilities to interact with them and debug them. Next, we design a framework that allows for a rich interaction between humans and models. We evaluate the effectiveness of interpretable **unsupervised** models on users' decision making.

1.3.2 Interactive and Interpretable Unsupervised Machine Learning for Label Induction and Document Annotation

We address the problem of inducing label sets and creating training sets for supervised models by exploiting interpretable unsupervised models and enabling users to interactively debug models.

Effective classification requires experts to annotate data with labels; these training data are time-consuming and expensive to obtain. If you know what labels you want, active learning can reduce the number of labeled data points needed. However, establishing the label set remains difficult. Annotators often lack the global knowledge that is needed to induce a label set.

In Chapter 4, we focus on annotating text data. Our goal is to exploit unsupervised methods to guide people's decision making (i.e., inducing labels and assigning them to documents in this case) and reduce the amount of human effort in annotating text data while preserving performance and quality in a limited time. Additionally, we want to evaluate the effectiveness of our methods with a controlled human-subject experiment.

We introduce ALTO—Active Learning with Topic Overviews—to address the problem of difficulty in inducing label sets and assigning labels to documents for learning text classifiers. ALTO brings humans and machines together to reduce the amount of manual human effort in annotation. It allows for a rich collaboration between humans and machines. Furthermore, our controlled user study provides insights on human behavior and blueprints on the effectiveness of different annotation strategies in different scenarios.

We use topic models to provide a global knowledge of the corpus to users and aid them in inducing global label sets. We also use active learning to provide a local knowledge of individual documents that users lack and guide them in labeling. Topic models are unsupervised models that provide a global overview of the corpus to guide label set induction, and active learning directs them to the right documents to label.

We evaluate ALTO with a controlled human-subject experiment. Our forty-annotator user study shows that while active learning alone is best in extremely resource limited conditions, topic models (even by themselves) lead to better label sets, and ALTO's combination is best overall. Our user study provides insights on which annotation methods are better suitable in different scenarios.

1.3.3 Human-in-the-Loop Machine Learning for a Real-World Use Case

Having demonstrated the success of ALTO—an interpretable and interactive framework—in helping users induce label sets and organize documents, we explore the effectiveness of a similar system on a real-world use case that requires exploring large document collections, finding documents of interest, and answering a question based on retrieved documents. We focus on **understanding science policy** and answering questions about funding policies of the national science foundation (NSF) as our use case.

In Chapter 5, we build on the ALTO interface and design an interactive framework to help users find NSF grants that are relevant to a question, review these grants, and answer questions about these grants. Our framework uses topic models to provide an overview of the NSF grants and help users navigate through them. Furthermore, it allows for information search within the existing grants via an information retrieval tool. Like ALTO, we use active learning to point users to the grants that would be more beneficial in automatically discovering relevant grants to a question.

We evaluate the effectiveness of our framework with a user study with twenty participants. We compare our framework with an alternative that lacks the overview of the NSF grants in terms of topics. Our user study results show that topics inspire users to search more for specific information and explore the corpus. Additionally, self-reported measures show that users find topic information helpful in answering the questions. Contrary to expectation, we do not find that topic information help users answer questions faster and more accurately. The findings and discussions in this chapter should encourage more empirical studies of the effectiveness of topic models in helping humans browse and understand large document collections.

1.4 Thesis Outline

All the proposed frameworks, claims, and hypotheses in this thesis are empirically evaluated with human-subject experiments. This thesis is organized as follows:

- Chapter 2 summarizes existing research work on interpretable supervised and unsupervised machine learning models;

- Chapter 3 introduces an interdisciplinary approach for manipulating and measuring the interpretability of supervised machine learning models. It proposes a template for measuring the effect of model design factors (that are commonly thought to make supervised machine learning models more or less interpretable) on people's abilities in completing (potentially complicated) tasks with the help of models;

- Chapter 4 introduces a framework that exploits unsupervised machine learning models to reduce the amount of manual human effort in classifying large document collections. Our framework allows for a rich collaboration between users and models. It enables users to interact with and build a machine learning model by inducing label sets and assigning labels to documents in a large corpora better and faster;
- Chapter 5 explores the effectiveness of similar approaches on a real-world use case. It introduces an interactive framework, which combines unsupervised machine learning models to provide insights to users with information retrieval systems to help users search for information.
- Chapter 6 concludes this thesis and discusses the possible applications and future extensions of the presented research work.

Chapter 2

Background

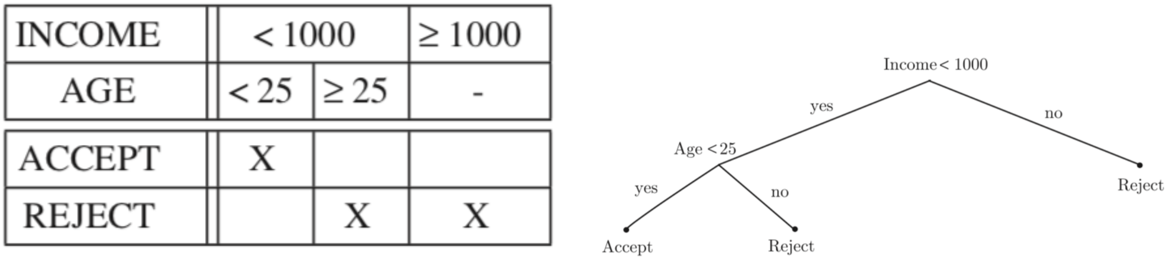
The research in this thesis approaches the problem of interpretability in machine learning from an interdisciplinary perspective. We argue that to understand and measure interpretability, we need to bring humans in the loop. We design interfaces that bring humans and machines together and evaluate the interpretability of our machine learning methods with a task-driven approach.

This chapter starts by summarizing existing research on interpretability of supervised (Section 2.1) and unsupervised models (Section 2.2) —the two primary categories of machine learning algorithms. Section 2.3 reviews topic modeling algorithms in more detail as they are unsupervised machine learning models that are commonly used for exploring large document collections and are key to the research presented in this thesis. Finally, Section 2.4 summarizes existing tools for visualizing unsupervised machine learning models with a focus on topic models.

2.1 Interpretability and Visualization of Supervised Models

Supervised learning is the task of inferring an output for an input using a set of input-output pairs. The set of input-output pairs is the **training set**. This training set is usually a subset of the original data set, where each data point has metadata (output) associated with it. This metadata is either in the discrete form of **labels** (e.g., whether an image is a dog, cat, or bird) or in the continuous value form (e.g., the rating of a movie in a review on social media). The former is referred as **classification** or **categorization** (Kotsiantis et al., 2007), while the latter is **regression** (Fox, 1997).

Supervised models have three primary components: input representation (features), model, and out-



INCOME	< 1000		≥ 1000
AGE	< 25	≥ 25	-
ACCEPT	X		
REJECT		X	X

(a) An example of a decision table.

(b) An example of a decision tree.

Figure 2.1: An Example of visualization of model internals for (a) a decision table and (b) a decision tree from (Huysmans et al., 2011). Decision tables and trees are considered to be "interpretable" to humans.

put (predictions). Interpretability is usually used in the context of **model internals**. For example, decision tables (Vanthienen and Wets, 1994) (Figure 2.1a) and decision trees (Safavian and Landgrebe, 1991) (Figure 2.1b) provide significant comprehensibility advantages over other models and thus are thought to be more suitable in scenarios where interpretability is a key requirement (Huysmans et al., 2011). As model-level interpretability has gained a lot of attention, several existing works attempt to visualize models to aid better and faster understanding and interpretation. Examples include a system for interactive construction, visualization, and exploration of decision trees (Teoh and Ma, 2003) and a visualization tool for providing insight on function of intermediate layers in convolutional networks (Zeiler and Fergus, 2014).

The other two components—features and predictions—can potentially affect interpretability as well. For example, while modern deep models (LeCun et al., 2015) usually lack model-level interpretability, they tend to operate on raw or lightly-processed features. Thus, if nothing else, the inputs are meaningful and one can provide easy-to-understand explanations for them. On the other hand, linear models have interpretable internals but they usually operate on heavily hand-engineered features to get comparable performance, which makes comprehension potentially harder.

As predictions made by supervised models are commonly associated with uncertainty, effective visualization and communication of uncertainty is a relevant research area. Understanding probability distributions and interpreting them is challenging for humans. For example, using natural frequency and discrete outcomes as replacement for abstract and continuous probabilities lead to more accurate precision estimate by medical experts (Hoffrage and Gigerenzer, 1998) and improved patient understanding of risk (Garcia-Retamero and Cokely, 2013). As such, effective visualization of uncertainty is a related and ongoing area of research (Spiegelhalter et al., 2011; Kay et al., 2016).

To address the problem with the lack of interpretability in different components of supervised models, a new line of research focuses on developing methods that are interpretable. There are two common approaches:

The first is to employ models that are intrinsically simple, such as models in which the impact of each feature on the model's prediction is easy to understand. Examples include generative additive models, where the dependent variables are modeled as the sum of univariate terms and pairwise interaction terms and thus the relationship between the univariate and interaction terms can be visualized via easy-to-understand two-dimensional or three-dimensional plots (Lou et al., 2012, 2013; Caruana et al., 2015) and point systems, which use sparse integer linear models that users can add, subtract, and multiply a few small numbers in order to make a prediction (Jung et al., 2017; Ustun and Rudin, 2016).

The second is to provide **post-hoc explanations** for (potentially complex) models. One thread of research in this direction looks at how to explain individual predictions by learning locally faithful linear approximations of the model around particular data points (Ribeiro et al., 2016), learning importance scores of each feature for a particular prediction (Lundberg and Lee, 2017), or estimating the influence of training examples on a particular prediction (Koh and Liang, 2017). Another thread of research in this area relies on visualizing model outputs for explaining predictions (Kulesza et al., 2015; Wattenberg et al., 2016).

Despite the activity and innovation in this area, there is still no consensus about how to define, quantify, or measure the interpretability of a supervised machine learning model. In Chapter 3, we argue that interpretability should be measured with humans in the loop. We hypothesize that users will better understand interpretable models and they will trust interpretable models more. We introduce an experimental template for systematically isolating the effect of factors that are thought to influence the interpretability of a supervised regression model and measuring their effect on users' abilities to simulate the model's prediction, trust the model, and detect the model's mistakes.

2.2 Interpretability of Unsupervised Models

Unsupervised learning is the task of inferring hidden structure from raw data sets (Hastie et al., 2009). Unlike supervised learning, there is no training set where the data points are associated with metadata. Unsupervised methods such as clustering (Jain et al., 1999) are often used for exploratory data analysis, which is the subfield of summarizing the characteristics of data sets, often with visual methods. As such, these models are popular in social sciences and thus, interpretability becomes more important as we discuss in Section 1.1.

Similar to supervised models, the representation of the data can affect interpretability of unsupervised models. Unlike supervised models, the output of unsupervised models is not in the form of a prediction, rather it is an inferred hidden structure from data. Therefore, **interpretability of the hidden structure** is the focus of the research in the area of unsupervised model interpretability. We now briefly review the existing work on interpretability of examples of different unsupervised frameworks.

Representation learning (Bengio et al., 2013) is a popular framework for unsupervised learning, where the goal is to automatically learn representations of data that can be used for many downstream tasks such as classification and regression. As more interpretable representations lead to better understanding of models and methods in these downstream tasks, Chen et al. (2016) use an information-theoretic method to learn interpretable and disentangled representations of data.

Word embedding algorithms (Mikolov et al., 2013a,b; Pennington et al., 2014) are another example of unsupervised models that their interpretability has gained attention. Word embedding models encode meanings of words to vector spaces. These embeddings capture semantic relationships between words and have been used to improve the performance in many NLP tasks such as part-of-speech tagging (Lin et al., 2015), sentiment analysis (Kim, 2014), and named entity recognition (Guo et al., 2014). Interpreting the dimensions of the vector space—which can provide insights in tasks that require semantic interpretation (e.g., named entity recognition)—is challenging because these vectors are usually dense.

One common approach to make embeddings more interpretable is to learn **sparse** vectors, where each word has a **small** number of active dimensions. Murphy et al. (2012) use non-negative matrix factorization, Faruqui and Dyer (2015) use pre-constructed linguistic resources such as WordNet (Miller, 1995), Faruqui et al. (2015) use sparse coding, and Subramanian et al. (2017) use $k-$sparse autoencoders (Makhzani and Frey, 2013) to induce sparse representations. Senel et al. (2017) take a different approach and propose a method to capture the hidden semantic concept of vectors in word embedding models based on conceptual categories (Vulić et al., 2017).

Topic models are another family of unsupervised models that are commonly used for exploratory analysis of large document collections. By inducing thematic structure from large corpora and automatically organizing documents based on their theme, these models help users wade through documents, understand them, and find information of interest better and faster. As topic models are key to the methodologies introduced in this thesis (Chapters 4 and 5), we now provide some preliminary background information on them.

2.3 Topic Models: Unsupervised Exploratory Tools for Large-Scale Text Data

Topic models automatically induce structure from a data set of documents. Given a corpus and a constant number K—the number of topics—topic models output (1) K topics, where each topic k is defined as a distribution over words ($\phi_{k,w}$) and (2) a distribution over topics for each document d ($\theta_{d,k}$). Topic models are exemplified by Latent Dirichlet Allocation (Blei et al., 2003, LDA) and have been widely used for different purposes such as information retrieval, visualization, statistical inference, multilingual modeling, and linguistic understanding (Boyd-Graber et al., 2017).

Similar to many other unsupervised models, topic models are commonly used for exploratory purposes such as understanding NIH-funded research (Talley et al., 2011) and historical trend of ideas in a research field (Hall et al., 2008). Therefore, interpretability is of special interest. Each discovered topic is usually interpreted by words with the highest marginal probability in the ϕ distribution. Figure 2.2 shows an example of topics and their prominent words from the frequently used 20 Newsgroups data set (Lang, 2007). When the goal is to explore and understand large corpora, the induced set of topics guide people's exploration and help them discover thematic structure from the content of documents. Similar to any other machine learning model, the evaluation of topic models should be done in the context of their intended goals

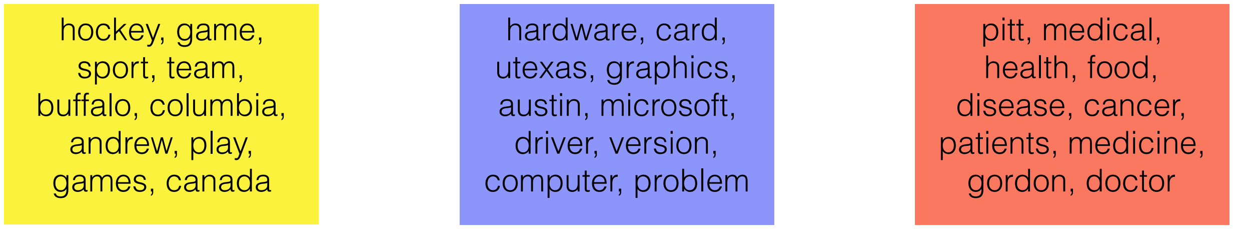

hockey, game, sport, team, buffalo, columbia, andrew, play, games, canada
hardware, card, utexas, graphics, austin, microsoft, driver, version, computer, problem
pitt, medical, health, food, disease, cancer, patients, medicine, gordon, doctor

Figure 2.2: Three out of nineteen topics discovered by LDA from the 20 Newsgroups corpus. Topics reveal thematic structure from large-scale textual data and can be interpreted using their prominent words.

(i.e., **users'** abilities in exploring and understanding large corpora) to ensure that humans can effectively trust and use them. We now elaborate on the existing methods for evaluating topic models and measuring their interpretability. While there are several variations and extensions of LDA such as supervised topic models that jointly model document content and their associated metadata (e.g., labels or ratings) associated with them (Mcauliffe and Blei, 2007; Zhu et al., 2012; Ramage et al., 2009; Nguyen et al., 2013, 2014b), in this thesis we focus on LDA as a tool for exploring and understanding documents.

2.3.1 Evaluation

Traditionally, topic modeling evaluation has focused on **perplexity**, which measures how well a model can predict words in unseen documents (Wallach et al., 2009). However, perplexity is not in line with the goal of users in scenarios that topic models are used for exploratory purposes. In these cases, **comprehensibility** is more important than **predictive power**. Therefore, topic-level interpretability, i.e., interpretability of the inferred hidden structure, is of interest. One usually cares about how good the topics capture the thematic structure of the corpus, rather than their predictive power. This is related to the prediction-explanation dilemma discussed in Chapter 1.

Word intrusion—a task-based evaluation method—is commonly used to evaluate topic models based on their topic-level interpretability (Chang et al., 2009). The idea is that if a topic is interpretable and semantically coherent, humans should be able to easily find the **intruder word**, which is a randomly selected word. For example, when humans are shown words metropolitan, carrier, rail, agriculture, freight, passenger, they will be able to detect agriculture as the intruder because the combination of other words construct an interpretable topic, which is generally about "transportation". Followed by this work, Lau et al. (2014) show that topic coherence (Newman et al., 2010)—an automatic way of measuring topic interpretability—correlates with what humans consider more or less interpretable. To calculate topic coherence, the co-occurrence probabilities of top-N topic words are looked up in a reference corpus (e.g., Wikipedia). Then, topic coherence C is calculated based on pairwise normalized point-wise mutual information between topic words:

$$C = \sum_{j=2}^{N} \sum_{i=1}^{j-1} \frac{\log \frac{P(w_j, w_i)}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)}, \tag{2.1}$$

where $P(w_j, w_i)$ is the probability of observing both w_i and w_j in a reference document, and $P(w_i)$ and $P(w_j)$ are the probabilities of observing w_i and w_j in a reference document, respectively. While several extensions of the discussed coherence score have been proposed (Morstatter and Liu, 2016; Ramrakhiyani et al., 2017), in this thesis, we use Equation 2.1 to calculate topic coherence.

A common challenge with using topic models in practice is creating a list of stop words, i.e., words that are extremely common. Stop words provide little value in helping users understand the semantics of the topics and thus lead to lower levels of interpretability and coherence scores. For example, words such as bill, session, and committee appear in almost every congressional bill and provide no help in understanding the content of the bills. To prevent the degradation in interpretability of topics, these words are usually excluded from the vocabulary. We use **term frequency - inverse document frequency (TF-IDF)** scores to hand-filter words based on their semantic content in the corpus domain and generate corpus-specific stop words. The TF-IDF score increases proportionally to the number of times a word w appears in the document d and decreases proportionally to the number of documents it appears in. Intuitively, the TF-IDF score is higher for a word that appears frequently in only a few documents:[1]

$$\text{TF-IDF}(w, d, D) = \text{TF}(w, d).\text{IDF}(w, D), \tag{2.2}$$

where D is the document collection, $\text{TF}(w, d)$ is the number of times w appears in document d and $\text{IDF}(w, D)$ is the inverse frequency of documents in the collection that w appears in:

$$\text{IDF}(w, D) = \frac{|D|}{|d \in D : w \in d|}. \tag{2.3}$$

[1] We calculate the average TF-IDF over all documents in the corpus to find stop words.

Like any clustering algorithm, setting the number of topics, K, is a challenge in topic modeling. Low numbers of K usually lead to topics that are general and high numbers of K usually lead to overly specific topics. Both of these cases lead to degradation in interpretability. In this thesis, since we are interested in using topic models for exploratory purposes and thus we are interested in maximizing interpretability, we set K based on the coherence metric described above. We calculate the average topic coherence in a pre-defined range for K and select the K value that leads to the maximum mean coherence score.

2.4 Visualization of Unsupervised Models

As unsupervised models are often used to understand large data sets and get insights, effective visualization of these models is of special interest. This section reviews some of the existing work in visualizing unsupervised models.

Unsupervised models commonly operate on high-dimensional vector representation of data. Several existing work focus on visualizing high-dimensional data representations using graphs (Di Battista et al., 1994) or pixel-based techniques (Keim, 2000). Dimensionality reduction methods such as Principal Component Analysis (Hotelling, 1933, PCA) and Maximum Variance Unfolding (Weinberger et al., 2004, MVU) make these visualizations more interpretable. A more recently proposed and widely used visualization technique for high-dimensional data (e.g., word embeddings) is t-Distributed Stochastic Neighbor Embedding (Maaten and Hinton, 2008, t-SNE), which visualizes the similarity between data **and** reveals important global structure such as clusters.

The visualization of topic models has also gained attention. Some of the existing work in the visualization of topic models focuses on visualizing individual topics. While the simplest and most commonly used visualization of topics is in the form of list of word that are ordered based on the marginal probability of each word in a topic, other visualization techniques such as word clouds and network graphs exist (Smith et al., 2014). Furthermore, image labels (Aletras and Stevenson, 2014) and textual labels (Mei et al., 2007; Magatti et al., 2009; Lau et al., 2010, 2011; Hulpus et al., 2013; Aletras and Stevenson, 2014; Wan and Wang, 2016) provide more compact representations, which explicitly identify the semantics of topics. Though compact, textual labels have been shown to be as effective as word lists in a document retrieval task (Aletras et al., 2014).

In Chapter 4, we compare the automatically generated document labels using the graph-based approach proposed by Aletras and Stevenson (2014) to the labels generated by humans. This method builds on the common approach to generate textual labels for topics, which is to find an article in a reference corpus (e.g., Wikipedia), that is the most representative of topic words. To find Wikipedia articles that are representative of topic words, Wikipedia is queried with top twenty words in each topic. The titles of retrieved Wikipedia articles are then treated as **candidate labels**. More candidate labels are generated by finding noun-chunks and n-grams in the noun-chunks that are Wikipedia articles themselves. Next, topic words are used to query a search engine and retrieve related articles. The words in the titles of these articles are nodes in a graph. The edges are created based on the word co-occurrences in the retrieved articles. Next, the PAGERANK algorithm (Page et al., 1999) is used to rank the nodes (word types). Each candidate label is then scored according to the score of its tokens and the label with the highest score is chosen as the topic label.

There are several tools that provide an overview of large corpora through the entirety of topic models. The topic browser (Gardner et al., 2010), TopicViz (Eisenstein et al., 2012), and the topic model visualization engine (Chaney and Blei, 2012) are examples of interactive tools that enable users to explore large corpora via topics and their associated documents. In these interactive tools, topics are a protocol to find documents of interest from the large corpus. More elaborate tools provide additional information from the topic model. For example, LDAVis (Sievert and Shirley, 2014) provides information about the relationship between topics and Soo Yi et al. (2005) use a dust-and-magnet visualization to show the forces of each topic on documents.

While there have been studies to understand whether people interpret particular visualizations of **individual** topics better than others (Aletras et al., 2014; Smith et al., 2017),[2] there has been no systematic study to find out whether the existing topic model based tools actually help users navigate through document

[2] This work, in collaboration with Alison Smith, Tak Yeon Lee, Jordan Boyd-Graber, Kevin Seppi, Niklas Elmqvist, and Leah Findlater, examines the interpretability of various topic visualizations techniques. I contributed by generating interpretable automatic textual labels for individual topics. This work was published in Transactions of the Association for Computational Linguistics (TACL), 2017.

collections, understand documents, and complete a task with their help. In Chapters 4 and 5, we design a similar interface and conduct a human-the-loop study to understand the effect of topics on users' abilities in understanding large corpora, labeling documents, and completing a task.

2.5 Summary

In this chapter, we summarized existing work on interpretability of supervised models and unsupervised models. Additionally, we reviewed the fundamentals of topic models, the evaluation methods that are inspired by interpretability rather than predictive power, and the existing tools that exploit them to help users get insights on large textual data sets. In the following chapter, we study interpretability of supervised models with humans in the loop. We introduce an experimental and task-driven approach to bring humans in the loop, expose them to a supervised machine learning model and its predictions, and ask them to complete a predictive task with the help of the model. This framework helps us understand and quantify the interpretability of a supervised regression model.

Chapter 3

Manipulating and Measuring Model Interpretability[1]

Despite the flurry of activity and innovation in developing "interpretable" machine learning methods that humans can interact with, understand and debug, and make informed decisions with (Chapter 2), there is still no consensus about how to define, quantify, or measure the interpretability of a machine learning model (Doshi-Velez and Kim, 2017). Indeed, different notions of interpretability, such as simulatability, trustworthiness, and simplicity, are often conflated (Lipton, 2016). This problem is exacerbated because different users of machine learning systems have different needs in different scenarios. For example, visualizations or explanations of interest for a regulator who wants to understand why a particular person was denied a loan may be different from the interests of a data scientist who tries to debug a machine learning model.

The difficulty of defining interpretability stems from the fact that interpretability is not something that can be directly manipulated or measured. Rather, interpretability is a latent property that can be influenced by different manipulable factors such as the number of features, the complexity of the model, the transparency of the model, or even the user interface. People's behavior such as their ability to simulate, trust, or debug the model depends on these manipulable factors. Different factors may influence people's behavior and abilities in different ways. As such, we argue that to understand interpretability, it is necessary to directly manipulate these factors and measure their influence on real people's abilities in successfully

[1] An earlier version of this chapter was published as: Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, Hanna M. Wallach. Manipulating and measuring model interpretability. In NIPS workshop on Transparent and Interpretable Machine Learning in Safety Critical Environments, 2017 (Poursabzi-Sangdeh et al., 2017). The updated version is in submission.

completing tasks.

The endeavor to understand the effect of different manipulable factors on people's behavior and abilities goes beyond the realm of typical machine learning research. While the factors that influence interpretability are properties of the system design, the outcomes that we would ultimately like to measure are properties of human behavior. Because of this, building interpretable machine learning models is not a purely computational problem. In other words, what is or is not "interpretable" is defined by people, not algorithms. We therefore take an interdisciplinary approach, building on decades of psychology and social science research on human trust in models (e.g., Önkal et al., 2009; Dietvorst et al., 2015; Logg, 2017). The general approach used in this literature is to run randomized human-subject experiments in order to isolate and measure the influence of different manipulable factors on trust. Our goal is to apply this approach in order to understand interpretability—i.e., the relationships between properties of the system design and properties of human behavior.

We present a sequence of large-scale randomized human-subject experiments, in which we vary factors that are thought to make models more or less interpretable (Glass et al., 2008; Lipton, 2016) and measure how these changes affect people's decision making. We focus on two factors that are often assumed to influence interpretability, but rarely studied formally: the number of features and the model transparency, i.e., whether the model internals are **clear** or **black-box**. A **clear** model has a completely transparent internals and users can see how, exactly, the model makes its predictions on any input. On the other hand, the internals of a **black-box** model is completely hidden from users.

We focus on laypeople as opposed to domain experts, and ask which factors help them simulate a model's predictions, gain trust in a model, and understand when a model will make mistakes. While others have used human-subject experiments to validate or evaluate particular machine learning innovations in the context of interpretability (e.g., Ribeiro et al., 2016; Lim et al., 2009), we attempt to isolate and measure the influence of different factors on human behavior in a sequence of large-scale and crowdsourced experiments.

In each of our experiments, which were pre-registered to avoid pitfalls raised by the "replication crisis" (Goldacre, 2016), participants are asked to predict the prices of apartments with the help of a machine learning model.[2] Each apartment is represented in terms of eight features: number of bedrooms, number of bathrooms, square footage, total rooms, days on the market, maintenance fee, distance from the subway, and distance from a school. All participants see the same set of apartments (i.e., the same feature values) and, crucially, the same model prediction for each apartment, which comes from a linear regression model. What varies between the experimental conditions is **only the presentation of the model**. As a result, any observed differences in the participants' behavior between the conditions can be attributed entirely to the model presentation.

We run three experiments: in the first experiment, we ask participants to predict the prices of apartments in a single neighborhood in New York City. The second experiment examines the effect of high prices of New York City apartments on participants' behavior in our first experiment. Finally, the third experiment measures an alternative metric for trust.

In our first experiment (Section 3.1), where participants are asked to predict the prices of apartments in a single neighborhood in New York City, we hypothesize that participants who are shown a clear model with a small number of features will be better able to simulate the model's predictions and more likely to trust (and thus follow) the model's predictions. We also hypothesize that participants in different conditions will exhibit varying abilities to correct the model's inaccurate predictions on unusual examples. As predicted, participants who are shown a clear model with a small number of features are better able to simulate the model's predictions; however, they are not more likely to trust the model's predictions. Additionally, participants who are shown a clear model are less able to correct inaccurate predictions.

In our second experiment (Section 3.2), we scale down the apartment prices and maintenance fees to match median housing prices in the U.S. in order to determine whether the findings from our first experiment were merely an artifact of New York City's high prices. Even with scaled-down prices and fees, the findings from our first experiment replicate.

In our third experiment (Section 3.3), we dig deeper into our finding that there is no difference in trust between the conditions. To make sure that this finding is not simply due to our measures of trust, we

[2] We choose real estate as a domain that would be familiar to participants, at least at a basic level (e.g., they should understand what the features mean).

instead use the **weight of advice** measure frequently used in the literature on advice-taking (Yaniv, 2004; Gino and Moore, 2007) and subsequently used in the context of algorithmic predictions by Logg (2017). We hypothesize that participants will give greater weight to the predictions of a clear model with a small number of features than the predictions of a black-box model with a large number of features, and update their own predictions accordingly. We also hypothesize that participants' behavior may differ if they are told that the predictions are made by a "human expert" instead of a black-box model with a large number of features. Even with the weight of advice measure, there is no difference in trust between the conditions. There is also no difference in the participants' behavior when they are told that the predictions are made by a human expert.

We view these experiments as a first step toward a larger agenda aimed at quantifying and measuring the impact of different manipulable factors that influence interpretability.

3.1 Experiment 1: Predicting Apartment Prices

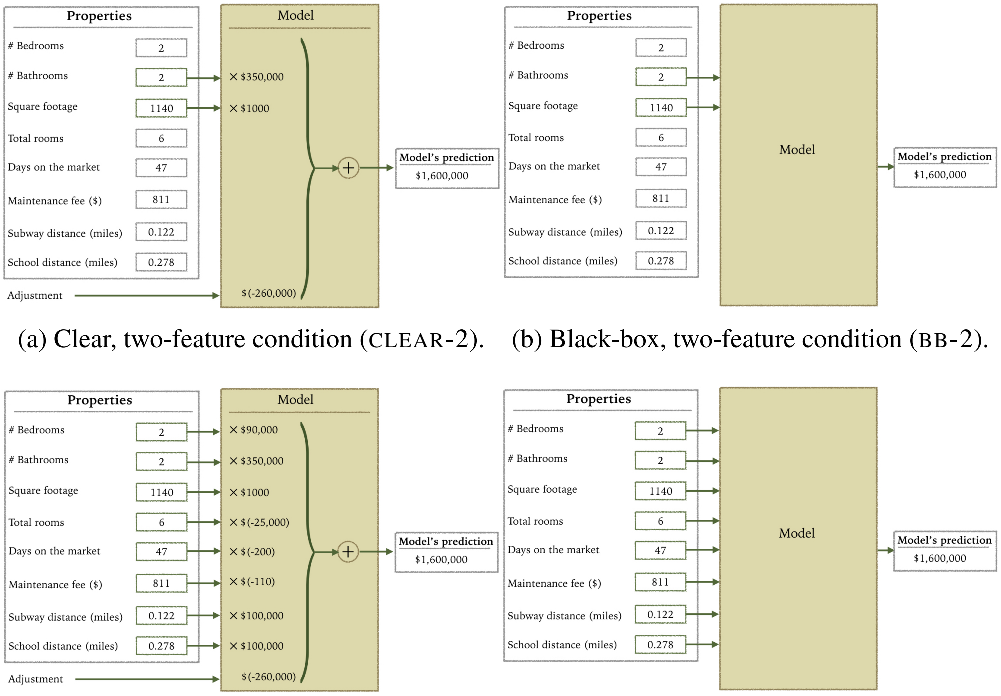
Our first experiment is designed to measure the influence of the number of features and the model transparency on three properties of human behavior that are commonly associated with interpretability: laypeople's abilities to simulate a model's predictions, gain trust in a model, and understand when a model will make mistakes. Before running the experiment, we posited and pre-registered three hypotheses:[3]

H1. **Simulation.** A clear model with a small number of features will be easiest for participants to simulate.

H2. **Trust.** Participants will be more likely to trust (and thus follow) the predictions of a clear model with a small number of features than the predictions of a black-box model with a large number of features.

H3. **Detection of mistakes.** Participants in different conditions will exhibit varying abilities to correct the model's inaccurate predictions on unusual examples.

[3] Pre-registered hypotheses are available at `https://aspredicted.org/xy5s6.pdf`.



(a) Clear, two-feature condition (CLEAR-2). (b) Black-box, two-feature condition (BB-2).

(c) Clear, eight-feature condition (CLEAR-8). (d) Black-box, eight-feature condition (BB-8).

Figure 3.1: The four primary experimental conditions. In the conditions on top, the model uses two features; on the bottom, it uses eight. In the conditions on the left, participants see the model internals; on the right, they are presented with the model as a black box. Crucially, participants in all conditions are shown the same apartment properties and the same model predictions. Therefore, any difference in participants' behavior can be attributed to the number of features that the model uses and whether the model is clear or black-box.

For unusual examples, we intentionally did not pre-register any hypotheses about which conditions would make participants more or less able to correct inaccurate predictions. On the one hand, if a participant understands the model better, she may be better equipped to correct examples on which the model makes mistakes. On the other hand, a participant may place greater trust in a model she understands well, leading her to closely follow its predictions.

Prediction error. Finally, we pre-registered our intent to analyze participants' prediction error in each condition, but intentionally did not pre-register any directional hypotheses.

3.1.1 Experimental Design

As explained in the previous section, we ask participants to predict apartment prices with the help of a machine learning model. We show all participants the same set of apartments and the same model prediction for each apartment. What varies between the experimental conditions is only the presentation of the model. We consider four primary experimental conditions in a 2×2 design:

- Some participants see a model that uses only two features (number of bathrooms and square footage—the two most predictive features, CLEAR-2 and BB-2), while some see a model that uses all eight features (CLEAR-8 and BB-8). (All eight feature values are visible to participants in all conditions.)

- Some participants see the model internals (i.e., a linear regression model with visible coefficients, CLEAR-2 and CLEAR-8), while some are presented with the model as a black box (BB-2 and BB-8).

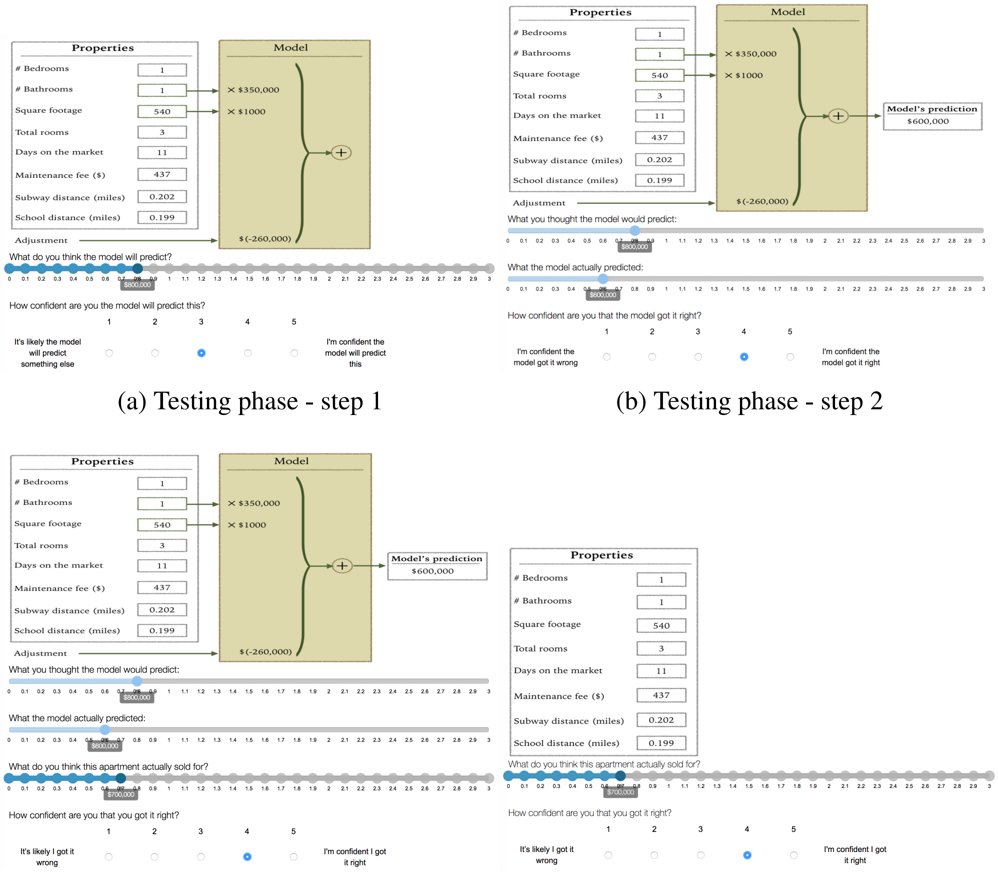
Screenshots from each of the four primary experimental conditions are shown in Figure 3.1. We additionally consider a baseline condition in which there is no model available.

We run the experiment on Amazon Mechanical Turk using psiTurk (Gureckis et al., 2016), an open-source platform for designing online experiments. The experiment was IRB-approved. We recruit 1,250 participants, all located in the U.S., with Mechanical Turk approval ratings greater than 97%. We randomly assign participants to the five conditions (CLEAR-2, $n = 248$ participants; CLEAR-8, $n = 247$; BB-2, $n = 247$; BB-8, $n = 256$; and NO-MODEL, $n = 252$).[4] Each participant received a flat payment of $2.50.[5]

Participants were first shown detailed instructions, including, in the clear conditions, a simple English description of the corresponding two- or eight-feature linear regression model (Appendix A.1), before proceeding with the experiment in two phases. In the **training phase**, participants were shown ten apartments in a random order. In the four primary experimental conditions, participants were shown the model's

[4] We do not screen for familiarity with the domain and we do not detect and filter out any outliers; randomization across our large sample of participants ensures that participants' levels of familiarity are similarly distributed across conditions.

[5] We estimated the total time that workers would spend on the task based on our pilot studies and determined the payments for all experiments to match an hourly minimum wage of ten dollars.



Properties	
# Bedrooms	1
# Bathrooms	1
Square footage	540
Total rooms	3
Days on the market	11
Maintenance fee ($)	437
Subway distance (miles)	0.202
School distance (miles)	0.199

What do you think the model will predict? $800,000

How confident are you the model will predict this? It's likely the model will predict something else ○ ○ ● ○ ○ I'm confident the model will predict this

(a) Testing phase - step 1

Model's prediction $600,000

What you thought the model would predict: $800,000

What the model actually predicted: $600,000

How confident are you that the model got it right? I'm confident the model got it wrong ○ ○ ○ ● ○ I'm confident the model got it right

(b) Testing phase - step 2

Model's prediction $600,000

What you thought the model would predict: $800,000

What the model actually predicted: $600,000

What do you think this apartment actually sold for? $700,000

How confident are you that you got it right? It's likely I got it wrong ○ ○ ○ ● ○ I'm confident I got it right

(c) Testing phase - step 3

What do you think this apartment actually sold for? $700,000

How confident are you that you got it right? It's likely I got it wrong ○ ○ ○ ● ○ I'm confident I got it right

(d) Testing phase - baseline

Figure 3.2: The testing phase in the first experiment: (a) participants are asked to guess the model's prediction and state their confidence (step 1), (b) participants are asked to state their confidence in the model (step 2), (c) participants are asked to make their own prediction and state their confidence (step 3), and (d) in the baseline condition, participants are asked to predict the price and indicate their confidence.

prediction of each apartment's price, asked to make their own prediction, and then shown the apartment's actual price. In the baseline condition, participants were asked to predict the price of each apartment and then shown the actual price. In the **testing phase**, participants were shown another twelve apartments. The order of the first ten was randomized, while the remaining two always appeared last, for reasons described below. In the four primary experimental conditions, participants were asked to guess what the model would predict for each apartment (i.e., simulate the model) and to indicate how confident they were in this guess on a five-point scale (Figure 3.2a). They were then shown the model's prediction and asked to indicate how confident they were that the model was correct (Figure 3.2b). Finally, they were asked to make their own

Feature	Actual Coefficient	Rounded Coefficient
#Bedrooms	89030.82	90,000
#Bathrooms	353252.43	350,000
#Square footage	997.76	1000
Total rooms	-25816.23	-25,000
Days on the market	-193.79	-200
Maintenance fee ($)	-111.87	-110
Subway distance (miles)	85437.63	100,000
School distance (miles)	71775.32	100,000
Intercept	-259615.78	-260,000

Table 3.1: The actual regression coefficients and the rounded coefficients that are used in our experiments.

prediction of the apartment's price and to indicate how confident they were in this prediction (Figure 3.2c). In the baseline condition, participants were asked to predict the price of each apartment and to indicate their confidence (Figure 3.2d).

The apartments shown to participants are selected from a data set of actual Upper West Side apartments taken from StreetEasy,[6] a popular and reliable New York City real estate website, between 2013 and 2015. To create the models for the four primary experimental conditions, we first train a two-feature linear regression model on our data set using ordinary least squares with Python's scikit-learn library (Pedregosa et al., 2011), rounding coefficients to "nice" numbers within a safe range.[7] To keep the models as similar as possible, we fix the coefficients for number of bathrooms and square footage and the intercept of the eight-feature model to match those of the two-feature model, and then train a linear regression model with the remaining six features, following the same rounding procedure to obtain "nice" numbers. The coefficients are shown in Table 3.1. When presenting the model predictions to participants, we round predictions to the nearest $100,000.

To enable comparisons across experimental conditions, the ten apartments used in the training phase and the first ten apartments used in the testing phase are selected from those apartments in our data set for which the rounded predictions of the two- and eight-feature models agree and chosen to cover a wide range of deviations between the models' predictions and the apartments' actual prices. By selecting only

[6] `https://streeteasy.com/`

[7] In particular, for each coefficient, we find a value that is divisible by the largest possible exponent of ten and is in the safe range, which is the coefficient value plus or minus stderr/4.

apartments for which the two- and eight-feature models agree, we are able to ensure that what varies between the experimental conditions is only the presentation of the model. As a result, any observed differences in the participants' behavior between the conditions can be attributed entirely to the model presentation.

The last two apartments used in the testing phase are chosen to test our third hypothesis—i.e., that participants in different conditions will exhibit varying abilities to correct the model's inaccurate predictions on unusual examples. To test this hypothesis, we would ideally use an apartment with strange or misleading features that causes the two- and eight-feature models to make the same bad prediction. Unfortunately, there is no such apartment in our data set, so we choose two examples to test different aspects of our hypothesis. Both of these examples exploit the models' large coefficient ($350,000) for number of bathrooms. The first (apartment 11) is a one-bedroom, two-bathroom apartment from our data set for which both models make high, but different, predictions. Comparisons between the two- and eight- feature conditions are therefore impossible, but we can examine differences in accuracy between the clear and black-box conditions. The second (apartment 12) is a synthetically generated one-bedroom, three-bathroom apartment for which both models make the same (high) prediction, allowing comparisons between all conditions, but ruling out accuracy comparisons since there is no ground truth. These apartments are always shown last to avoid the previously studied phenomenon in which people trust a model less after seeing it make a mistake (Dietvorst et al., 2015).

3.1.2 Results

Having run our experiment, we compare participants' behavior across the conditions. We compare multiple responses from multiple participants, which is complicated by possible correlations among any given participant's responses. For example, some people might consistently overestimate apartment prices regardless of the condition they are assigned to, while others might consistently provide underestimates. We address this by fitting a mixed-effects model for each measure of interest to capture differences across conditions while controlling for participant-level effects—a standard approach for analyzing repeated measures experimental designs (Bates et al., 2015). We derive all plots and statistical tests from these models; plots show averages with one standard error by condition from the fitted models, and statistical tests report degrees of freedom, test statistics, and p-values under the models.[8] Unless otherwise noted, all plots and statistical tests correspond to just the first ten apartments from the testing phase.

H1. **Simulation.** We define a participant's simulation error to be the absolute deviation between the model's prediction, m, and the participant's guess for that prediction, u_m—that is, $|m - u_m|$.[9] Figure 3.3a shows the mean simulation error in the testing phase. As hypothesized, participants in the CLEAR-2 condition have lower simulation error, on average, than participants in the other conditions ($t(996) = 11.91$, $p < .001$ for the contrast of CLEAR-2 with the other three primary conditions). On average, participants in this condition have some understanding of how the model works. Participants in the CLEAR-8 condition appear to have **higher** simulation error, on average, than participants in the BB-8 condition who could not see the model's internals ($t(996) = 3.00$, $p = .002$ for the contrast of CLEAR-8 with BB-8), though we note that this comparison is not one we pre-registered and could be due to chance.

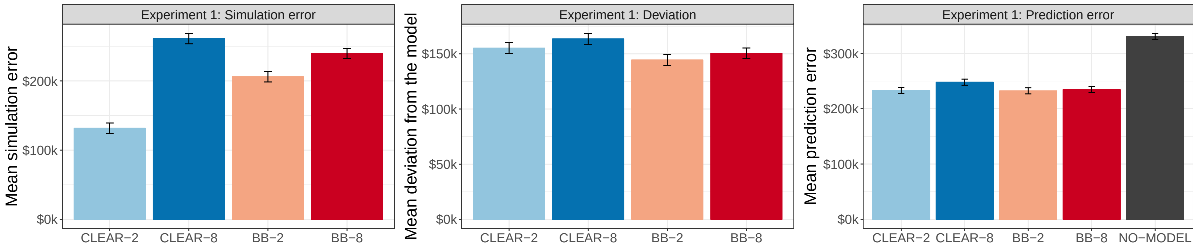
H2. **Trust.** To measure trust, we calculate the absolute deviation between the model's prediction, m, and the participant's prediction of the apartment's price, u_a—that is, $|m - u_a|$: if a participant trusts the model to a good degree, she should **follow** the model in making her own prediction of the price; a smaller value indicates higher trust.[10] Figure 3.3b shows that contrary to our second hypothesis, there is no significant difference in participants' deviation from the model between CLEAR-2 and BB-8. (There are statistically but not practically significant differences in participants' self-reported confidence in the models' predictions.)

H3. **Detection of mistakes.** We use the last two apartments in the testing phase (apartment 11 and apartment 12) to test our third hypothesis. The models make erroneously high predictions on these examples. For both apartments, participants in the four primary experimental conditions overestimate the apartments' prices, compared to participants in the baseline condition (Figures 3.5a and 3.5b). We suspect that this is due

[8] We follow standard notation, where, e.g., the result of a t-test with n degrees of freedom is reported as $t(n) = x, p = y$, where x is the value of the test statistic and y is the corresponding p-value.

[9] Relative deviation between the model's prediction and the participant's guess for that prediction ($|m - u_m|/m$) leads to similar trends in all experiments.

[10] Relative deviation between the model's prediction and the participant's prediction of the apartment's price ($|m - u_a|/m$) leads to similar trends in all experiments.



(a) (b) (c)

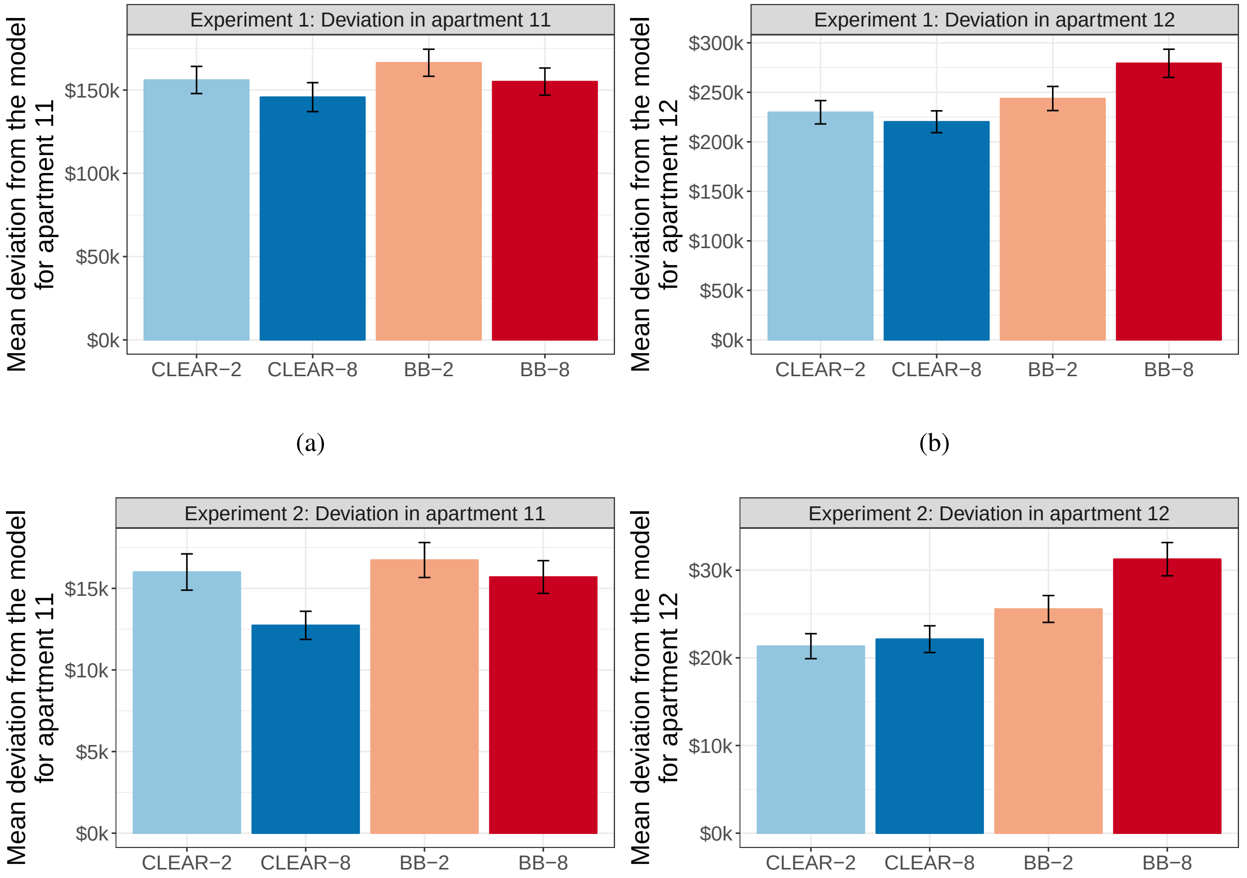
Figure 3.3: Results from our first experiment: (a) mean simulation error, (b) mean deviation of participants' predictions from the model's prediction (a smaller value indicates higher trust), and (c) mean prediction error. Error bars indicate one standard error. While participants in the CLEAR-2 condition are better able to simulate the model's predictions, there is no significant difference across the four primary conditions in terms of participants' trust in the model (in terms of deviation from the model) and their final prediction error.

to an anchoring effect around the models' predictions. For apartment 11, there is no significant difference in participants' deviation from the model's prediction between the four primary conditions (Figure 3.4a). For apartment 12, there is a significant difference between the clear and black-box conditions ($t(996) = 2.96$, $p = .003$ for the contrast of CLEAR-2 and CLEAR-8 with BB-2 and BB-8). In particular, participants in the clear conditions deviate from the model's prediction less, on average, than participants in the black-box conditions, resulting in even worse final predictions of the apartment's price (Figure 3.4b). This result contradicts the common intuition that transparency enables users to understand when a model will make mistakes.

Prediction error. We define prediction error to be the absolute deviation between the apartment's actual price, a, and the participant's prediction of the apartment's price, u_a—that is, $|a - u_a|$.[11] There is no significant difference between the four primary experimental conditions (Figure 3.3c). However, participants in the baseline condition have significantly higher error than participants in the four primary conditions ($t(1248) = 15.27$, $p < .001$ for the contrast of the baseline with the four primary conditions).

Summary. As predicted, participants who are shown a clear model with a small number of features

[11] Relative deviation between the apartment's actual price and the participant's prediction of the apartment's price ($|a - u_a|/a$) leads to similar trends in all experiments.



(a)

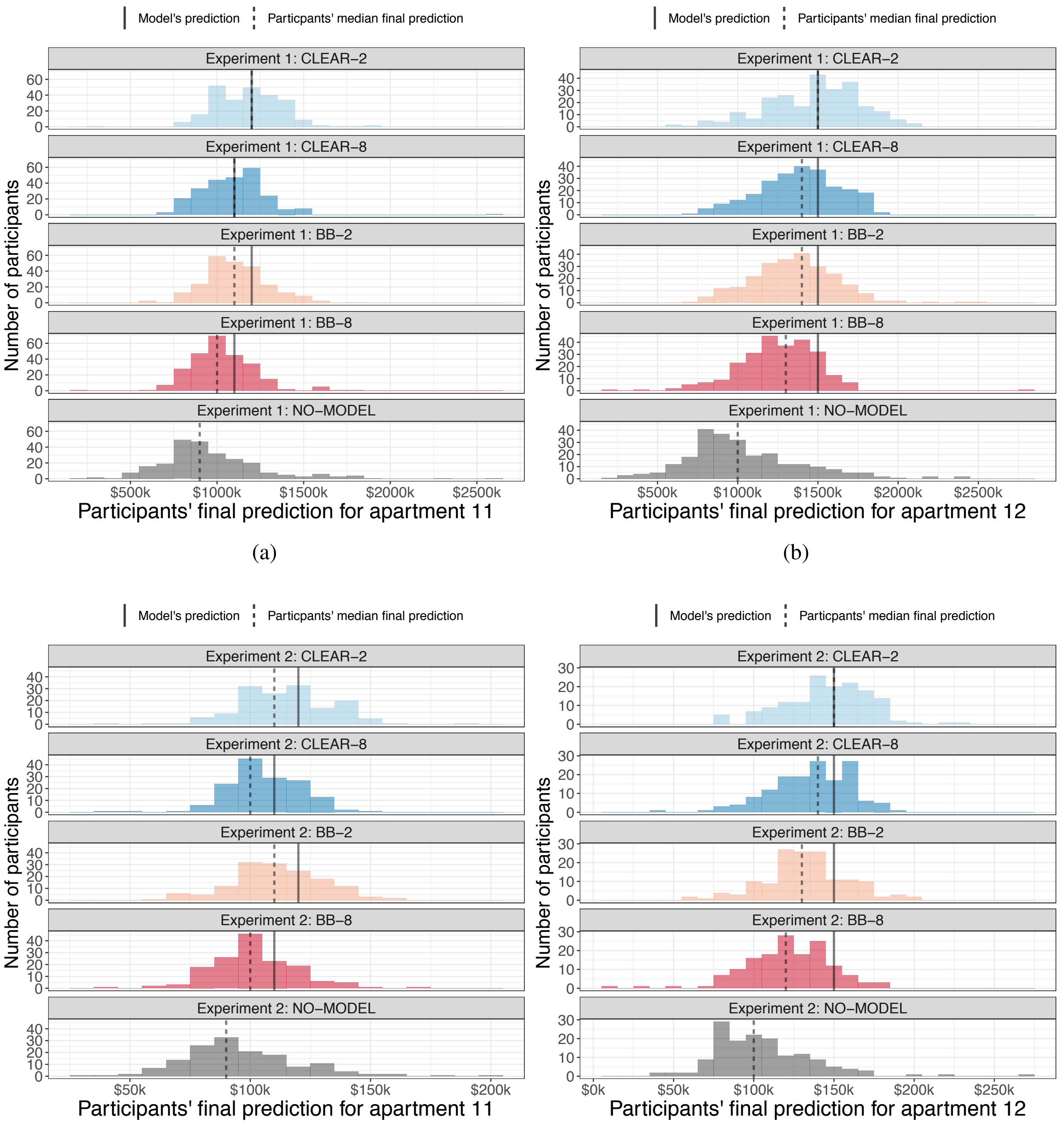
(b)

(c)

(d)

Figure 3.4: Mean deviation from the model for apartments 11 and 12 in our first experiment (top) and in our second experiment (bottom). Error bars indicate one standard error. In both experiments, contrary to intuition, participants who are shown a clear model are less able to correct inaccurate predictions of the model on unusual examples. (Note that for apartment 11, comparisons between the two- and eight-feature conditions are not possible because the models make different predictions.)

are better able to simulate the model's predictions; however, they are not more likely to trust the model's predictions, as indicated by the deviation of their own prediction from the model's prediction. Additionally, contrary to intuition, participants who are shown a clear model are less able to correct inaccurate predictions. Finally, there is no difference in prediction error between the four primary experimental conditions, though participants who have the help of a model are more accurate than those in the baseline condition who do not.



(a)

(b)

(c)

(d)

Figure 3.5: The distribution of participants' final predictions for apartments 11 and 12 in our first experiment (top) and in our second experiment (bottom). Participants in the four primary conditions overestimate the apartment prices compared to the participants in the baseline condition. We suspect that this is due to an anchoring effect around the models' predictions.

3.2 Experiment 2: Scaled-Down Prices

One potential explanation for participants' equal trust in the model across conditions may be their lack of familiarity with New York City's unusually high apartment prices. For example, if a participant finds Upper West Side prices to be unreasonably high, she may not pay attention to how the model works, even when the model is presented as a clear model. Our second experiment is designed to address this issue by replicating our first experiment with apartment prices and maintenance fees scaled down to match median housing prices in the U.S. Before running this experiment we pre-registered three hypotheses.[12] The first two hypotheses (H4 and H5) are identical to H1 and H2 from our first experiment. We make the third hypothesis more precise than H3 to reflect the results of our first experiment and a small pilot with scaled-down prices:

H6. **Detection of mistakes.** Participants will be less likely to correct inaccurate predictions on unusual examples of a clear model compared to a black-box model.

3.2.1 Experimental Design

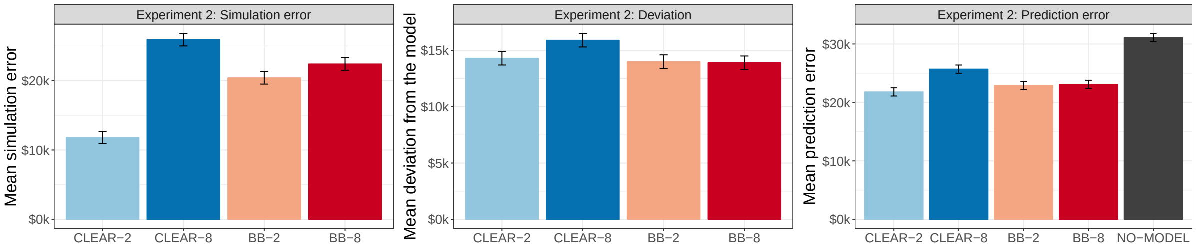
We first scale down the apartment prices and maintenance fees[13] from our first experiment by a factor of ten. To account for this change, we also scale down all regression coefficients (except for the coefficient for maintenance fee) by a factor of ten. Apart from the description of the neighborhood from which the apartments are selected in the task instructions, the experimental design is unchanged. We again run the experiment on Amazon Mechanical Turk. We exclude people who participated in our first experiment, and recruit 750 new participants all of whom satisfy the selection criteria from our first experiment. The participants are randomly assigned to the five conditions (CLEAR-2, $n = 150$; CLEAR-8, $n = 150$; BB-2, $n = 147$; BB-8, $n = 151$; and NO-MODEL, $n = 152$). Each participant received a flat payment of \$2.50.

3.2.2 Results

H4. **Simulation.** As hypothesized, and shown in Figure 3.6a, participants in the CLEAR-2 condition have significantly lower simulation error, on average, than participants in the other conditions ($t(596) =$

[12] Pre-registered hypotheses are available at `https://aspredicted.org/3bv8i.pdf`.

[13] Maintenance fee is a feature of apartments, which is in terms of dollar amounts.



(a) (b) (c)

Figure 3.6: Results from our second experiment: (a) mean simulation error, (b) mean deviation of participants' predictions from the model's predictions (a smaller value indicates higher trust), and (c) mean prediction error. Error bars indicate one standard error. Similar to the first experiment, participants in the CLEAR-2 condition are better able to simulate the model's predictions. Participants trust the CLEAR-2 model and the BB-8 model equally (in terms of deviation from the model). Additionally, participants in the CLEAR-2 condition have statistically but not practically significantly lower prediction error.

10.28, $p < .001$ for the contrast of CLEAR-2 with the other three primary conditions). This is in line with the results for H1 in our first experiment.

H5. **Trust.** Contrary to our second hypothesis, and in line with the results from our first experiment, there is no significant difference in participants' trust, as indicated by their deviation from the model, between CLEAR-2 and BB-8 (Figure 3.6b).

H6. **Detection of mistakes.** In line with the results from our first experiment, there is no significant difference in participants' deviation from the model's prediction between the four primary conditions for apartment 11 (Figure 3.4c). Similar to the first experiment, we observe an anchoring effect around the models' predictions (Figures 3.5c and 3.5d). As hypothesized, and in line with the results from our first experiment, participants in the clear conditions deviate from the model's prediction less, on average, than participants in the black-box conditions for apartment 12, resulting in even worse final predictions of the apartment's price (Figure 3.4d). These results suggest that New York City's unusually high apartment prices do not explain equal trust in models across conditions and participants' poor abilities to correct inaccurate predictions.

Prediction error. Participants in the CLEAR-2 condition have statistically but not practically ($<$ \$3,000) significantly lower prediction error ($t(596) = 3.17$, $p = .001$ for the contrast of CLEAR-2 with the other three primary conditions).

Summary. This experiment confirms our results from the first experiment. Again, participants who are shown a clear model with a small number of features are better able to simulate the model's predictions, though they are not more likely to trust the model's predictions, as measured by the deviation of their own prediction from the model's prediction. Additionally, participants who are shown a clear model are less able to correct inaccurate predictions. In the next section, we dig deeper into trust and design an experiment to measure an alternative metric for trust.

3.3 Experiment 3: Alternative Measure of Trust

The results from our first two experiments demonstrate that participants are no more likely to trust the predictions of a clear model with a small number of features than the predictions of a black-box model with a large number of features, as indicated by the deviation of their own predictions from the model's prediction. However, perhaps another measure of trust would reveal differences between the conditions. In this section, we therefore present our third experiment, which is designed to allow us to compare participants' trust across the conditions using an alternative measure of trust: the **weight of advice** measure frequently used in the literature on advice-taking (Yaniv, 2004; Gino and Moore, 2007; Logg, 2017).

Weight of advice quantifies the degree to which people update their beliefs (e.g., predictions made **before** seeing the model's predictions) toward advice they are given (e.g., the model's predictions). In the context of our experiment, it is defined as $|u_2 - u_1| \,/\, |m - u_1|$, where m is the model's prediction, u_1 is the participant's initial prediction of the apartment's price before seeing m, and u_2 is the participant's final prediction of the apartment's price after seeing m. It is equal to 1 if the participant's final prediction matches the model's prediction and equal to 0.5 if the participant averages their initial prediction and the model's prediction.

To understand the benefits of comparing weight of advice across the conditions, consider the scenario in which u_2 is close to m. There are different reasons why this might happen. On the one hand, it could be the case that u_1 was far from m and the participant made a significant update to their initial prediction based on the model. On the other hand, it could be the case that u_1 was already close to m and the participant did not update her prediction at all. These two scenarios are indistinguishable in terms of the participant's deviation from the model's prediction. In contrast, weight of advice would be high in the first case and low in the second.

We additionally use this experiment as a chance to see whether participants' behavior would differ if they are told that the predictions are made by a "human expert" instead of a model. Previous studies have examined this question from different perspectives with differing results (e.g., Önkal et al., 2009; Dietvorst et al., 2015). Most closely related to our experiment, Logg (2017) find that when people are presented with predictions from either an algorithm or a human expert, they update their own predictions toward predictions from an algorithm more than they do toward predictions from a human expert in a variety of domains. We are interested to see whether this finding replicates.

We pre-registered four hypotheses:[14]

H7. **Trust (deviation).** Participants' predictions will deviate less from the predictions of a clear model with a small number of features than the predictions of a black-box model with a large number of features.

H8. **Trust (weight of advice).** Weight of advice will be higher for participants who see a clear model with a small number of features than for those who see a black-box model with a large number of features.

H9. **Humans vs. machines.** Participants will trust a human expert and a black-box model to differing extents. As a result, their deviation from the model's predictions and their weight of advice will also differ.

H10. **Detection of mistakes.** Participants in different conditions will exhibit varying abilities to correct the model's inaccurate predictions on unusual examples.

The first two hypotheses are variations on H2 from our first experiment, while the last hypothesis is identical to H3.

[14] Pre-registered hypotheses are available at `https://aspredicted.org/795du.pdf`.

3.3.1 Experimental Design

We consider the same four primary experimental conditions as in the first two experiments plus a new condition, EXPERT, in which participants see the same information as in BB-8, but with the black-box model labeled as "Human Expert" instead of "Model." We do not include a baseline condition because the most natural baseline would be to simply ask participants to predict apartment prices (i.e., the first step of the testing phase described below).

We again run the experiment on Amazon Mechanical Turk. We exclude people who participated in our first two experiments, and recruit 1,000 new participants all of whom satisfy the selection criteria from our first two experiments. The participants are randomly assigned to the five conditions (CLEAR-2, $n = 202$; CLEAR-8, $n = 200$; BB-2, $n = 202$; BB-8, $n = 198$; and EXPERT, $n = 197$).[15] Each participant received a flat payment of $1.50.

We ask participants to predict apartment prices for the same set of apartments used in the first two experiments. However, in order to calculate weight of advice, we modify the experiment design so that participants are asked for two predictions for each apartment during the testing phase: an initial prediction before being shown the model's prediction and a final prediction after being shown the model's prediction. To ensure that participants' initial predictions are the same across the conditions, we ask for their initial predictions for all twelve apartments before introducing them to the model or human expert and before informing them that they would be able to update their predictions. This design has the added benefit of potentially reducing the amount of anchoring on the model or expert's predictions.

Participants were first shown detailed instructions (which intentionally did not include any information about the corresponding model or human expert), before proceeding with the experiment in two phases. In the (short) training phase, participants were shown three apartments, asked to predict each apartment's price, and shown the apartment's actual price. The testing phase consisted of two steps. In the first step, participants were shown another twelve apartments. The order of all twelve apartments was randomized. Participants were asked to predict the price of each apartment. In the second step, participants were intro-

[15] We excluded data from one participant who reported technical difficulties.

duced to the model or human expert before revisiting the twelve apartments. As in the first two experiments, the order of the first ten apartments was randomized, while the remaining two (apartments 11 and 12) always appeared last. For each apartment, participants were first reminded of their initial prediction, then shown the model or expert's prediction, and then asked to make their final prediction of the apartment's price.

3.3.2 Results

H7. **Trust (deviation).** Contrary to our first hypothesis, and in line with the results from our first two experiments, there is no significant difference in participants' deviation from the model between CLEAR-2 and BB-8 (Figure 3.7a).

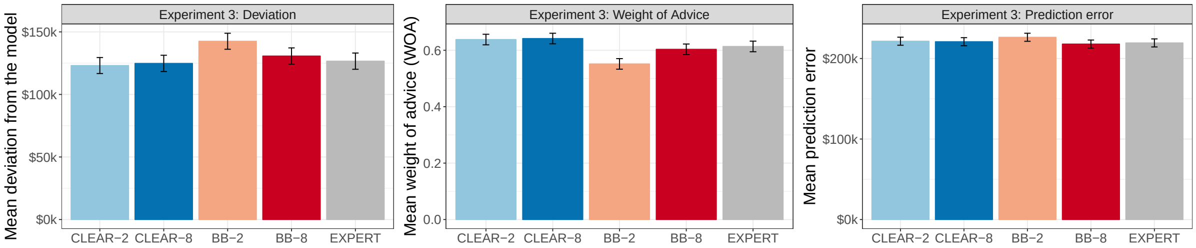
H8. **Trust (weight of advice).** Weight of advice is not well defined when a participant's initial prediction matches the model's prediction (i.e., $u_1 = m$). For each condition, we therefore calculate the mean weight of advice over all participant–apartment pairs for which the participant's initial prediction does not match the model's prediction.[16] This calculation can be viewed as calculating the mean conditioned on there being initial disagreement between the participant and the model. Contrary to our second hypothesis, and in line with the results for the measures of trust in our first two experiments, there is no significant difference in participants' weight of advice between the CLEAR-2 and BB-8 conditions (Figure 3.7b).

H9. **Humans vs. machines.** Contrary to our third hypothesis, there is no significant difference in participants' trust, as indicated by either the deviation of their predictions from the model (Figure 3.7a) or expert's prediction or by their weight of advice (Figure 3.7b), between the BB-8 and EXPERT conditions.

H10. **Detection of mistakes.** In contrast to our first two experiments, there is no significant difference in participants' abilities to correct inaccurate predictions of clear conditions compared to black-box conditions.

Summary. This experiment confirms the results from our first two experiments regarding trust. Again, participants are no more likely to trust the predictions of a clear model with a small number of features than the predictions of a black-box model with a large number of features, this time as indicated by their weight of advice as well as by the deviation of their own predictions from the model's prediction.

[16] There is no significant difference in the fraction of times that participants' initial predictions matched the model's predictions.



(a) (b) (c)

Figure 3.7: Results from our third experiment: (a) mean deviation of participants' predictions from the model's prediction (a smaller value indicates higher trust), (b) mean weight of advice, and (c) mean prediction error. Error bars indicate one standard error. Participants trust the CLEAR-2 model and BB-8 model equally (both in terms of deviation from the model and weight of advice). Additionally, participants trust the BB-8 model and the human expert equally.

Additionally, participants are no more or less likely to trust a human expert than a black-box model. Finally, in contrast to our first two experiments, participants in the clear conditions and black-box conditions are equally able to correct inaccurate predictions on unusual examples.

3.4 Discussion and Future Work

3.4.1 Other Measures of Trust

In this chapter, we consider several ways of measuring trust: deviation of a participant's prediction from the model's, self-reported confidence in the model, and weight of advice. Of course, this list is not exclusive. Another potential measure of trust, also closely related to simulatability, is the extent to which a user learns to mimic a model when making predictions.

In the process of designing the experiment described in Section 3.3, we considered an alternative design in which for each apartment, a participant made a prediction of the apartment's price, saw the model's prediction, and then updated their own prediction, all before moving onto the next apartment. While piloting this design, participants began to change the way in which they made their initial predictions before seeing the model. To test whether participants were in fact updating the way in which they initially made predictions based on the model they observed, we ran a larger version of this study, hypothesizing that the participants' **initial** predictions (before seeing the model) would deviate less from the model's predictions in the CLEAR-2

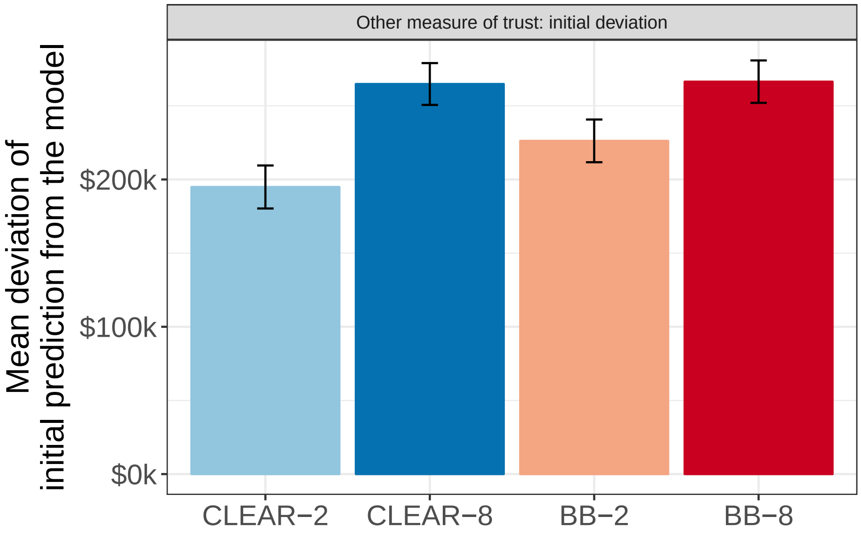


Figure 3.8: Mean deviation of participants' **initial** predictions from the model's predictions. Participants in the CLEAR-2 condition have a significantly lower deviation from the model than participants in other conditions.

condition compared with the others.[17] This was indeed the case ($t(241) = -3.41$, $p < .001$, Figure 3.8).

The general experimental approach that we introduce—i.e., presenting people with models that make identical predictions but varying the presentation of these model in order to isolate and measure the impact of different factors on people's abilities to perform well-defined tasks—can be applied in a wide range of different contexts and may lead to different conclusions in each. For example, instead of a linear regression model, one could examine decision trees or rule lists in a classification setting. Or our experiments can be repeated with participants who are domain experts, data scientists, or researchers in lieu of laypeople recruited on Amazon Mechanical Turk. Likewise, there are many other scenarios to be explored such as debugging a poorly performing model, assessing bias in a model's predictions, or explaining why an individual prediction was made. We hope that our work can serve as a useful template for examining the importance of interpretability in these and other contexts.

[17] Pre-registered hypotheses are available at `https://aspredicted.org/zi8yy.pdf`.

3.5 Summary

In this chapter, we investigated how two factors that are thought to influence a supervised model's interpretability—the number of features and the model transparency—impact laypeople's abilities to simulate a model's predictions, gain trust in a model, and understand when a model will make mistakes. Although a clear model with a small number of features was easier for participants to simulate, there was no difference in trust. Additionally, participants were less able to correct inaccurate predictions when they were shown a clear model instead of a black box. Given these results, one should not take for granted that a "simple" or "transparent" model always leads to higher trust. However, we caution readers against jumping to the conclusion that interpretable models are not valuable. Our experiments focused on just one model, presented to one specific subpopulation, for only a subset of the scenarios in which interpretability might play an important role. Instead, we see this work as the first of many steps towards a larger goal of rigorously quantifying and measuring when and why interpretability matters.

The experiments in this chapter focused on the effect of interpretability on users behavior in a predictive task, where users were asked to make prediction with the help of a **supervised** machine learning model. In our experiments, the interactions that users had with the model was very limited. Existing work shows that when people are given the ability to get involved in building and correcting algorithms, they tend to trust them more (Hu et al., 2014; Hoque and Carenini, 2015; Fails and Olsen Jr, 2003). In the next chapter, we design a framework that allows for richer interactions between humans and machines. We exploit interpretable **unsupervised** models to enable users to interact with, build, and correct algorithms. We evaluate our framework via experiments with humans in the loop.

Chapter 4

ALTO: Active Learning with Topic Overviews for Speeding Label Induction and Document Labeling[1]

Many fields depend on texts labeled by human experts; computational linguistics uses such annotation to determine word senses and sentiment (Kelly and Stone, 1975; Kim and Hovy, 2004); while social science uses "coding" to scale up and systematize content analysis (Budge, 2001; Klingemann et al., 2006). Classification takes these labeled data as a training set and labels new data automatically. Creating a broadly applicable and consistent label set that generalizes well is time-consuming and difficult, requiring expensive annotators to examine large swaths of the data. Effective NLP systems must measure (Hwa, 2004; Osborne and Baldridge, 2004; Ngai and Yarowsky, 2000) and reduce annotation cost (Tomanek et al., 2007). Annotation is hard because it requires both **global** and **local** knowledge of the entire data set. Global knowledge is required to create the set of labels, and local knowledge is required to annotate the most useful documents to serve as a training set for an automatic classifier.

We create a single interface—ALTO (Active Learning with Topic Overviews)—to address both global and local challenges using two machine learning tools: **topic models** (Section 2.3) and **active learning** (we will review active learning in Section 4.1.2). Topic models address the need for annotators to have a **global overview** of the data, exposing the broad themes of the corpus so annotators know what labels to create. Active learning **selects** documents that help the classifier understand the differences between labels and directs the user's attention to them **locally**.

[1] Parts of this chapter was published as: Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Leah Findlater, and Kevin Seppi. Alto: active learning with topic overviews for speeding label induction and document labeling. In Association for Computational Linguistics (ACL), 2016 (Poursabzi-Sangdeh et al., 2016).

After reviewing the challenges associated with labeling in more detail and summarizing existing solutions n Section 4.1, we introduce how we unify topic models and active learning to address the challenges and pitfalls of existing methods (Section 4.2). We then describe our four experimental conditions to compare the effects of providing users with either a topic model or a simple list of documents, with or without active learning suggestions (Section 4.3). Following this section we then describe our data and evaluation metrics (Section 4.4).

Through both synthetic experiments (Section 4.5) and a user study (Section 4.6) with 40 participants, we evaluate ALTO and its constituent components by comparing results from the four conditions introduced above. We first examine user strategies for organizing documents, user satisfaction, and user efficiency. Finally, we evaluate the overall effectiveness of the label set in a post-study crowdsourced task.

4.1 Classification Challenges and Existing Solutions

Machine learning algorithms fall in two primary categories: supervised and unsupervised. While unsupervised methods learn to generalize from a raw and unorganized data set, supervised methods, such as classification, require a **training set** (Chapter 2).

Classification, one of the most commonly used methods in machine learning, is the process of predicting one or more categorical label for unlabeled data points from the training set.[2] Classification is a well-trodden area of machine learning research and has been used with great success for reducing manual human effort and automatizing categorization of data sets in many fields such as computational linguistics (Pang et al., 2002; Navigli, 2009), social science (Hillard et al., 2008), and political science (Grimmer and Stewart, 2013; Paul and Girju, 2010). The difficulty is often **annotation**, i.e., creating the training set, which is usually done by domain experts and requires a significant amount of effort (Hwa, 2004; Osborne and Baldridge, 2004); coding theory is an entire subfield of social science devoted to creating, formulating, and applying labels to text data (Saldana, 2012; Musialek et al., 2016).

Semi-supervised learning (Zhu, 2005) exploits the unlabeled data points for classification and reduces the amount of required human effort in annotation. Additionally, dually supervised methods learn from both

[2] In this thesis, we focus on classification where each data point is associated with one label.

document-level labels and word-level labels (Melville et al., 2009; Settles, 2011). The idea is that because getting labels in the word level is cheaper, faster, and easier, one can learn from these labels to train a classifier with reduced amount of human effort.

Given that the quality of predictions that a classifier makes largely depends on the quality of the training set, ensuring that the data is labeled carefully and has high quality is necessary (Chuang et al., 2014). However, the problem with annotation is usually overlooked and still remains. Perhaps part of the reason is that the annotation problem is not a purely computational problem and it requires interacting with humans. We now review common approaches to overcome the annotation challenges and discuss their pitfalls. Next, we address these pitfalls with a human-in-the-loop approach in the context of document classification.

4.1.1 Crowdsourcing

To ensure quality for the training set, we would ideally turn to domain experts to do the annotation. However, experts are usually expensive and not available. Therefore, researchers usually use on-demand crowdsourcing platforms such as Amazon Mechanical Turk[3] or Figure Eight (previously known as Crowdflower)[4] for annotating their data sets. In these crowdsourcing platforms, requesters post small jobs often referred as Human Intelligence Tasks (HITs) and a specified payment for completing each HIT. Workers can then choose from available HITs and complete them in exchange for the specified payment.

Annotating data points for classification is one of the most common types of HITs that are posted on crowdsourcing platforms. However, given that workers on these platforms are usually non-experts, quality control becomes an issue (Snow et al., 2008). Researchers use several methods to ensure that high quality data are obtained from non-experts. Examples of such methods include requiring a minimum approval rate from the workers to be able to work on the HIT, paying workers based on their performance (Kamar and Horvitz, 2012; Ho et al., 2015; Shah and Zhou, 2016), having multiple workers do the same task and aggregating their answers (Hung et al., 2013; Felt et al., 2015), correcting individual workers' biases (Snow

[3] https://www.mturk.com/

[4] https://www.figure-eight.com/

et al., 2008), and filtering low quality annotations based on annotator level noise, ambiguity, and uncertainty (Hsueh et al., 2009).

Crowdsourcing diminishes the need for domain expert annotators in many scenarios. However, sometimes using crowdsourcing is not feasible, such as cases where privacy is a concern. In such cases, **active learning** can be used to reduce the number of labeled data points, while still achieving a reasonable classifier performance (Settles, 2012). We now provide a review of existing work on active learning.

4.1.2 Active Learning

Active learning methods reduce manual human effort in annotation by directing users' attention to the data points that are most useful to label when training a classifier. These data points are usually the ones that the classifier is most **confused** about their label and the idea is that if the annotator spends their time on labeling these data points, a better classifier can be trained faster.

Active learning provides a systematic way for human annotators to constantly interact with machine learning models to create a training set. Some previous work exists on interfaces that use active learning to iteratively and interactively **query** human annotators for the label of data points. For example, DUALIST is an interactive system that uses active learning on both the documents and features to reduce annotation cost and train a classifier that achieves high accuracy with minimally-labeled data (Settles, 2011).

An important component of active learning is the **querying strategy**. This strategy defines how the next data point to query the annotator is selected. Once the data point is selected, the human annotator is asked to provide a label for it. Then the labeled data point is added to the training set and a new model is trained with the new training set. Repeatedly, the new model selects the next data point to label until we are satisfied with the current training set or we run out of resources (e.g, time). Different querying strategies select data points based on different criterion. We now review some of the most commonly used strategies.

Uncertainty sampling (Lewis and Gale, 1994) is perhaps the most straightforward querying strategy. This strategy selects the data point that the current classifier is most **uncertain** about its label. This approach is well-suited for probabilistic classifiers. Entropy (Shannon, 2001) of the posterior label distribution is commonly used as a measure of uncertainty:

$$d^* = \arg\max_{d} - \sum_{i} P(y_i \mid d)\text{log}P(y_i \mid d), \tag{4.1}$$

where y_i ranges over all the possible labels.

Query-by-committee (Seung et al., 1992) is another commonly used strategy. This strategy selects the data point that a **committee** of classifiers **disagree** the most on its label. Bagging (Breiman, 1996) and boosting (Freund and Schapire, 1997) are commonly used to construct a committee of classifiers (Mamitsuka et al., 1998). Moreover, **vote entropy** is used for measuring the level of disagreement among the committee members.

The advantage of using query-by-committee over uncertainty sampling is that rather than relying on one classifier, multiple classifiers are considered (Freund et al., 1997). However, query-by-committee usually does not scale well and thus might not be well-suited for human-in-the-loop systems.

Other families of querying strategies have been proposed, which perform based on the amount of change in the model (Settles et al., 2008), the amount of generalization error reduction (Roy and McCallum, 2001), or minimization of variance when minimizing error is intractable (Settles, 2012). However, these methods are usually computationally expensive and not well-suited for human-in-the-loop systems.

Discussion

In this section, we reviewed some of the querying strategies used in active learning methods. The choice of the strategy depends on several factors such as domain of the data set, expertise of the annotators, and available resources. The strategies we reviewed have one common assumption: all data points require the same amount of effort to label. Therefore their goal is to minimize the **number** of labeled data points. However, in most real-world scenarios, some data points are harder to label than others. For example, in the context of document labeling, longer documents or documents that have a mixture of topics are usually harder for annotators to label. This brings up the notion of "cost". If the goal of active learning is to reduce the amount of human effort, one might consider the amount of time annotators spend on labeling as the cost rather than the number of data points they label. Thus, active learning methods should be evaluated in the context of cost (Haertel et al., 2008). We take the same approach to develop querying strategies that reduce the **time** annotators spend on labeling in our human-in-the-loop system.

Crowdsourcing and active learning are commonly used to reduce annotation cost. However, these methods can only be applied after a label set exists. In many scenarios such as organizing conferences and legal discovery, there is no pre-defined set of labels and labels should be defined by the annotators. Creating a broad and applicable label set requires a **global** knowledge of the data set, which annotators usually lack when they first start exploring the data set and labeling. In the context of text data, topic models (Chapter 2) are a method to provide an overview of the data set to the users. We now elaborate on our framework that unifies topic overviews and active learning to speed label induction and document labeling.

4.2 Topic Overviews and Active Learning

ALTO,[5] a framework for assigning labels to documents that uses both global and local knowledge to help users create and assign document labels, has two main components: topic **overview** and active learning **selection**. We explain how ALTO uses topic models to aid label induction and document labeling. We then explain how it uses active learning to direct user attention and speed document labeling.

4.2.1 Topic Models

Topic models (Blei et al., 2003) automatically induce structure from a text corpus (Chapter 2). Given a corpus and a constant K for the number of topics, topic models output (1) a distribution over words for each topic k ($\phi_{k,w}$) and (2) a distribution over topics for each document ($\theta_{d,k}$). Each topic's most probable words and associated documents can help a user understand what the collection is about. Table 4.1 shows examples of topics and their highest associated documents from our corpus of U.S. congressional bills.

Our hypothesis is that showing documents grouped by topics will be more effective than having the user wade through an undifferentiated list of random documents and **mentally sort the major themes themselves**.

[5] Code available at `https://github.com/Foroughp/ALTO-ACL-2016`.

Topic words	Document Title
metropolitan, carrier, rail, freight, passenger, driver, airport, traffic, transit, vehicles	A bill to improve the safety of motorcoaches, and for other purposes.
violence, sexual, criminal, assault, offense, victims, domestic, crime, abuse, trafficking	A bill to provide criminal penalties for stalking.
agricultural, farm, agriculture, rural, producer, dairy, crop, producers, commodity, nutrition	To amend the Federal Crop Insurance Act to extend certain supplemental agricultural disaster assistance programs through fiscal year 2017, and for other purposes.

Table 4.1: Given a data set—in this case, the U.S. congressional bills data set—topics are automatically discovered sorted lists of terms that summarize segments of a document collection. Topics also are associated with documents. These topics give users a sense of documents' main themes and help users create high-quality labels.

4.2.2 Active Learning

Active learning directs users' attention to the examples that would be most useful to label when training a classifier. When user time is scarce, active learning builds a more effective training set than random labeling.

In contrast to topic models, active learning provides local information: this is the individual document you should pay attention to. Our hypothesis is that active learning, when used as a **preference function** to direct the users to documents most beneficial to label, will not only be more effective than randomly selecting documents but will also **complement** the global information provided by topic models. Section 4.3.3 describes the preference functions for the experimental conditions.

4.3 Study Conditions

Our goal is to characterize how local and global knowledge can aid users in annotating a data set. This section describes our four experimental conditions and outlines the user's process for labeling documents.

4.3.1 Study Design

The study uses a 2 × 2 between-subjects design, with factors of document collection **overview** (two levels: topic model or list) and document **selection** (two levels: active or random). The four conditions, with the TA condition representing ALTO, are:

		Overview	
		Topic	List
Selection	Active	TA	LA
	Random	TR	LR

Table 4.2: Our 2 × 2 study design and the four experimental conditions. There are two factors: document collection overview and document selection.

(1) Topic model and active selection (TA)

(2) Topic model and random selection (TR)

(3) List and active selection (LA)

(4) List and random selection (LR)

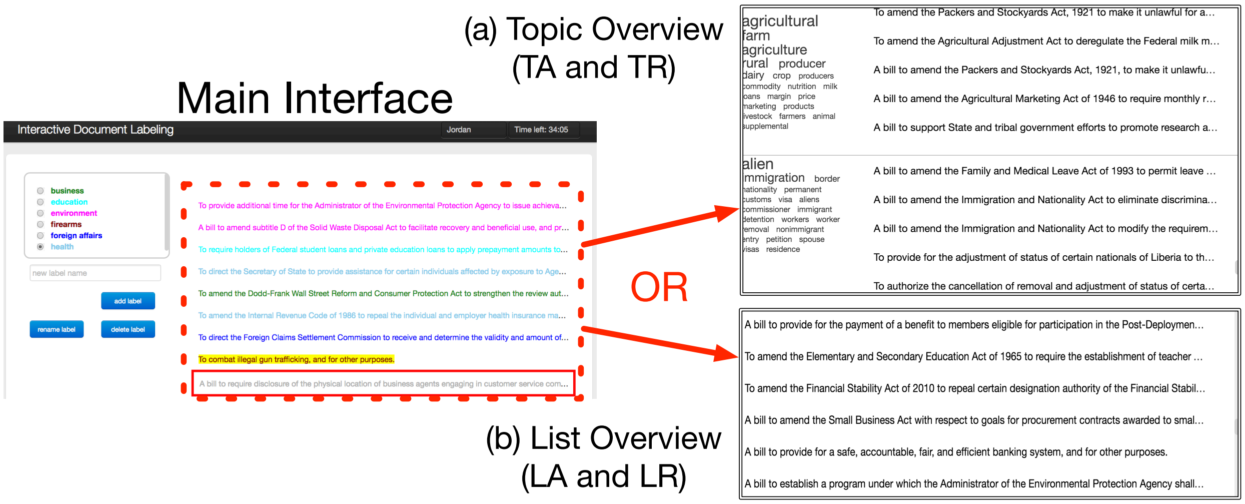
Table 4.2 shows our experimental factors and conditions.

4.3.2 Document Collection Overview

The topic and list overviews offer different overall structure but the same basic elements for users to create, modify, and apply labels (Section 4.3.4). The topic overview (Figure 4.1a) builds on (Hu et al., 2014): for each topic, the top twenty words are shown alongside twenty document titles. Topic words (w) are sized based on their probability $\phi_{k,w}$ in the topic k and the documents with the highest probability of that topic ($\theta_{d,k}$) are shown. The list overview, in contrast, presents documents as a simple, randomly ordered list of titles (Figure 4.1b). We display the same number of documents ($20K$, where K is the total number of topics) in both the topic model and list overviews, but the list overview provides no topic information.

4.3.3 Document Selection

To provide consistency across the four conditions, all of the conditions use a document preference function U to direct the user's attention to a document to label. For the random selection conditions, TR and



To provide additional time for the Administrator of the Environmental Protection Agency to issue achieva…

A bill to amend subtitle D of the Solid Waste Disposal Act to facilitate recovery and beneficial use, and pr…

To require holders of Federal student loans and private education loans to apply prepayment amounts to…

To direct the Secretary of State to provide assistance for certain individuals affected by exposure to Age…

To amend the Dodd-Frank Wall Street Reform and Consumer Protection Act to strengthen the review aut…

To amend the Internal Revenue Code of 1986 to repeal the individual and employer health insurance ma…

To direct the Foreign Claims Settlement Commission to receive and determine the validity and amount of…

To combat illegal gun trafficking, and for other purposes.

A bill to require disclosure of the physical location of business agents engaging in customer service com…

To amend the Packers and Stockyards Act, 1921 to make it unlawful for a…

To amend the Agricultural Adjustment Act to deregulate the Federal milk m…

A bill to amend the Packers and Stockyards Act, 1921, to make it unlawfu…

A bill to amend the Agricultural Marketing Act of 1946 to require monthly r…

A bill to support State and tribal government efforts to promote research a…

A bill to amend the Family and Medical Leave Act of 1993 to permit leave …

A bill to amend the Immigration and Nationality Act to eliminate discrimina…

A bill to amend the Immigration and Nationality Act to modify the requirem…

To provide for the adjustment of status of certain nationals of Liberia to th…

To authorize the cancellation of removal and adjustment of status of certa…

A bill to provide for the payment of a benefit to members eligible for participation in the Post-Deploymen…

To amend the Elementary and Secondary Education Act of 1965 to require the establishment of teacher …

To amend the Financial Stability Act of 2010 to repeal certain designation authority of the Financial Stabil…

A bill to amend the Small Business Act with respect to goals for procurement contracts awarded to smal…

A bill to provide for a safe, accountable, fair, and efficient banking system, and for other purposes.

A bill to establish a program under which the Administrator of the Environmental Protection Agency shall…

Figure 4.1: Our Annotation system in different conditions: Initially, the user sees lists of documents organized in either (a) grouped into topics (only two topics are shown here; users can scroll to additional document) or a (b) list format. The user can click on a document to label it.

LR, document selection is random, within a topic or globally. We expect this to be less useful than active learning. The document preference functions are:

LA: LA uses traditional uncertainty sampling:

$$U_d^{\mathrm{LA}} = \mathbb{H}_C\left[Y_d\right], \tag{4.2}$$

where $\mathbb{H}_C\left[y_d\right] = -\sum_i P(y_i \,|\, d) log P(y_i \,|\, d)$ is the classifier entropy. Entropy measures how confused (uncertain) classifier C is about its prediction of a document d's label y. Intuitively, it prefers documents that most of the labels are likely to be predicted to documents that only one of the labels is highly likely to be chosen.

LR: LR's approach is the same as LA's except we replace $\mathbb{H}_C\left[y_d\right]$ with a uniform random number:

$$U_d^{\mathrm{LR}} \sim \mathrm{unif}(0, 1). \tag{4.3}$$

In contrast to LA, which suggests the most uncertain document, LR suggests a random document.

TA: Dasgupta and Hsu (2008) argue that clustering should inform active learning criteria, balancing coverage against classifier accuracy. We adapt their method to flat topic models—in contrast to their hierarchical cluster trees—by creating a composite measure of document uncertainty within a topic:

$$U_d^{\text{TA}} = \mathbb{H}_C\left[y_d\right]\theta_{d,k}, \tag{4.4}$$

where k is the prominent topic for document d. U_d^{TA} prefers documents that are **representative** of a topic (i.e., have a high value of $\theta_{d,k}$ for that topic) and are informative for the classifier.

TR: TR's approach is the same as TA's except we replace $\mathbb{H}_C\left[Y_d\right]$ with a uniformly random number:

$$U_d^{\text{TR}} = \text{unif}(0, 1)\theta_{d,k}. \tag{4.5}$$

Similar to TA, this prefers documents that are representative of a topic, but not any particular such document. Incorporating the random component encourages for covering different documents in diverse topics.

In LA and LR, the preference function directly chooses a document and directs the user to it. On the other hand, U_d^{TA} and U_d^{TR} are topic dependent: To avoid the negative and misleading effect of topics, users' attention should be drawn to documents that are representative of a specific topic. Therefore, the factor $\theta_{d,k}$ appears in both. Thus, they require that a topic be chosen first and then the document with maximum preference, U, within that topic can be chosen. In TR, the topic is chosen randomly. In TA, the topic is chosen by

$$k^* = \arg\max_k(\text{median}_d(\mathbb{H}_C\left[y_d\right]\theta_{d,k}). \tag{4.6}$$

That is the topic with the maximum median U. Median encodes how "confusing" a topic is.[6] Intuitively, k^* is the topic that the classifier is confused about its documents' labels.

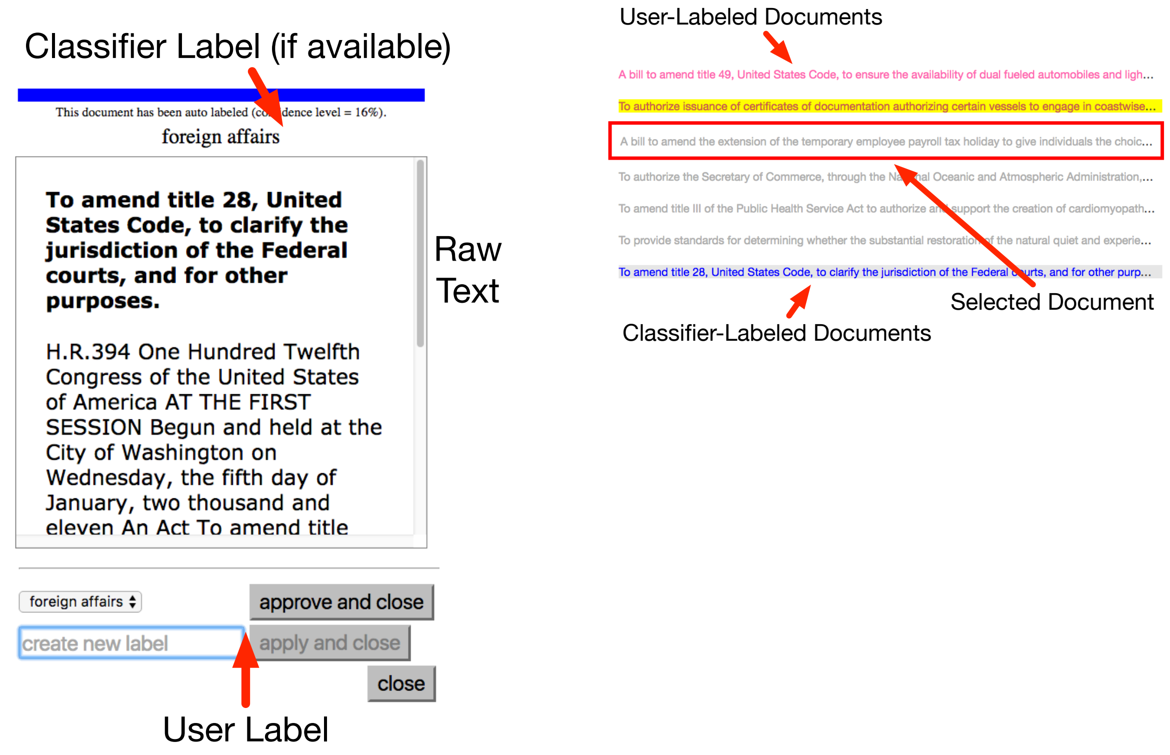
4.3.4 User Labeling Process

The user's labeling process is the same in all four conditions. The **overview** (topic or list) allows users to examine individual documents (Figure 4.1). Clicking on a document opens a dialog box (Figure 4.2a) with the text of the document and three options:

(1) Create and assign a new label to the document.

(2) Choose an existing label for the document.

[6] Outliers skew other measures (e.g., max or mean).



This document has been auto labeled (confidence level = 16%).

foreign affairs

To amend title 28, United States Code, to clarify the jurisdiction of the Federal courts, and for other purposes.

H.R.394 One Hundred Twelfth Congress of the United States of America AT THE FIRST SESSION Begun and held at the City of Washington on Wednesday, the fifth day of January, two thousand and eleven An Act To amend title

A bill to amend title 49, United States Code, to ensure the availability of dual fueled automobiles and ligh...

To authorize issuance of certificates of documentation authorizing certain vessels to engage in coastwise...

A bill to amend the extension of the temporary employee payroll tax holiday to give individuals the choic...

To authorize the Secretary of Commerce, through the National Oceanic and Atmospheric Administration,...

To amend title III of the Public Health Service Act to authorize and support the creation of cardiomyopath...

To provide standards for determining whether the substantial restoration of the natural quiet and experie...

To amend title 28, United States Code, to clarify the jurisdiction of the Federal courts, and for other purp...

(a) Document view: after clicking on a document from the list or topic overview, the user inspects the text and provides a label. If the classifier has a guess at the label, the user can confirm the guess.

(b) After the user has labeled some documents, the system can automatically label other documents and select which documents would be most helpful to annotate next. In the random selection setting, random documents are selected.

Figure 4.2: Document view (a) and document selection (b) in our annotation system.

(3) Skip the document.

Once the user has labeled two documents with different labels, the displayed documents are replaced based on the preference function (Section 4.3.3), every time the user labels (or updates labels for) a document. In TA and TR, each topic's documents are replaced with the twenty highest ranked documents. In LA and LR, all documents are updated with the top $20K$ ranked documents.[7]

The system also suggests one document to consider by auto-scrolling to it and drawing a red box around its title (Figure 4.2b). The user may ignore that document and click on any other document. After the

[7] In all conditions, the number of displayed unlabeled documents is adjusted based on the number of manually labeled documents. i.e. if the user has labeled n documents in topic k, n manually labeled documents followed by top $20 - n$ uncertain documents will be shown in topic k.

user labels ten documents, the classifier runs and assigns labels to other documents.[8] For classifier-labeled documents, the user can either approve the label or assign a different label. The process continues until the user is satisfied or a time runs out (forty minutes in our user study, Section 4.6). We use time to control for the varying difficulty of assigning documents: active learning will select more difficult documents to annotate, but they may be more useful; time is a more fair basis of comparison in real-world tasks.

4.4 Data and Evaluation Metrics

In this section, we describe our data, the machine learning techniques to learn classifiers from examples, and the evaluation metrics to know whether the final labeling of the complete documents collection was successful.

4.4.1 Data sets

Our experiments require corpora to compare user labels with gold standard labels. We experiment with two corpora: 20 Newsgroups (Lang, 2007) and U.S. congressional bills from GovTrack.[9]

For U.S. congressional bills, GovTrack provides bill information such as the title and text, while the Congressional Bills Project (Adler and Wilkerson, 2006) provides labels and sub-labels for the bills. Examples of labels are agriculture and health, while sub-labels include agricultural trade and comprehensive health care reform. The twenty top-level labels have been developed by consensus over many years by a team of top political scientists to create a reliable, robust data set. We use the 112th Congress; after filtering,[10] this data set has 5558 documents. We use this data set in both the synthetic experiments (Section 4.5) and the user study (Section 4.6).

The 20 Newsgroups corpus has $19,997$ documents grouped in twenty news groups that are further grouped into six more general topics. Examples are talk.politics.guns and sci.electronics, which belong to the general topics of politics and science. We use this data set in synthetic experiments (Section 4.5).

[8] To reduce user confusion, for each existing label, only the top 100 documents get a label assigned in the UI.

[9] `https://www.govtrack.us/`

[10] We remove bills that have less than fifty words, no assigned gold label, duplicate titles, or have the gold label GOVERNMENT OPERATIONS or SOCIAL WELFARE, which are broad and difficult for users to label.

4.4.2 Machine Learning Techniques

Topic Modeling:

To choose the number of topics (K), we calculate average topic coherence (Lau et al., 2014) (Equation 2.1 in Chapter 2) on U.S. Congressional Bills, between ten and forty topics and choose $K = 19$, as it has the maximum coherence score. For consistency, we use the same number of topics ($K = 19$) for the 20 Newsgroups corpus. After filtering words based on TF-IDF (Equation 2.2), we use Mallet (McCallum, 2002) with default options to learn topics.

Features and Classification:

A logistic regression classifier predicts labels for documents and provides the classification uncertainty for active learning. To make classification and active learning updates efficient, we use incremental learning (Carpenter, 2008, LingPipe). We update classification parameters using stochastic gradient descent, restarting with the previously learned parameters as new labeled documents become available.[11] We use cross validation, based on the prominent topic as the label for each document, to set the parameters for learning the classifier.[12]

The features for classification include topic probabilities, unigrams, and the fraction of labeled documents in each document's prominent topic. The intuition behind adding this last feature is to allow active learning to suggest documents in a diverse range of topics if it finds this feature a useful indicator of uncertainty.[13]

4.4.3 Evaluation Metrics

Our goal is to create a system that allows users to quickly induce a high-quality label set. We compare the user-created label sets against the data's gold label sets. Comparing different clusterings is a difficult task, so we use three clustering evaluation metrics: purity (Zhao and Karypis, 2001), rand index (Rand,

[11] Exceptions are when a new label is added, a document's label is deleted, or a label is deleted. In those cases, we train the classifier from scratch. Also, for final results in Section 4.6, we train a classifier from scratch.

[12] We use `blockSize` = 1/#examples, `minEpochs` = 100, `learningRate` = 0.1, `minImprovement` = 0.01, `maxEpochs` = 1000, and `rollingAverageSize` = 5. The regression is unregularized.

[13] However, the final classifier's coefficients suggested that this feature did not have a large effect.

1971, RI), and normalized mutual information (Strehl and Ghosh, 2003, NMI).[14]

Purity: Purity measures how "pure" user clusters are compared to gold clusters. Given each user cluster, it measures what fraction of the documents in a user cluster belong to the most frequent gold label in that cluster:

$$\text{purity}(\mathbf{U}, \mathbf{G}) = \frac{1}{N} \sum_l \max_j |U_l \cap G_j|, \tag{4.7}$$

where L is the number of labels the user creates, $\mathbf{U} = \{U_1, U_2, \dots, U_L\}$ is the user clustering of documents, $\mathbf{G} = \{G_1, G_2, \dots, G_J\}$ is the gold clustering of documents, and N is the total number of documents. The user U_l and gold G_j labels are interpreted as sets containing all documents assigned to that label.

Rand index (RI): RI is a **pair counting** measure, where cluster evaluation is considered as a series of decisions. If two documents have the same gold label and the same user label (TP), and if they do not have the same gold label and are not assigned the same user label (TN), the decision is right. Otherwise, it is wrong (FP and FN). RI measures the percentage of decisions that are right:

$$\text{RI} = \frac{TP + TN}{TP + FP + TN + FN}. \tag{4.8}$$

Normalized mutual information (NMI): NMI is an **information theoretic** measure that measures the amount of information one gets about the gold clusters by knowing what the user clusters are:

$$\text{NMI}(\mathbf{U}, \mathbf{G}) = \frac{2\mathbb{I}(\mathbf{U}, \mathbf{G})}{\mathbb{H}_\mathbf{U} + \mathbb{H}_\mathbf{G}}, \tag{4.9}$$

where $\mathbf{U}$ and $\mathbf{G}$ are user and gold clusters, $\mathbb{H}$ is the entropy and $\mathbb{I}$ is mutual information (Bouma, 2009).

While purity, RI, and NMI are all normalized within $[0, 1]$ (higher is better), they measure different things. Purity measures the intersection between two clusterings, is sensitive to the number of clusters, and is not symmetric.

On the other hand, RI and NMI are less sensitive to the number of clusters and are symmetric. RI measures pairwise agreement in contrast to purity that directly measures intersection. Moreover, NMI measures shared information between two clusterings.

[14] We avoided using adjusted rand index (Hubert and Arabie, 1985), because it can yield negative values, which is not consistent with purity and NMI. We also computed variation of information (Meilă, 2003) and normalized information distance (Vitányi et al., 2009) and observed consistent trends. We omit these results for the sake of space.

None of these metrics are perfect: purity can be exploited by putting each document in its own label, RI does not distinguish separating similar documents with distinct labels from giving dissimilar documents the same label, and NMI's ability to compare different numbers of clusters means that it sometimes gives high scores for clusterings by chance. Given the diverse nature of these metrics, if a labeling does well in all three of them, we can be relatively confident that it is not a degenerate solution that games the system.

4.5 Synthetic Experiments

Before running a user study, we test our hypothesis that topic model overviews and active learning selection improve final cluster quality compared to standard baselines: list overview and random selection. We simulate the four conditions on Congressional Bills and 20 Newsgroups.

Since we believe annotators create more specific labels compared to the gold labels, we use sub-labels as simulated user labels and labels as gold labels (we give examples of labels and sub-labels in Section 4.4.1). We start with two randomly selected documents that have different sub-labels, assign the corresponding sub-labels, then add more labels based on each condition's preference function (Section 4.3.3). We follow the condition's preference function and incrementally add labels until 100 documents have been labeled (100 documents are representative of what a human can label in about an hour). Given these labels, we compute purity, RI, and NMI over time. This procedure is repeated fifteen times (to account for the randomness of initial document selections and the preference functions with randomness).[15]

Synthetic results validate our hypothesis that topic overview and active learning selection can help label a corpus more efficiently (Figure 4.3). LA shows early gains, but tends to falter eventually compared to both topic overview and topic overview combined with active learning selection (TR and TA).

However, these experiments do not validate ALTO. Not all documents require the same time or effort to label, and active learning focuses on the hardest examples, which may confuse users. Thus, we need to evaluate how effectively actual users annotate a collection's documents.

[15] Synthetic experiment data is available at `http://github.com/Pinafore/publications/tree/master/2016_acl_doclabel/data/synthetic_exp`.

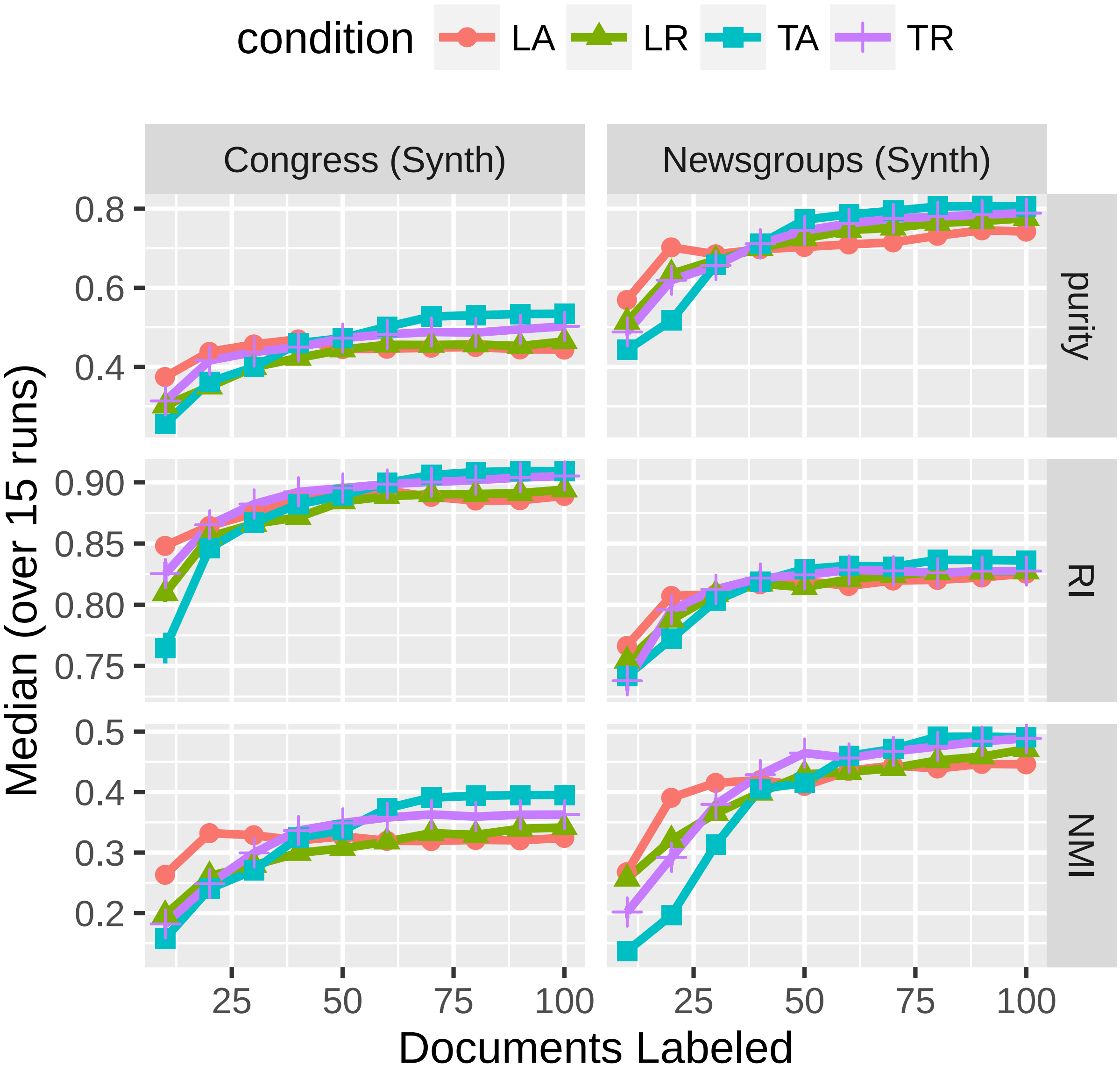


Figure 4.3: Synthetic results on U.S. Congressional Bills and 20 Newsgroups data sets. Topic models help guide annotation attention to diverse segments of the data.

4.6 User Study

Following the synthetic experiments, we conduct a user study with forty participants to evaluate ALTO (TA condition) against three alternatives that lack topic overview (LA), active learning selection (TR), or both (LR) (Sections 4.6.1 and 4.6.2). Then, we conduct a crowdsourced study to compare the overall effectiveness

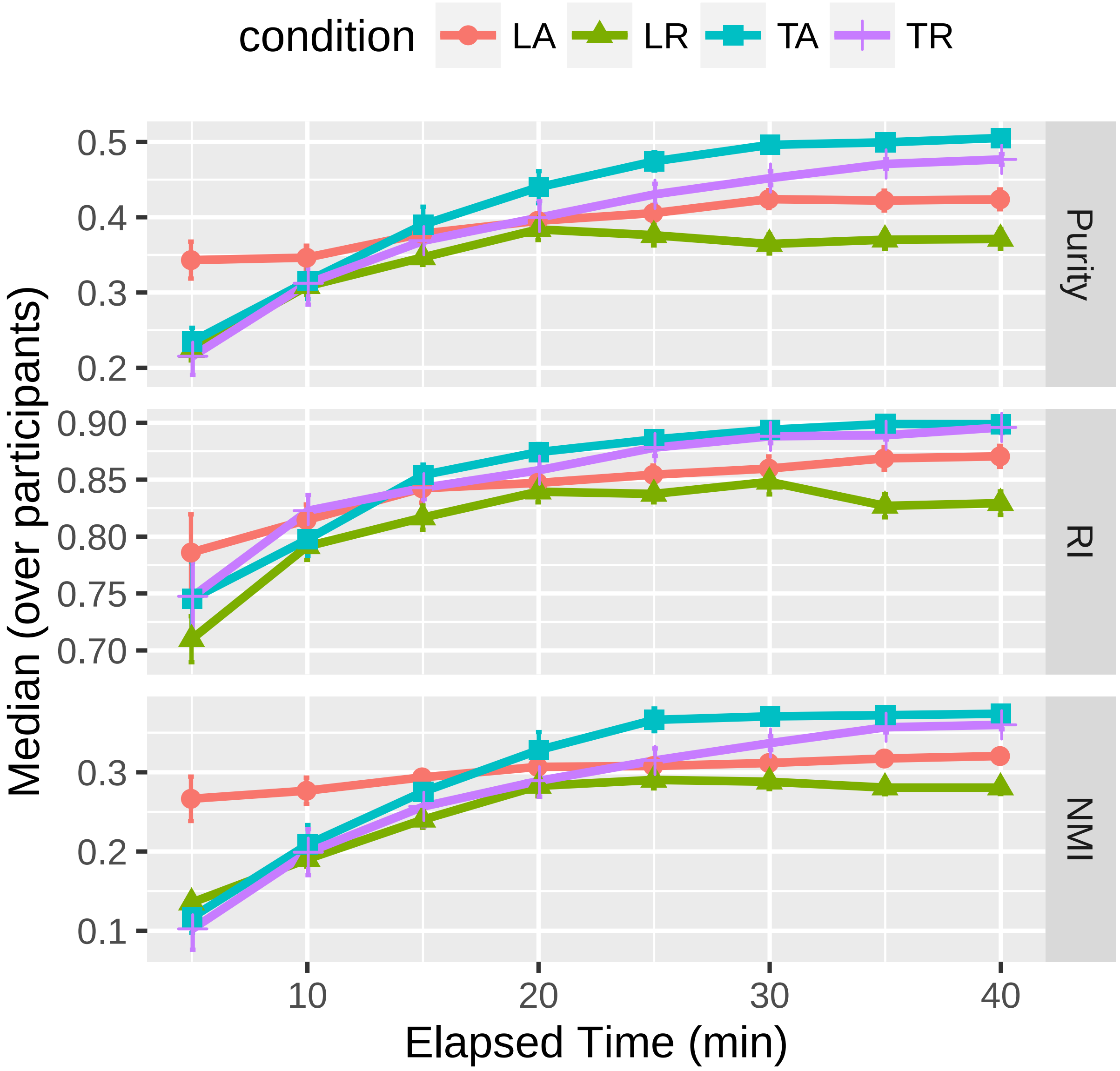


Figure 4.4: User study results on U.S. Congressional Bills data set. Active learning selection helps initially, but the combination of active learning selection and topic model overview has highest quality labels by the end of the task.

of the label set generated by the participants in the four conditions (Section 4.6.5).

4.6.1 Method

We use the freelance marketplace Upwork[16] to recruit online participants. Communicating with participants and instructing them on a specific task is usually easier on Upwork compared to crowdsourcing platforms such as Amazon Mechanical Turk or Figure Eight (previously known as Crowdflower). We require participants to have more than 90% job success on Upwork, English fluency, and U.S. residency. Participants are randomly assigned to one of the four conditions and we recruited ten participants per condition.

Participants completed a demographic questionnaire (Appendix B.1), viewed a video of task instructions, and then interacted with the system and labeled documents until satisfied with the labels or forty minutes had elapsed.[17] The session ended with a survey (Appendix B.2), where participants rated mental, physical, and temporal demand, and performance, effort, and frustration on 20-point scales, using questions adapted from the NASA Task Load Index (Hart and Staveland, 1988, TLX). The survey also included 7-point scales for ease of coming up with labels, usefulness and satisfaction with the system, and—for TR and TA—topic information helpfulness. Each participant was paid fifteen dollars.[18]

For statistical analysis, we primarily use 2×2 (**overview** $\times$ **selection**) ANOVAs with Aligned Rank Transform (Wobbrock et al., 2011, ART), which is a non-parametric alternative to a standard ANOVA that is appropriate when data are not expected to meet the normality assumption of ANOVA.

4.6.2 Document Cluster Evaluation

We analyze the data by dividing the forty-minute labeling task into five minute intervals. If a participant stopped before the time limit, we consider their final data set to stay the same for any remaining intervals. Figure 4.4 shows the measures across study conditions, with similar trends for all three measures.

[16] `http://Upwork.com`

[17] Forty minutes of activity, excluding system time to classify and update documents. Participants nearly exhausted the time: 39.3 average minutes in TA, 38.8 in TR, 40.0 in LA, and 35.9 in LR.

[18] User study data is available at `http://github.com/Pinafore/publications/tree/master/2016_acl_doclabel/data/user_exp`.

	F		p	
	Overview	Selection	Overview	Selection
final purity	81.03	7.18	$< .001$	.011
final RI	39.89	6.28	$< .001$	.017
final NMI	70.92	9.87	$< .001$	.003

df(1,36) for all reported results

Table 4.3: Results from 2×2 ANOVA with ART analyses on the final purity, RI, and NMI metrics. Only main effects for the factors of **overview** and **selection** are shown; no interaction effects were statistically significant. Topics and active learning both had significant effects on quality scores.

Topic model overview and active learning both significantly improve final data set measures.

The topic overview and active selection conditions significantly outperform the list overview and random selection, respectively, on the final label quality metrics. Table 4.3 shows the results of separate 2×2 ANOVAs with ART with each of final purity, RI, and NMI scores. There are significant main effects of **overview** and **selection** on all three metrics; no interaction effects were significant.

TR outperforms LA.

Topic models by themselves outperform traditional active learning strategies (Figure 4.4). LA performed better than LR; while active learning was useful, it was not as useful as the topic model overview (TR and TA).

LA provides an initial benefit.

Average purity, NMI and RI were highest with LA for the earliest labeling time intervals. Thus, when time is very limited, using traditional active learning (LA) is preferable to topic overviews; users need time to explore the topics and a subset of documents within them. Table 4.4 shows the metrics after ten minutes. Separate 2×2 ANOVAs with ART on the means of purity, NMI and RI revealed a significant interaction effect between **overview** and **selection** on mean NMI ($F(1, 36) = 5.58$, p = .024), confirming the early performance trends seen in Figure 4.4 at least for NMI. No other main or interaction effects were significant, likely due to low statistical power.

	$M \pm SD\,[median]$		
	purity	RI	NMI
TA	$0.31 \pm 0.08\,[0.32]$	$0.80 \pm 0.05\,[0.80]$	$0.19 \pm 0.08\,[0.21]$
TR	$0.32 \pm 0.09\,[0.31]$	$0.82 \pm 0.04\,[0.82]$	$0.21 \pm 0.09\,[0.20]$
LA	$0.35 \pm 0.05\,[0.35]$	$0.82 \pm 0.04\,[0.81]$	$0.27 \pm 0.05\,[0.28]$
LR	$0.31 \pm 0.04\,[0.31]$	$0.79 \pm 0.04\,[0.79]$	$0.19 \pm 0.03\,[0.19]$

Table 4.4: Mean, standard deviation, and median purity, RI, and NMI after ten minutes. NMI in particular shows the benefit of LA over other conditions at early time intervals.

			$M \pm SD\,[median]$			
Condition	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
TA	$9.8 \pm 5.6\,[10]$	$2.9 \pm 3.4\,[2]$	$9 \pm 7.8\,[7]$	$5.5 \pm 5.8\,[1.5]$	$9.4 \pm 6.3\,[10]$	$4.5 \pm 5.5\,[1.5]$
TR	$10.6 \pm 4.5\,[11]$	$2.4 \pm 2.8\,[1]$	$7.4 \pm 4.1\,[9]$	$8.8 \pm 6.1\,[7.5]$	$9.8 \pm 3.7\,[10]$	$3.9 \pm 3.0\,[3.5]$
LA	$9.1 \pm 5.5\,[10]$	$1.7 \pm 1.3\,[1]$	$10.2 \pm 4.8\,[11]$	$8.6 \pm 5.3\,[10]$	$10.7 \pm 6.2\,[12.5]$	$6.7 \pm 5.1\,[5.5]$
LR	$9.8 \pm 6.1\,[10]$	$3.3 \pm 2.9\,[2]$	$9.3 \pm 5.7\,[10]$	$9.4 \pm 5.6\,[10]$	$9.4 \pm 6.2\,[10]$	$7.9 \pm 5.4\,[8]$

Table 4.5: Mean, standard deviation, and median results from NASA-TLX post-survey. All questions are scaled 1 (low)–20 (high), except performance, which is scaled 1 (good)–20 (poor). Users found topic model overview conditions, TR and TA, to be significantly less frustrating than the list overview conditions.

4.6.3 Subjective Ratings

Table 4.5 shows the average scores given for the six NASA-TLX questions in different conditions. Separate 2×2 ANOVA with ART for each of the measures revealed only one significant result: participants who used the topic model overview find the task to be significantly less frustrating ($M = 4.2$ and $median = 2$) than those who used the list overview ($M = 7.3$ and $median = 6.5$) on a scale from 1 (low frustration) to 20 (high frustration) ($F(1, 36) = 4.43$, $p = .042$), confirming that the topic overview helps users organize their thoughts and experience less stress during labeling.

Participants in the TA and TR conditions rate topic information to be useful in completing the task ($M = 5.0$ and $median = 5$) on a scale from 1 (not useful at all) to 7 (very useful). Overall, users were positive about their experience with the system. Participants in all conditions rated overall satisfaction with the interface positively ($M = 5.8$ and $median = 6$) on a scale from 1 (not satisfied at all) to 7 (very satisfied).

Topic Words	Automatic Label
metropolitan, carrier, rail, freight, passenger, driver, airport, traffic, transit, vehicles	Rail transport
violence, sexual, criminal, assault, offense, victims, domestic, crime, abuse, trafficking	Sexual violence
agricultural, farm, agriculture, rural, producer, dairy, crop, producers, commodity, nutrition	Dairy farming
academic, youth, elementary, learning, teachers, language, literacy, subpart, early, workforce	Education

Table 4.6: Topic words and their automatically generated label.

4.6.4 Discussion

One can argue that using topic overviews for labeling could have a negative effect: users may ignore the document content and focus on topics for labeling. We tried to avoid this issue by making it clear in the instructions that they need to focus on document content and use topics as a guidance. On average, the participants in TR created 1.96 labels per topic and the participants in TA created 2.26 labels per topic. This suggests that participants are going beyond what they see in topics for labeling, at least in the TA condition.

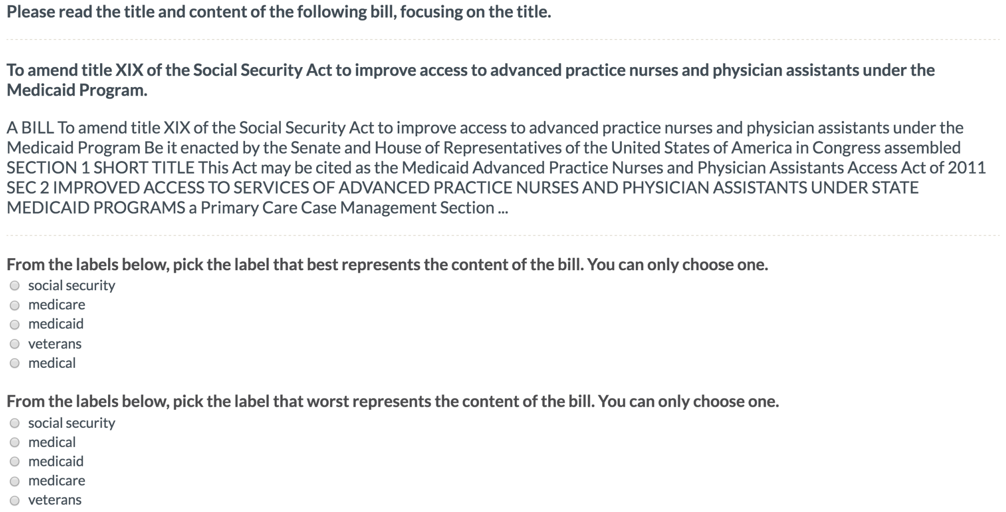
4.6.5 Label Evaluation Results

Section 4.6.2 compares clusters of documents in different conditions against the gold clustering but does not evaluate the quality of the labels themselves. Since one of the main contributions of ALTO is to accelerate the induction of a high quality label set, we use crowdsourcing to assess how the final induced label sets compare in different conditions.

For completeness, we also compare labels against a fully automatic labeling method (Aletras and Stevenson, 2014, Chapter 2) that does not require human intervention. We assign **automatic** labels to documents based on their most prominent topic. Table 4.6 show examples of topics and their automatic label.

We ask users on a crowdsourcing platform to **vote** for the "best" and "worst" label that describes the content of a U.S. congressional bill (we use Crowdflower[19] and require contributors to be in the U.S.).

[19] Crowdflower is now known as Figure Eight.



Please read the title and content of the following bill, focusing on the title.

To amend title XIX of the Social Security Act to improve access to advanced practice nurses and physician assistants under the Medicaid Program.

A BILL To amend title XIX of the Social Security Act to improve access to advanced practice nurses and physician assistants under the Medicaid Program Be it enacted by the Senate and House of Representatives of the United States of America in Congress assembled SECTION 1 SHORT TITLE This Act may be cited as the Medicaid Advanced Practice Nurses and Physician Assistants Access Act of 2011 SEC 2 IMPROVED ACCESS TO SERVICES OF ADVANCED PRACTICE NURSES AND PHYSICIAN ASSISTANTS UNDER STATE MEDICAID PROGRAMS a Primary Care Case Management Section ...

From the labels below, pick the label that best represents the content of the bill. You can only choose one.

○ social security
○ medicare
○ medicaid
○ veterans
○ medical

From the labels below, pick the label that worst represents the content of the bill. You can only choose one.

○ social security
○ medical
○ medicaid
○ medicare
○ veterans

Figure 4.5: Example best and worst label annotation task. Workers choose which labels best and worst describe a congressional bill.

Document title	Auto	LA	LR	TA	TR	Best	Worst
To amend title XIX of the Social Security Act to improve access to advanced practice nurses and physician assistants under the Medicaid Program.	medicaid	social security	veterans	medicare	medical	medical	veterans
A bill to authorize the use of certain offshore oil and gas platforms in the Gulf of Mexico for artificial reefs, and for other purposes.	coast	endangered	environment	coastal restoration	natural resources	natural resources	endangered
To establish pilot programs to encourage the use of shared appreciation mortgage modifications, and for other purposes.	mortgage bank	banking and finance	banking and finance	mortgage	securities regulation	mortgage	securities regulation
To amend the Clean Air Act to conform the definition of renewable biomass to the definition given the term in the Farm Security and Rural Investment Act of 2002.	dairy farming	agriculture	energy	renewable fuel	agriculture	renewable fuel	dairy farming

Table 4.7: Examples of documents, their assigned labels in different conditions, and the chosen best and worst labels.

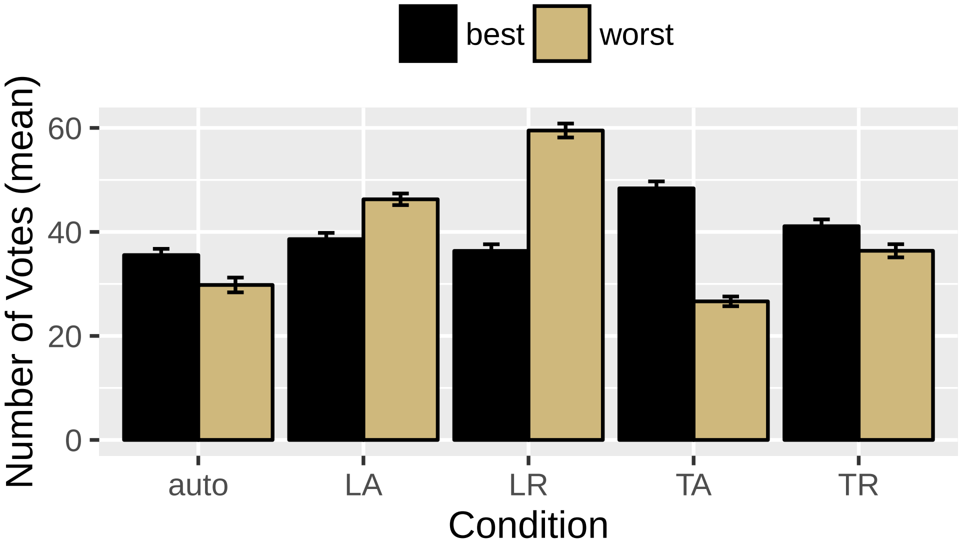


Figure 4.6: Best and worst votes for document labels. Error bars are standard error from bootstrap sample. ALTO (TA) gets the most best votes and the fewest worst votes.

Figure 4.5 shows an example of the task on Crowdflower. Table 4.7 shows examples of documents, labels in different conditions, and the best and the worst chosen labels.

Five users label each document and we use the aggregated results generated by Crowdflower. The user gets $0.20 for each task.

We randomly choose 200 documents from our data set (Section 4.4.1). For each chosen document, we randomly choose a participant from all four conditions (TA, TR, LA, LR). The labels assigned in different conditions and the automatic label of the document's prominent topic construct the candidate labels for the document.[20] Identical labels are merged into one label to avoid showing duplicate labels to users. If a merged label gets a "best" or "worst" vote, we split that vote across all the identical instances.[21] Figure 4.6 shows the average number of "best" and "worst" votes for each condition and the automatic method. ALTO (TA) receives the most "best" votes and the fewest "worst" votes. LR receive the most worst votes. The

[20] Some participants had typos in the labels. We corrected all the typos using pyEnchant (http://pythonhosted.org/pyenchant/) spellchecker. If the corrected label was still wrong, we corrected it manually.

[21] Evaluation data is available at http://github.com/Pinafore/publications/tree/master/2016_acl_doclabel/data/label_eval.

automatic labels, interestingly, appear to do at least as well as the list view labels, with a similar number of best votes and fewer worst votes. This indicates that automatic labels have reasonable quality compared to at least some manually generated labels. However, when users are provided with topic model overview, with or without active learning selection, they can generate label sets that improve upon automatic labels and labels assigned without the topic model overview.

4.7 Related Work

ALTO quantitatively shows that corpus overviews aid text understanding, building on traditional interfaces for gaining both local and global information (Hearst and Pedersen, 1996). More elaborate interfaces provide richer information given a fixed topic model (Chapter 2). Because topic models are imperfect (Boyd-Graber et al., 2014), enabling users to interact with topic models and refine the underlying topics can potentially improve users' understanding of a corpus. This direction has a close connection with bridging the gap between users and models (Chapter 1) and has been approached by focusing on systematically studying the effect of unpredictability in topic refinements on users' experience and proposing potential solutions to improve users' trust (Smith et al., 2016; Lee et al., 2017; Smith et al., 2018).

Another related direction of research focuses on representation of individual topics. Summarizing document collections through discovered topics can happen through raw topics labeled manually by users (Talley et al., 2011), automatically (Mei et al., 2007; Magatti et al., 2009; Lau et al., 2010, 2011; Hulpus et al., 2013; Aletras and Stevenson, 2014; Wan and Wang, 2016), or by learning a mapping from labels to topics (Ramage et al., 2009). Depending on the target end users and specific scenarios, these alternative visualizations and representations can be helpful.

When there is not a direct correspondence between topics and labels, supervised topic models jointly model document content and labels to learn topics that are in line with labels (Mcauliffe and Blei, 2007; Zhu et al., 2009; Nguyen et al., 2015). In this study, because we wanted topics to be consistent between users, we used static topics. However, we believe ALTO can be extended to use supervised topic models that dynamically update topics and provide an overview of the corpus that is more in line with user labels.

4.8 Summary

We introduced ALTO, an interactive framework that combines active learning **selections** with topic **overviews** to both help users induce a label set and assign labels to documents. We showed that users can more effectively and efficiently induce a label set and create training data using ALTO compared to other conditions, which lack either topic **overviews** or active **selections**.

ALTO exemplifies an interactive framework that brings humans and machine learning models together to guide humans in completing a classification task by exploiting interpretable unsupervised models. We used a task-driven approach to evaluate ALTO with humans in the loop. Our user study results provide insights on how to make the best use of user effort under different scenarios and time constraints. In the next chapter, we turn to a real-world use case that requires exploring and understanding large document collections. We design a framework to help humans understand **science policy** and find answers to a set of questions from a large data set of documents. We evaluate the effectiveness of a set of machine learning and information retrieval tools with human-subject experiments.

Chapter 5

Understanding Science Policy via a Human-in-the-Loop Approach[1]

In Chapter 4, we demonstrated an interactive system that uses topic models to help humans induce label sets, assign them to documents, and create the training set for a classifier. Several interfaces have been proposed that use topic models to help humans navigate through large document collections, understand them, and find documents of interest, some of which we review in Section 2.4. However, there have been no systematic studies to assess the effectiveness of such tools on people's abilities in completing real-world tasks that require understanding large corpora. In this chapter, we turn to a real-world use case that requires humans to interactively explore and understand large document collections and complete a task with the knowledge they have gained. We use a task-driven approach to evaluate the effectiveness of a set of ML tools.

We focus on **understanding science policy** as a use case that requires understanding large data sets. Science policy is concerned with allocation of funding and resources for doing scientific research and experiments (Archibugi and Filippetti, 2015). Common topics of interest in science policy include funding scientific research, transferring research into technological innovation, and promoting product development. Understanding science policy requires making sense of the large data set of funded scientific research documents. This use case is inspired by the questions that the funding agency representatives are frequently asked on the details of the research they are funding. These questions are usually about the amount of money that is awarded in a specific area of research, the research topics that are funded, and the future plans in a research

[1] The work in this chapter was done in collaboration with You Lu, Leah Findlater, Kevin Seppi, and Jordan Boyd-Graber and is in submission.

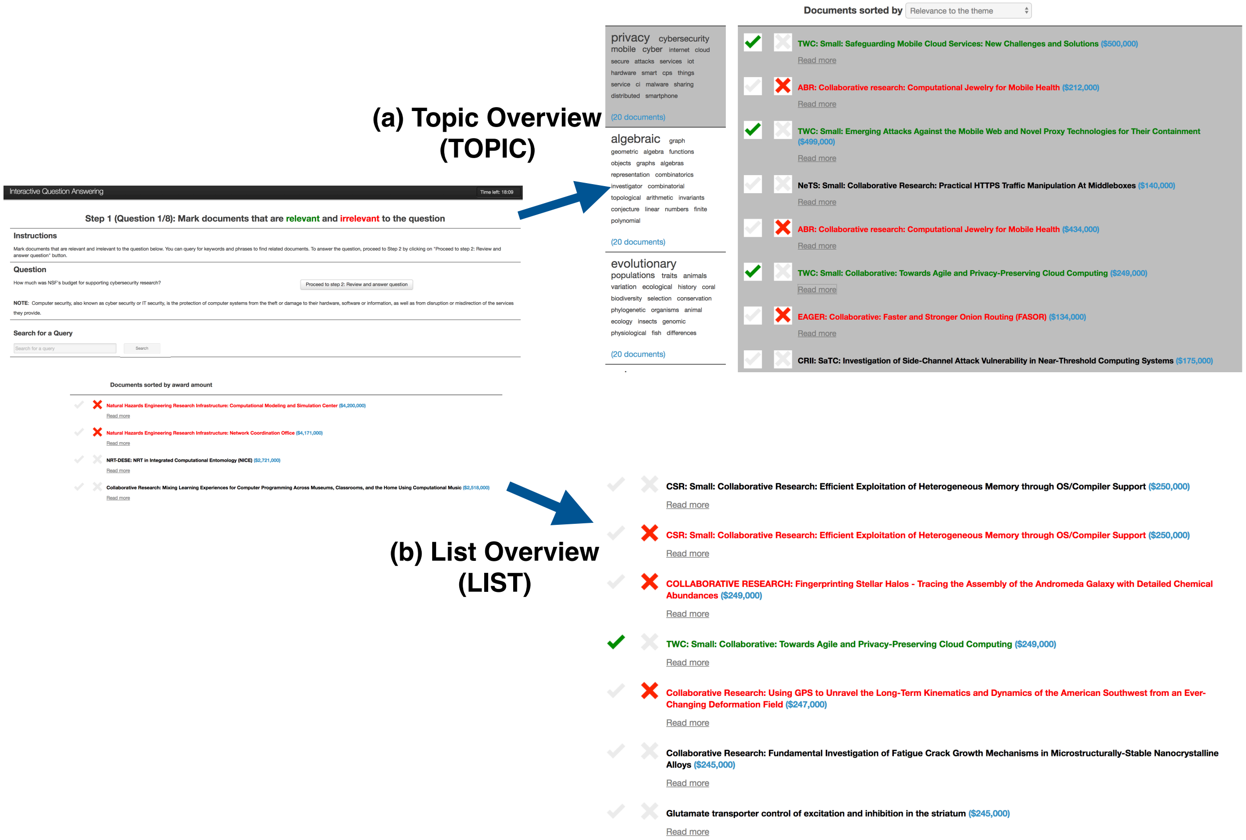
area. Coming up with accurate and convincing answers to these questions requires an understanding of the large number of funded grants and is hard to do manually.

We take a machine learning with a human-in-the-loop approach to help users answer such questions. We build on the ALTO interface presented in Chapter 4 to design a framework and help users find grants that are **relevant** to a question, view a summary of those grants, and answer questions (Section 5.1). We create two experimental conditions to compare the effect of providing users with either the topic model information or a simple list of documents (Section 5.2). Following this section, we then describe our interactive system and machine learning tools we have used to help users in this task (Section 5.3). Through a user study with twenty participants, we evaluate the effectiveness of our framework by comparing results from the two conditions introduced above (Section 5.4).

5.1 Understanding Science Policy Using Topic Overviews and Active Learning

We create a single interface to help users find the answer to a given question from a large collection of documents about science. Finding answers to questions from a large corpus is hard because users need to wade through the large unorganized collection, understand the collection, find relevant documents to the question, and finally find the answer to the question from the documents they have deemed relevant. Classification algorithms can be used to automatically find relevant documents to the question. However, this can only be done **after** we have a training set of **relevant** and **irrelevant** documents. While Chapter 4 addresses the difficulty of inducing **topical** labels and creating training sets for classifying documents in **topical** categories, this chapter assumes a known label set (**relevant** or **irrelevant**) and addresses the problem with finding documents of interest.

Having demonstrated that topic models and active learning help users induce a global label set from a large corpus and assign them to individual documents (Chapter 4), we use the same approach to help users mark documents that are **relevant** or **irrelevant** to a given question about the policies of a funding agency. Next, we train a classifier using the user-generated training set to automatically find additional relevant documents. Finally, we provide users with a summary of the retrieved relevant documents and ask them to answer the question.



Step 1 (Question 1/8): Mark documents that are relevant and irrelevant to the question

Instructions

Mark documents that are relevant and irrelevant to the question below. You can query for keywords and phrases to find related documents. To answer the question, proceed to Step 2 by clicking on "Proceed to step 2: Review and answer question" button.

Question

How much was NSF's budget for supporting cybersecurity research?

NOTE: Computer security, also known as cyber security or IT security, is the protection of computer systems from the theft or damage to their hardware, software or information, as well as from disruption or misdirection of the services they provide.

Search for a Query

Documents sorted by award amount

- ✗ Natural Hazards Engineering Research Infrastructure: Computational Modeling and Simulation Center ($4,200,000)
- ✗ Natural Hazards Engineering Research Infrastructure: Network Coordination Office ($4,171,000)
- NRT-DESE: NRT in Integrated Computational Entomology (NICE) ($2,721,000)
- Collaborative Research: Mixing Learning Experiences for Computer Programming Across Museums, Classrooms, and the Home Using Computational Music ($2,518,000)

privacy cybersecurity mobile cyber internet cloud secure attacks services iot hardware smart cps things service ci malware sharing distributed smartphone
(20 documents)

algebraic graph geometric algebra functions objects graphs algebras representation combinatorics investigator combinatorial topological arithmetic invariants conjecture linear numbers finite polynomial
(20 documents)

evolutionary populations traits animals variation ecological history coral biodiversity selection conservation phylogenetic organisms animal ecology insects genomic physiological fish differences
(20 documents)

- ✓ TWC: Small: Safeguarding Mobile Cloud Services: New Challenges and Solutions ($500,000)
- ✗ ABR: Collaborative research: Computational Jewelry for Mobile Health ($212,000)
- ✓ TWC: Small: Emerging Attacks Against the Mobile Web and Novel Proxy Technologies for Their Containment ($499,000)
- NeTS: Small: Collaborative Research: Practical HTTPS Traffic Manipulation At Middleboxes ($140,000)
- ✗ ABR: Collaborative research: Computational Jewelry for Mobile Health ($434,000)
- ✓ TWC: Small: Collaborative: Towards Agile and Privacy-Preserving Cloud Computing ($249,000)
- ✗ EAGER: Collaborative: Faster and Stronger Onion Routing (FASOR) ($134,000)
- CRII: SaTC: Investigation of Side-Channel Attack Vulnerability in Near-Threshold Computing Systems ($175,000)

- CSR: Small: Collaborative Research: Efficient Exploitation of Heterogeneous Memory through OS/Compiler Support ($250,000)
- ✗ CSR: Small: Collaborative Research: Efficient Exploitation of Heterogeneous Memory through OS/Compiler Support ($250,000)
- ✗ COLLABORATIVE RESEARCH: Fingerprinting Stellar Halos - Tracing the Assembly of the Andromeda Galaxy with Detailed Chemical Abundances ($249,000)
- ✓ TWC: Small: Collaborative: Towards Agile and Privacy-Preserving Cloud Computing ($249,000)
- ✗ Collaborative Research: Using GPS to Unravel the Long-Term Kinematics and Dynamics of the American Southwest from an Ever-Changing Deformation Field ($247,000)
- Collaborative Research: Fundamental Investigation of Fatigue Crack Growth Mechanisms in Microstructurally-Stable Nanocrystalline Alloys ($245,000)
- Glutamate transporter control of excitation and inhibition in the striatum ($245,000)

Figure 5.1: Our interactive interface to help users understand science policy in the two conditions: (a) topic overview and (b) list overview. Topic overview shows documents organized by the extracted topics. The user can click on a topic to see the documents that are associated with the topic. List overview shows documents in a simple and unsorted list format.

Similar to our hypotheses in Chapter 4, we hypothesize that users who see documents organized by topics, which combined, provide an overview of the corpus, will be able to answer questions **better** and **faster** compared to the users who see a list of unsorted documents.

5.2 Experimental Design

Our goal is to characterize how providing a global overview of the corpus via topics that are automatically extracted from the document collection using topic models such as LDA, can aid users to find documents of interest from a large corpus and answer specific questions. This section describes our experimental conditions and data.

5.2.1 Experimental Conditions

Our study uses a between-subject design[2] with one factor of **data set overview**. The two conditions are (1) topic overview (TOPIC) and (2) list overview (LIST).

Like ALTO, the **topic overview** presents documents organized by their topic, which are displayed as an ordered list of words (Figure 5.1a). On the other hand, the **list overview** presents documents in a simple and unsorted list format (Figure 5.1b). In both conditions, we display the same number of documents ($20K$, where K is the number of topics), but the list overview lacks the topic information. There has already been sufficient work that shows the effectiveness of active learning **selections** in helping people label documents faster (Settles, 2011; Poursabzi-Sangdeh et al., 2016). Therefore, we do not consider document selection as a factor in this study; we use active learning in both conditions.

5.2.2 Data

The data set that we use in our experiments includes the grants funded by the National Science Foundation (NSF) in 2016 (NSF-2016). The titles, abstracts, and amounts awarded for all the grants are publicly available.[3] After filtering the grants that have duplicate content, this data set has 10,333 documents.

5.3 Topic Assisted Document Browsing and Understanding

In this section, we describe the machine learning tools that we use in our interactive system. We then describe the process of answering questions by a user.

In addition to the topic model, which is only used in the TOPIC condition, our system has three main components: (1) ranker, (2) classifier, and (3) selector. The **ranker** displays documents that are related to a query made by users. The **classifier** periodically identifies documents that are likely **relevant** or **irrelevant** and displays them to users. The **selector** is the active learning component, which periodically selects documents that are more beneficial in learning a good classifier and points users' attention to them. We now describe these components in more detail.

[2] In a between-subject design, two or more groups of subjects each get tested by a different factor simultaneously.

[3] `https://www.nsf.gov/awardsearch/download.jsp`

5.3.1 Ranker

One of the most common interactions of humans with machines is via information search using search engines. As such, most users are familiar with creating search queries to find pieces of information of interest. Therefore, we use the ranker to sort documents based on their relevance to the queries users make.

Users can search for phrases to rank and retrieve documents related to their query. The ranker assigns a **relevance score** for each document in the corpus. Next, the documents are sorted based on their score and displayed to the user. The ranker exposes users to a new set of documents, other than the $20K$ documents that are highly associated with the topics, every time they make a new query. For consistency, we use the same ranking method in the TOPIC and LIST conditions. In the LIST condition, these documents are sorted by their relevance score. In the TOPIC condition, the documents are still displayed organized by their prominent topic, and the topics are sorted based on the sum of their associated documents' relevance scores.

The ranker uses an LDA-based method by Wei and Croft (2006) to rank and retrieve documents that are relevant to the user's query. Unlike the traditional retrieval methods that perform on documents modeled as a "bag of words", the LDA-based method models documents as a linear combination of a "bag of words" **document model (DM)** with smoothing (Zhai and Lafferty, 2001) and an **LDA model**:

$$P(w \mid d) = \lambda \, P_{\text{DM}}(w \mid d) + (1 - \lambda) \, P_{\text{LDA}}(w \mid d), \tag{5.1}$$

where w is a word in the query and λ is a hyper-parameter that controls the importance of the document model versus the LDA model. $P_{\text{LDA}}(w \mid d)$ and $P_{\text{DM}}(w \mid d)$ are the probabilities of observing w in document d under the LDA model and the DM model, respectively. Additionally,

$$P_{\text{DM}}(w \mid d) = \frac{N_d}{N_d + \mu} P_{\text{ML}}(w \mid d) + (1 - \frac{N_d}{N_d + \mu}) \, P_{\text{ML}}(w \mid \text{corpus}), \tag{5.2}$$

where μ is the parameter for Dirichlet smoothing.

The relevance score of a document d is then calculated as the likelihood of the model generating the query:

$$S_{Q,d} = \prod_{w \in Q} P(w \mid d), \tag{5.3}$$

where Q is the query phrase and w is a word in the query. We refer the reader to the original paper by Wei and Croft (2006) for more details on the LDA-based ranking.

Parameter Tuning

We need to set two parameters when using the LDA-based method: the Dirichlet prior for smoothing (μ) and the weight of the document model (λ). We consider $\mu \in \{1, 10, 100, 1000, 10000\}$ and $\lambda \in \{0, 0.1, 0.2, ..., 1\}$ as possible parameters. To set these parameters, we create synthetic queries from each document title and use the LDA-based method with all the possible parameters to rank the document. Then, we select the parameters that lead to the lowest average ranking of the documents. To create the queries for a document, we extract the nouns from the document title, rank them based on their frequency in the document, and add them incrementally to make queries. For example, for the document with the title Studies in Commutative Algebra, the noun algebra appears more frequently in the document than studies. Therefore, we make two queries: algebra and algebra studies. We then calculate the average rank of this document with these two queries and all the possible parameters. We do the same for all other documents and select $\mu = 1000$ and $\lambda = 0.7$ as these values lead to the minimum average ranking.

5.3.2 Classifier

A classifier automatically identifies documents that would be helpful for the user to label. It also automatically identifies relevant documents based on the user-generated training set and displays them to the user to help them review and answer the question. We use a logistic regression classifier (Carpenter, 2008, LingPipe). The parameters of this classifier are optimized with cross validation based on the prominent topic as the label for each document.[4] The classifier uses unigrams and the topic probabilities as features.

5.3.3 Selector

We use active learning to periodically **select** documents that are more beneficial in learning a good classifier faster. The selector identifies informative documents based on an active learning strategy and

[4] We use `blockSize` = 1/#examples, `minEpochs` = 100, `learningRate` = 0.1, `minImprovement` = 0.001, `maxEpochs` = 1000, and `rollingAverageSize` = 100. The regression is unregularized.

directs users' attention to them. For consistency, we use the same strategy in both conditions. Similar to ALTO, documents are sorted and selected based on the uncertainty level of the classifier (U):

$$U_d = \mathbb{H}_C \left[y_d \right], \tag{5.4}$$

where $\mathbb{H}_C \left[y_d \right]$ is the classifier entropy. Entropy is a measure of how confused a classifier C is about the label y of a document d. If the user has queried for a phrase, we also consider relevance scores of the documents:

$$U_d^Q = 0.5 \, \mathbb{H}_C \left[y_d \right] + 0.5 \, S_{Q,d}, \tag{5.5}$$

where $S_{Q,d}$ is the relevance score of document d to the user query Q (Equation 5.3).[5] Intuitively, U_d^Q selects documents that are relevant to the user query **and** are informative for the classifier. We then mix the documents selected by Equation 5.4 and Equation 5.5 by interleaving and display them to the user.

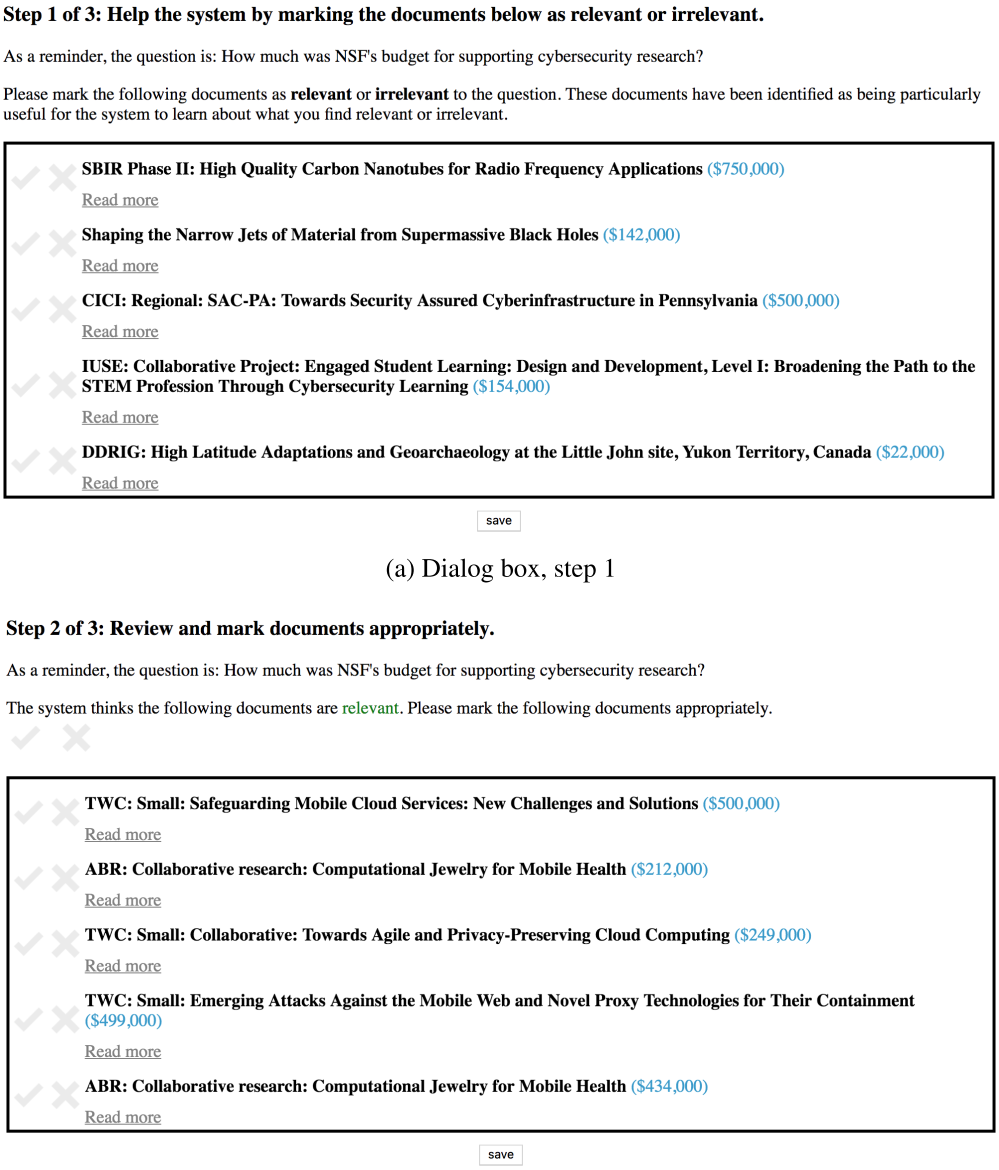
5.3.4 User Answering Process

The user's answering process is the same in both conditions, with the difference that the TOPIC condition allows users to examine the topics that are associated with each document. First, the user reviews documents and labels them as **relevant** or **irrelevant** to the question. Then, the system provides a summary of the documents they have deemed relevant as well as additional ones it predicts are also relevant. The user reviews these documents and answers the question.

Users start by seeing a question and a set of document titles along with the amount given for each grant.[6] In the TOPIC condition, by default, the user sees the top twenty documents in the first topic (Figure 5.1a). Clicking on each topic shows the top twenty documents associated with that topic. In this condition, the documents are by default sorted based on their relevance to the topic. Users can sort them based on the grant amounts if they want to focus on the grants with high amounts. In the LIST condition, the documents are sorted based on the grant amounts (Figure 5.1b). In both conditions, clicking on the "Read more" link under the title of each document displays the full text of the document. The user can label documents as relevant or irrelevant by clicking on ✔ and ✗ buttons (Figure 5.1).

[5] We standardize the entropy and relevance scores to have a mean of zero and standard deviation of one.

[6] When presenting the grant amounts to the users, we round the amounts to the nearest \$1000.



Step 1 of 3: Help the system by marking the documents below as relevant or irrelevant.

As a reminder, the question is: How much was NSF's budget for supporting cybersecurity research?

Please mark the following documents as **relevant** or **irrelevant** to the question. These documents have been identified as being particularly useful for the system to learn about what you find relevant or irrelevant.

✓ ✗ **SBIR Phase II: High Quality Carbon Nanotubes for Radio Frequency Applications** ($750,000)
Read more

✓ ✗ **Shaping the Narrow Jets of Material from Supermassive Black Holes** ($142,000)
Read more

✓ ✗ **CICI: Regional: SAC-PA: Towards Security Assured Cyberinfrastructure in Pennsylvania** ($500,000)
Read more

✓ ✗ **IUSE: Collaborative Project: Engaged Student Learning: Design and Development, Level I: Broadening the Path to the STEM Profession Through Cybersecurity Learning** ($154,000)
Read more

✓ ✗ **DDRIG: High Latitude Adaptations and Geoarchaeology at the Little John site, Yukon Territory, Canada** ($22,000)
Read more

(a) Dialog box, step 1

Step 2 of 3: Review and mark documents appropriately.

As a reminder, the question is: How much was NSF's budget for supporting cybersecurity research?

The system thinks the following documents are relevant. Please mark the following documents appropriately.

✓ ✗

✓ ✗ **TWC: Small: Safeguarding Mobile Cloud Services: New Challenges and Solutions** ($500,000)
Read more

✓ ✗ **ABR: Collaborative research: Computational Jewelry for Mobile Health** ($212,000)
Read more

✓ ✗ **TWC: Small: Collaborative: Towards Agile and Privacy-Preserving Cloud Computing** ($249,000)
Read more

✓ ✗ **TWC: Small: Emerging Attacks Against the Mobile Web and Novel Proxy Technologies for Their Containment** ($499,000)
Read more

✓ ✗ **ABR: Collaborative research: Computational Jewelry for Mobile Health** ($434,000)
Read more

(b) Dialog box, step 2

Figure 5.2: After the user has labeled ten documents, a dialog box opens with three steps to complete: (a) top ten documents extracted by active learning, (b) top ten documents identified by the classifier as being relevant, and top ten documents identified by the classifier as being irrelevant. After completing these three steps, the user is redirected back to the main window to continue labeling documents.

After the user labels ten documents, the classifier runs and identifies some documents for the user to label. These documents are displayed in a dialog box (Figure 5.2). The user should complete three steps with ten documents in each step in this dialog box.[7] The following documents are displayed in each step:

(1) Step 1: top ten documents identified by the selector (active learning - Figure 5.2a).

(2) Step 2: Top ten documents identified by the classifier as being relevant (Figure 5.2b).

(3) Step 3: Top ten documents identified by the classifier as being irrelevant.

In Step 1, we display documents identified by the selector (the active learning component). Unlike ALTO, where the selected documents are displayed in the main window, we show these documents in a separate dialog box. This was done to avoid inconsistency and confusion in the sorting criteria in the main interface: if the active learning documents are generated at a time that the user has made a query, we would like to display documents sorted based on uncertainty (Equation 5.5). On the other hand, the user would expect the documents to be sorted based on their relevance to the query.

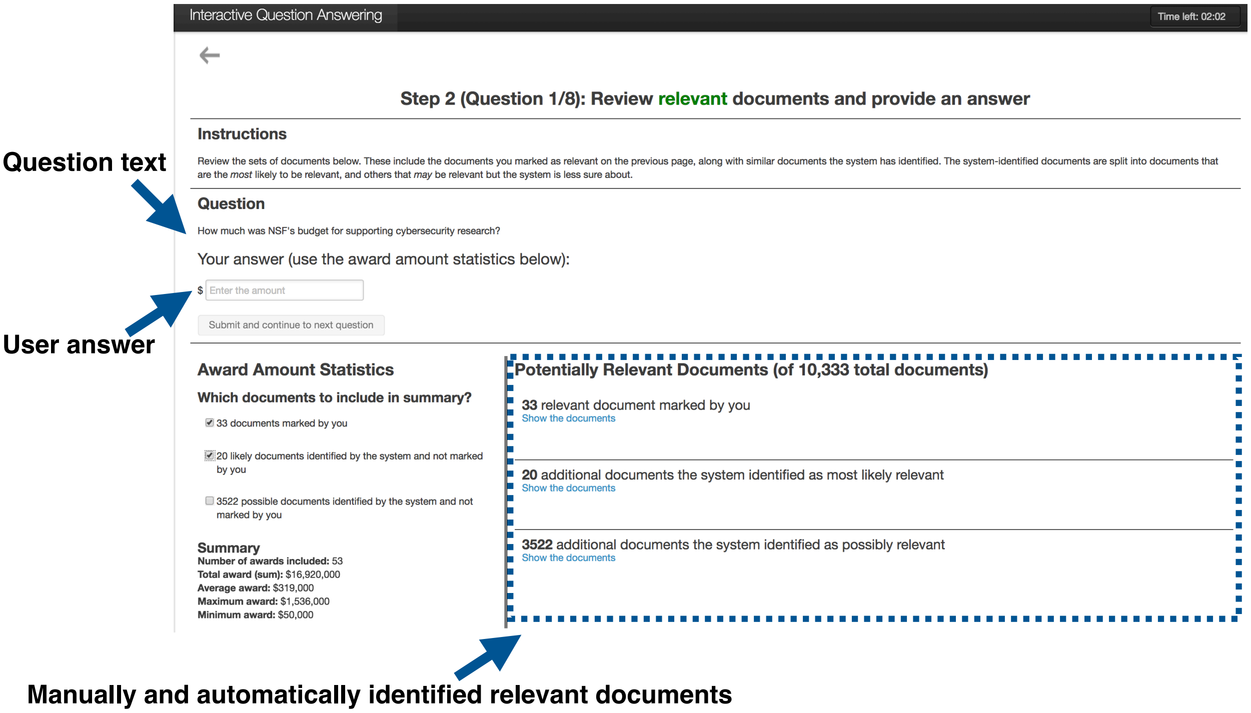
In Step 2 and Step 3, the user can save time and confirm the classifier assigned labels by clicking on the **select all as relevant** or **select all as irrelevant** buttons. Unlike ALTO, we do not assign labels automatically to documents. The answer to quantitative questions can be dramatically changed if we automatically label some documents as relevant. Therefore, we show the top ten documents that the classifier is confident they are relevant or irrelevant and ask the user to confirm or correct these predictions.

Users can also make queries using a search button to see other documents that are related to the query. The ranker (Section 5.3.1) sorts all documents based on their relevance to the user query and displays the top fifty. These documents are, by default, sorted based on their relevance score to the query. Similar to the documents in the main window, users can sort them based on the grant amounts. After they have gone through these documents, they can remove the search results and go back to the main window.

Once the user labels enough documents and is ready to answer the question, we direct them to the **review page** (Figure 5.3). In this page, the user sees a summary of the relevant documents they have manually marked as well as the ones that the classifier has automatically identified as relevant[8] and answers the question. The classifier-identified relevant documents are divided in two sets: 1) top twenty documents,

[7] The user may ignore the dialog box and continue labeling in the main window by closing it.

[8] The classifier runs in the background using the training set the user has created by labeling document as relevant or irrelevant.



Interactive Question Answering

Time left: 02:02

Step 2 (Question 1/8): Review relevant documents and provide an answer

Instructions

Review the sets of documents below. These include the documents you marked as relevant on the previous page, along with similar documents the system has identified. The system-identified documents are split into documents that are the *most* likely to be relevant, and others that *may* be relevant but the system is less sure about.

Question

How much was NSF's budget for supporting cybersecurity research?

Your answer (use the award amount statistics below):

$ Enter the amount

Submit and continue to next question

Award Amount Statistics

Which documents to include in summary?

☑ 33 documents marked by you

☑ 20 likely documents identified by the system and not marked by you

☐ 3522 possible documents identified by the system and not marked by you

Summary
Number of awards included: 53
Total award (sum): $16,920,000
Average award: $319,000
Maximum award: $1,536,000
Minimum award: $50,000

Potentially Relevant Documents (of 10,333 total documents)

33 relevant document marked by you
Show the documents

20 additional documents the system identified as most likely relevant
Show the documents

3522 additional documents the system identified as possibly relevant
Show the documents

Figure 5.3: The review page in our interface. After the user has labeled enough documents and is ready to answer the question, they see a summary of the relevant documents they have manually marked as well as the ones that the classifier has automatically identified as relevant and answer the question.

which the classifier is most confident that they are relevant and 2) all other documents the classifier predicts are relevant.[9] The user then clicks on "Show the documents" to see the documents in each category, inspects the content, and writes the answer based on the relevant document. Some basic statistics (sum, average, maximum, and minimum) is also provided on the amounts of the existing relevant grants, which the user can use to answer quantitative questions.

Discussion

In the process of piloting our study, users did not label a diverse set of documents as irrelevant, causing the classifier to perform poorly both in identifying uncertain documents to point the user to and in identifying relevant documents in the review page. Users did not spend time for exploring documents in the main interface, which includes documents with diverse content. Instead, they focused on labeling documents

[9] The classifier-identified relevant documents in each set are displayed in a random order to help users make a realistic and informed decision on whether or not to include the amount of all documents in a set in their final answer.

Topic words	Document Title
privacy, cybersecurity, mobile, cyber, internet, cloud, secure, attacks, services, iot	Establishment of a Mentored Cybersecurity Research Workshop for Graduate Students and Support for the Conference on Cybersecurity Education, Research and Practice
soil, ecosystems, ecosystem, ecological, nitrogen, forest, land, river, coastal, nutrient	Collaborative Research: The Sustainability of Riparian Forests in Expanding Amazonian Agricultural Landscapes
speech, languages, linguistic, speakers, children, english, words, sound, linguistics, reading	Doctoral Dissertation Research: The role of tongue position in voicing contrasts in cross-linguistic contexts

Table 5.1: Automatically discovered sorted lists of terms (topics) from our NSF-2016 data set. These topics give users a sense of documents' main themes and help users navigate through the corpus and find documents of interest.

that were retrieved by the ranker. To avoid this issue and encourage users to take actions that will help the system best, we do not allow the users to make queries in the first three minutes for each question. Users are instructed to use the first three minutes to explore the documents (and topics in the TOPIC condition) and get an overview of the corpus. Doing so encourages them to review and label a diverse set of irrelevant documents, which in turn leads to a better classifier performance and better suggestions by the selector.

5.4 Experiments and Evaluation

We use LDA to generate topics. To choose the number of topics (K), we calculate the average topic coherence (Lau et al., 2014, Equation 2.1) between $K = 5$ and $K = 200$ and choose $K = 45$, as it has the maximum coherence score. After filtering words based on TF-IDF (Equation 2.2), we use Mallet (McCallum, 2002) to learn topics.[10] Table 5.1 shows examples of topics and their highest associated documents from our NSF-2016 corpus.

The questions that we ask users to answer are inspired by **Questions for the Record (QFR)**.[11] QFRs are a set of questions that are asked following a hearing in Congress, which are held regarding the details and specifics of a proposed bill or funding request. We manually extract questions that are about the amount of money the NSF has spent on a specific area of research from these hearings. After inspecting the

[10] We use $\alpha = 0.1$ and optimize-interval= 10.

[11] https://www.congress.gov/congressional-record

ID	Question
Q$_1$	How much was NSF's budget for supporting cybersecurity research?
Q$_2$	How much did NSF spend on Climate Change research?
Q$_3$	How much was NSF's budget for supporting research on causes and responses to violence?

Table 5.2: The three QFR-inspired questions that we use for our experiments.

questions and getting feedback from several experts, we revise the questions so that they are more aligned with the intended task. The selected questions are inspired by the actual questions that are asked during the hearings—we only remove potential sources of ambiguity for non-experts. Table 5.2 shows the final questions.

5.4.1 Experiment with Domain Experts

Our experiments require gold standard answers and gold standard relevant grant sets for each question. Therefore, we first run the experiment with four domain experts. To recruit experts, we posted an advertisement on the Science of Science and Innovation Policy (SciSIP) email list and hired four experts. After providing them with detailed instructions of the system in the TOPIC condition,[12] they interacted with the system, labeled grants as **relevant** or **irrelevant**, and answered the questions. Each of the experts were paid with a $35 Amazon gift card.

To measure the level of agreement between experts, we calculate Fleiss' kappa (Fleiss, 1971), which is commonly used as a statistical measure of how consistent the labels are among several annotators. Because experts do not manually label all the grants, in addition to **relevant** and **irrelevant** labels, we first assign **neutral** for the grants that the expert does not manually label and then calculate κ. If the experts agree completely, then κ will equal to 1. If there is no agreement other than what would be expected by chance, then κ will be less than zero. Based on the interpretation provided by Landis and Koch (1977), experts fairly agree on Q$_1$ ($\kappa = 0.202$) and on Q$_3$ ($\kappa = 0.286$) and they slightly agree on Q$_2$ ($\kappa = 0.197$).

We aggregate the answers and grant labels from our experts to find gold standard answers. Table 5.3

[12] We expect the gold answers to be generated with all the possible available information. Given that our hypothesis is that topics provide an overview of the corpus to help users, we assign all experts to the TOPIC condition.

Question	$M \pm SD\,[median]$
	Gold Answer ($)
Q_1	$36,623,000 \pm 18,420,130\,[22,086,000]$
Q_2	$265,423,500 \pm 205,283,387\,[90,073,500]$
Q_3	$2,800,500 \pm 333,520\,[2,824,000]$

Table 5.3: Mean, standard deviation, and median of answers given for the three questions by experts.

Question		$M \pm SD\,[median]$	
	# ManRel	# ManIrrel	# FinalRel
Q_1	$61.25 \pm 44.32\,[52]$	$73.5 \pm 71.85\,[68]$	$119.75 \pm 118.44\,[72]$
Q_2	$43.5 \pm 23.56\,[34.5]$	$90.75 \pm 54.06\,[93.5]$	$750.75 \pm 1016.36\,[393.5]$
Q_3	$31.5 \pm 13.48\,[29]$	$133.75 \pm 55.93\,[143]$	$31.5 \pm 13.48\,[29]$

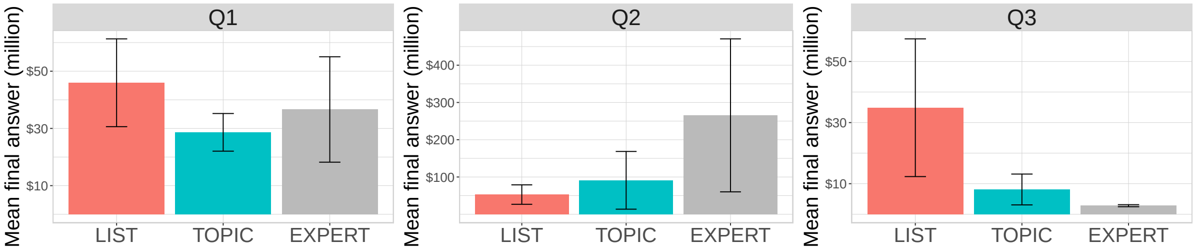
Table 5.4: Mean, standard deviation, and median of the number of manually labeled grants as relevant (# manRel), the number of manually labeled grants as irrelevant (# manIrrel), and the number of relevant grants considered in the final answer (# FinalRel) for the three questions by experts.

shows the average final numerical answers and Table 5.4 shows the average number of labeled grants and the average number of relevant grants that are considered in the final answer for the three questions. To generate **gold labels** for each question, we first aggregate the grants that experts manually label as relevant and irrelevant by finding the union of grants that experts manually mark.[13] Next, we use the aggregated relevant and irrelevant sets to train a classifier (Section 5.3.2) and predict the gold labels for the remaining grants.

5.4.2 User Study

We conduct a user study with 20 participants to evaluate the effectiveness of topic overviews provided by topic models against an alternative that lacks topic overviews.

[13] We exclude the grants that are manually marked as relevant (or irrelevant) by at least one expert from the aggregated irrelevant (or relevant) set.



(a) (b) (c)

Figure 5.4: Mean final answer of experts compared to the mean final answer of participants in the LIST condition and TOPIC condition for the three questions. Error bars indicate one standard errors. There is no significant difference between the LIST condition and the TOPIC condition in terms of participants' final answers to the three questions.

5.4.2.1 Method

Our recruitment procedure is similar to the procedure in Chapter 4. We use the freelance marketplace Upwork to recruit 20 online participants, all of whom satisfy the same selection criteria. Participants are randomly assigned to either the LIST condition or the TOPIC condition. We end with ten participants in each condition.

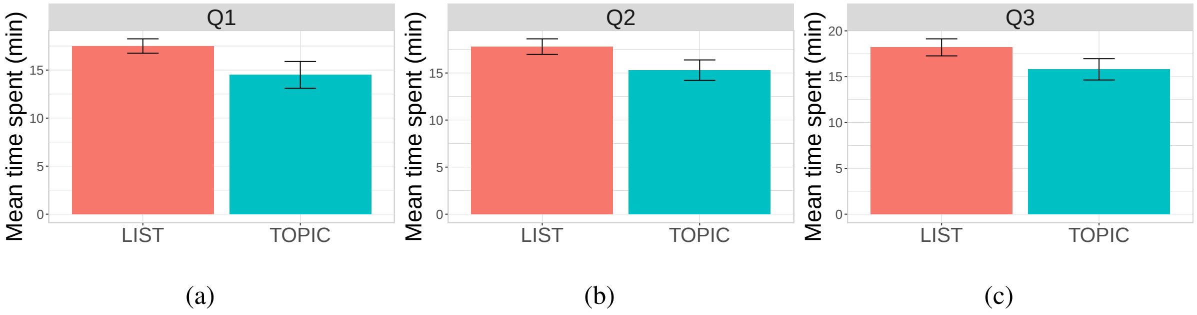
Participants completed a demographic questionnaire (Appendix C.1), viewed a video of task instructions, and then interacted with the system and answered the three questions. Participants had a maximum of twenty minutes for each question.[14] The session ended with a survey (Appendix C.2) similar to the one in the ALTO study. Each participants was paid fifteen dollars. For statistical analysis, we use one-way ANOVAS with Aligned Rank Transform (Wobbrock et al., 2011, ART).

5.4.2.2 Results

Do topics help users answer questions more accurately and faster?

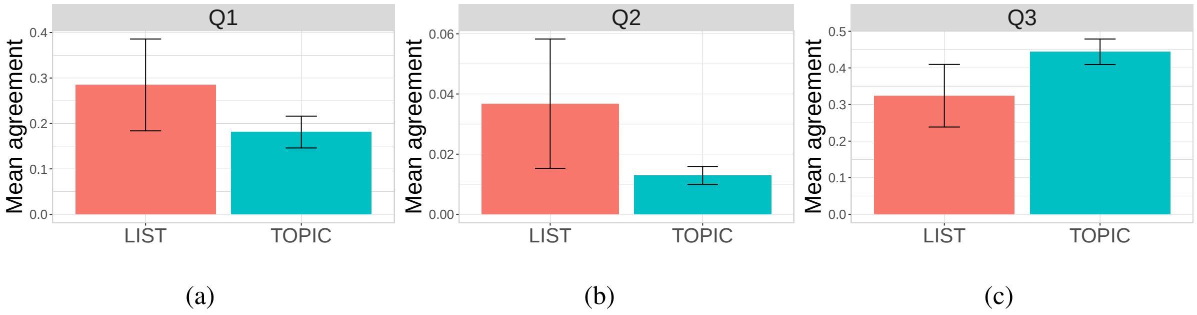
We compare the answers provided by participants to the aggregated gold answers provided in table 5.3. Figure 5.4 shows the mean final answer of participants in the LIST condition, participants in the TOPIC condition, and the experts for each of the three questions. There is no significant difference between the LIST

[14] Twenty minutes of activity, excluding system time to classify, display dialog box, and display the review page.



(a) (b) (c)

Figure 5.5: Mean time spent to answer a question by participants in the LIST condition and TOPIC condition for the three questions. Error bars indicate one standard errors. There is no significant difference between the LIST condition and the TOPIC condition in terms of the time it takes for participants to answer the question.



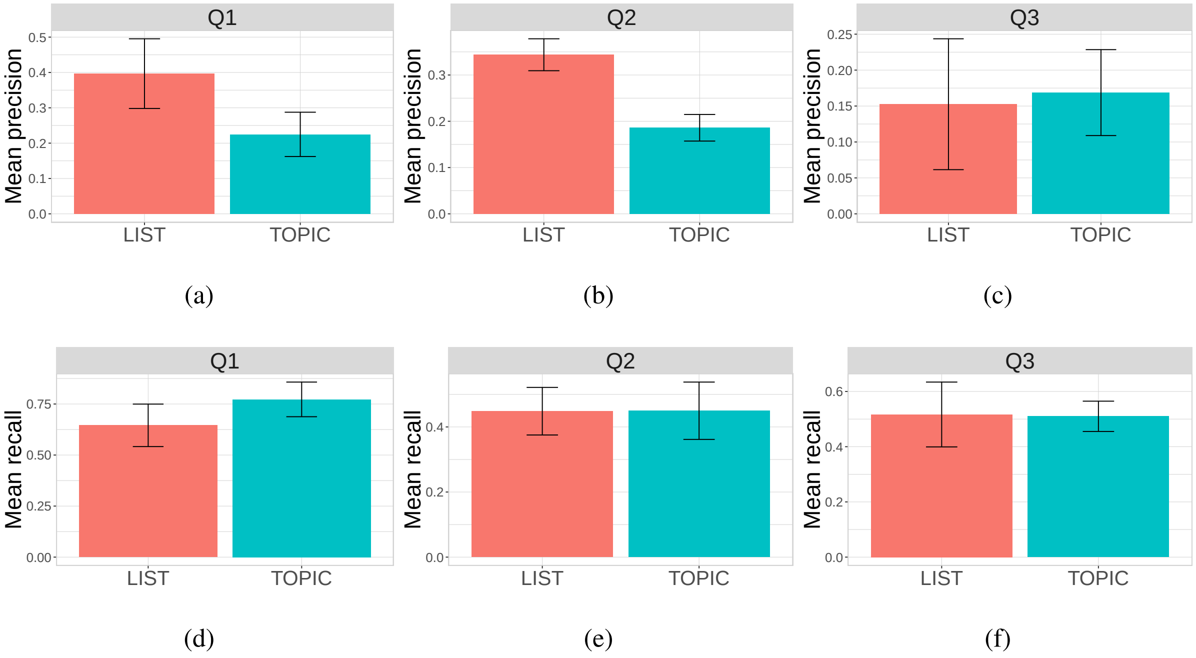
(a) (b) (c)

Figure 5.6: Mean agreement (the fraction of gold relevant grants that is considered as relevant in the final answer) of participants in the LIST condition and the mean agreement of participants in the TOPIC condition with experts for the three questions. Error bars indicate one standard errors. There is no significant difference between the LIST condition and the TOPIC condition in terms of participants' agreement with experts.

condition and the TOPIC condition in terms of the value of final answers to all three questions. Additionally, Figure 5.5 shows the mean time participants spend for answering each of the three questions. There is no significant difference in the time it takes for participants to answer the questions between the LIST condition and the TOPIC condition.

Do participants who see topic overviews agree with experts more?

We define a participant's agreement with experts to be the fraction of gold relevant grants that the participant considers as relevant in her final answer. Figure 5.6 shows the mean agreement of participants in the LIST condition and the mean agreement of participants in the TOPIC condition with experts for the three



(a) (b) (c)

(d) (e) (f)

Figure 5.7: Mean precision (top) and recall (bottom) of the relevant grants for the three questions. Error bars indicate one standard errors. Participants in the LIST condition have significantly higher precision in Q$_2$. No other significant difference is found between the LIST condition and the TOPIC condition.

questions. There is no significant difference between the LIST condition and the TOPIC condition in terms of the participants' agreement with experts.

Do topics help users create a better training set for classifiers?

One of our hypotheses was that topic models will help users find documents of interest, e.g., relevant documents to a question. To see whether this hypothesis holds, we learn a classifier using the training set that each participant creates by manually marking relevant and irrelevant grants. This classifier predicts whether the remaining grants are relevant or irrelevant. We then compare these predictions against the gold labels (generated from aggregated labels by experts, Section 5.4.1). Figure 5.7 shows the mean precision and recall of participants in the LIST condition and the TOPIC condition. While participants in the LIST condition have significantly higher precision for Q$_2$ than participants in the TOPIC condition ($F(1, 18) = 15.18$, $p = .001$), there is no other significant difference between the LIST condition and the TOPIC condition in terms of

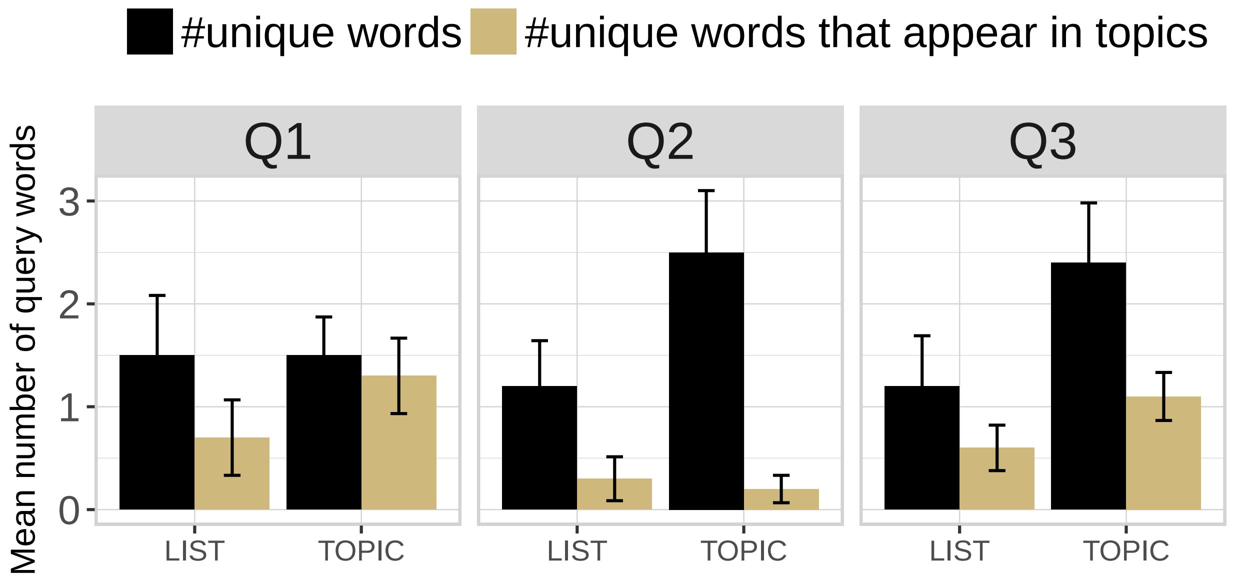


Figure 5.8: Mean number of unique query words and mean number of unique query words that appear in topic words. Error bars indicate one standard errors. There is no significant difference between the LIST and TOPIC condition in terms of the number of unique words in queries and the number of unique words in queries that are topic words.

precision and recall for the three questions.

Do topics lead to insights for information search?

Making queries and searching for information helps users find grants that are about specific phrases and exposes users to different sets of grants. Participants in the TOPIC condition make significantly more queries ($M = 1.77$ and $median = 1$) than participants in the LIST condition ($M = 1.07$ and $median = 1$) for all questions combined ($F(1, 58) = 7.66$, $p = .007$). One of our hypotheses was that topics will help users make better and more queries. Figure 5.8 shows the mean number of unique query words and the mean number of unique query words that appear in topic words for each question. While there is no statistically significant difference between the LIST condition and the TOPIC condition in terms of the number of unique words in queries and the number of unique words in queries that are topic words, the results show some evidence of participants being inspired by topics to make more queries that are related to topic words. For example, a participant in the TOPIC condition searched for ecosystem in Q_2, which appears in the topic soil, ecosystems, ecosystem, ecological, nitrogen, forest, land, river, coastal, nutrient. Another participant in the topic condition searched for atmosphere for the same question, which appears in another topic atmospheric, weather, tropical, cloud, precipitation, variability, atmosphere, clouds, el, ni.

Subjective Ratings:

One-way ANOVAs with ART for each of the subjective ratings revealed two significant results:

(1) Participants in the TOPIC condition find the task significantly less mentally demanding ($M = 10.1$ and $median = 10$) than participants in the LIST condition ($M = 15$ and $median = 15$) on a scale from 1 (low) to 20 (high) ($F(1, 18) = 4.94$, $p = .039$).

(2) Participants rate the interface in the TOPIC condition ($M = 5.7$ and $median = 5.5$) significantly more helpful than the interface in the LIST condition ($M = 4.5$ and $median = 4$) on a scale from 1 (not helpful at all) to 7 (very helpful) ($F(1, 18) = 7.02, p = .016$).

Participants in the TOPIC condition find the topic information to be useful in completing the task ($M = 6.4$ and $median = 6.5$) on a scale from 1 (not helpful at all) to 7 (very helpful). On a similar scale, participants in both conditions rate the **search for queries** feature helpful ($M = 5.95$ and $median = 6$). Overall, participants were positive about their experience with the system, rating their overall satisfaction with the interface positively ($M = 5.7$ and $median = 6$) on a scale from 1 (not satisfied at all) to 7 (very satisfied).

5.5 Discussion

Following our results in Chapter 4 that showed the effectiveness of topic overviews in helping users induce labels for large corpora and create the training set for classifiers, we had hypothesized that topic overviews will be helpful for users to understand large corpora and answer specific questions about them. However, our user study results did not reveal such an effect. We believe this unexpected outcome was due to several factors.

First, the NSF-2016 data set and the questions that we used for our experiments have technical and scientific content and require expertise. This makes the task more mentally demanding and frustrating for non-experts. We believe that giving users the ability to choose the domain and questions that they are invested in, they care about, and have basic knowledge on could potentially lead to different results.

Reliability of gold answers is another related issue. We used the data from experts, who at least had strong interest in science policy and were extensively instructed on the system to find gold answers and gold relevant and irrelevant grants for each question. However, the agreement among the experts was worryingly low, probably due to a high amount of variation in the data (Section 5.4.1).

Similarly, we observed a high amount of variation in the data from non-experts in our user study. One major source of variability comes from the sets of relevant grants that users include in their final answer (Figure 5.3): if a user includes the classifier identified relevant grants in the final answer and another user does not, their answers will be dramatically different, even if the grants that they manually marked were similar. Additionally, running the study with a larger sample size can reduce the amount of variation in the data.

One clear difference of the experiments in this chapter with the experiments in Chapter 4 is in assigning binary labels (**relevant** or **irrelevant**) to documents rather than **topical** labels. While we hypothesized that topic overviews would guide users in searching for and exploring documents of interest, they could also serve as a source of distraction, especially if their benefit is not immediately clear for users.

The ranker (Section 5.3.1) was included in the framework to enable users find relevant documents to the queries they make. Because searching for keywords is a relatively intuitive and natural interaction for humans, it can serve as a distraction from other features of the framework, i.e., topic overviews. Particularly, if users trust the ranker and are confident that they can interact with the ranker effectively, they will focus on the ranked documents and ignore the overview of the corpus in terms of topics. Another source of distraction from topics is the dialog box, which periodically opens and active learning selected documents along with relevant and irrelevant documents to confirm are displayed (Figures 5.2a and 5.2b). This dialog box is identical in the TOPIC condition and the LIST condition—the topic overviews are not displayed. Therefore, users in the TOPIC condition cannot benefit from the topic overview information when they are spending the time to mark documents in the dialog box.

5.6 Summary

In this chapter, we introduced a framework to test the effectiveness of topic models in helping users understand a large document collection, find documents of interest, and find answers to a set of questions from the documents in the context of a real-world use case. We ran a user study to evaluate the effectiveness of topic overviews provided by topic models against an alternative that lacks topic overviews. Our results showed that topics inspired participants to further explore the corpus by searching for information. Self-reported measures showed that participants thought topic information was helpful in doing the task. However, our experiment revealed no significant effect of topic information existence in helping users answer the questions, find relevant documents, and train a classifier. We hope that the findings and discussions in this chapter will encourage more empirical studies of the effectiveness of topic models in helping humans browse and understand large document collections.

Chapter 6

Conclusion and Future Directions

In this thesis, we discussed the problem of interpretability in machine learning from an interdisciplinary and human-in-the-loop perspective. We developed and proposed a template for human-subject experiments that isolates and measures the effect of supervised model interpretability on human understanding, behavior, and trust. Additionally, we proposed interactive frameworks that exploit interpretable and unsupervised machine learning models to help users complete real-world tasks. We evaluated the proposed frameworks with controlled user studies. These frameworks have some limitations and can be extended in several aspects to improve user experience and performance.

In Chapter 3, we introduced an experimental template for measuring the effect of several manipulable factors (that are thought to affect interpretability) on users' trust and their abilities to simulate the model's predictions and detect the model's mistakes. The user interactions with supervised regression models in these experiments were very limited. Supporting users' interaction with the model can help us understand human abilities and behaviors better. For example, allowing users to manipulate regression coefficients and/or do feature engineering by removing features or adding interaction features can lead to both better performance of users and better understanding of interpretability.

Our proposed interactive framework in Chapter 4 (ALTO) is currently deployed in the standard workflow of Snagajob,[1] an online employment website, to categorize job postings efficiently. ALTO was evaluated and is currently used in an annotation task that requires an understanding of thematic structure in large corpora. Evaluating the generalizability of ALTO on other similar tasks such as sentiment annotation requires

[1] `https://www.snagajob.com/`

more user studies to understand how humans use topical overviews which are not **directly** and **immediately** helpful for successfully completing a task.

Interactive systems that use machine learning models to guide humans in completing tasks should reduce sources of distraction and possible advert effects of automatic suggestions by models. For example, automatically generated topics in the frameworks introduced in Chapter 4 and Chapter 5 might not be as useful to an expert user and thus lead to distraction and degradation in performance. Allowing users to debug or personalize topics (Hu et al., 2014; Hoque and Carenini, 2015; Fails and Olsen Jr, 2003) in these interfaces can prevent distraction and improve user experience.

Our experiments should encourage more empirical studies in the machine learning community and our findings and insights should help develop the next-generation machine learning models considering humans and interactivity as a factor. We now discuss future directions.

6.1 Future Directions

Extensions to ALTO.

We can further improve ALTO in Chapter 4 to help users gain better and faster understanding of text data. Currently, ALTO limits users to view only $20K$ documents at a time and allows for one label assignment per document. Moreover, the topics are static and do not adapt to better reflect users' labels. Users should have better support for browsing documents and assigning multiple labels. Topics can also improve via supervised topic models such SLDA (Mcauliffe and Blei, 2007), LLDA (Ramage et al., 2009), or SANCHOR (Nguyen et al., 2015) as users add labels. Given that inferring such topics is often slow and not suitable for an interactive system, we first need to evaluate their effectiveness and efficiency with synthetic experiments.[2] Additionally, with slight changes to what the system considers a document, we believe ALTO can be extended to NLP applications other than classification, such as named entity recognition or semantic role labeling, to reduce the annotation effort.

More feedback and involvement of users.

[2] This work is in collaboration with Thang Nguyen and Jordan Boyd-Graber and is in submission.

One shortcoming of the frameworks in Chapter 3 and Chapter 5 is the lack of explicit feedback to users on how they have done so far. In experiments in Chapter 3, we tried to do so in the training phase, where participants saw the actual prices of apartments and how that compared to their predictions of the price. Similarly, in Chapter 5, participants were instructed to use automatically identified relevant documents in the review page as an evaluation of how reasonable they had marked relevant and irrelevant grants. We can provide more feedback followed by clear instructions on what actions users can take to correct an automatic decision. By enabling users to do so, we can both improve users' experience and the systems' performance.

More critical and familiar domains.

With the ubiquity of machine learning in many domains, it is important to evaluate methods and models in the context and domain of interest. Therefore, if the goal is to evaluate the system with non-expert participants, we should select domains that most people are familiar with and care about. The real-estate data that we used for experiments in Chapter 3 would be irrelevant to many participants who do not reside in New York City and the NSF data that we used in Chapter 5 can potentially be hard to understand for participants without background and interest in science. As interpretability is usually a concern in critical domains such as healthcare, it is worth experimenting with these domains that non-expert users are familiar with and are aware of the potential consequences of automatically made prediction.

Measures of trust in the context of machine learning.

Interpretability in machine learning is often motivated by people's trust in models. In Chapter 3, we experimented and measured three different metrics for trust: the amount to which users follow the model, weight of advice, and the amount to which they mimic the model. However, measuring trust of users in different models and in different scenarios is not trivial and remains a persistent problem. While we believe that trust should be measured based on people's abilities and behavior using a task-driven approach, we think the subfield of interpretability in machine learning can benefit from more research on appealing metrics of trust in different scenarios.

Interpretability of different families of models.

Measuring and comparing interpretability of different families of models (e.g., decision trees vs. deep neural networks) is a challenging yet interesting problem that has a close connection to the work in this thesis. We believe similar experiments to the experiments proposes in Chapter 3 can shed light on how humans understand and make decisions with the help of different families of models. However, the first challenge and interesting future direction is to come up with consistent visualizations of models that change as little as possible between conditions (i.e., different families of models) to be able to run controlled studies.

Understanding when and why interpretability is important.

The work in this thesis focused on interpretability motivated by being able to explain **why** and **how** a model makes predictions so that humans can understand how these models will affect people and decide whether to trust them. On the other hand, our user study results in Chapter 3 regarding the cases that the model made highly inaccurate predictions indicated that model transparency lowers abilities of users in detecting model's mistakes. A related psychology research study found that people are more easily convinced if there is an explanation, even if the explanation is bad, compared to when there is no explanation (Langer et al., 1978). We believe the unexpected finding in our study is worth exploring further to understand in what scenarios interpretability can be helpful and potentially, when can interpretability harm people's informed decision making and lead to over-trust.

Bibliography

E Scott Adler and John Wilkerson. Congressional bills project. NSF, 880066:00880061, 2006.

Nikolaos Aletras and Mark Stevenson. Labelling topics using unsupervised graph-based methods. In Association for Computational Linguistics (ACL), pages 631–636, 2014.

Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. Representing topics labels for exploring digital libraries. In Digital Libraries (JCDL), 2014 IEEE/ACM Joint Conference on, pages 239–248. IEEE, 2014.

Daniele Archibugi and Andrea Filippetti. The handbook of global science, technology, and innovation. John Wiley & Sons, 2015.

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpretability via model extraction. arXiv preprint arXiv:1706.09773, 2017.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. Journal of Statistical Software, 67(1):1–48, 2015.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. Journal of Machine Learning Research (JMLR), 3(Jan):993–1022, 2003.

Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. In The Biennial GSCL Conference, pages 31–40, 2009.

Jordan Boyd-Graber, David Mimno, and David Newman. Care and feeding of topic models: Problems, diagnostics, and improvements. Handbook of Mixed Membership Models and Their Applications; CRC Press: Boca Raton, FL, USA, 2014.

Jordan Boyd-Graber, Yuening Hu, and David Mimno. Applications of Topic Models, volume 11 of Foundations and Trends in Information Retrieval. NOW Publishers, 2017. URL `http://www.nowpublishers.com/article/Details/INR-030`.

Leo Breiman. Bagging predictors. Machine Learning, 24(2):123–140, 1996.

Ian Budge. Mapping policy preferences: estimates for parties, electors, and governments, 1945-1998, volume 1. Oxford University Press, 2001.

Bob Carpenter. Lingpipe 4.1.0. http://alias-i.com/lingpipe, 2008.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1721–1730. ACM, 2015.

Allison June-Barlow Chaney and David M Blei. Visualizing topic models. In International Conference on Web andD Soacial Media (ICWSM), 2012.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In Neural Information Processing Systems (NIPS), 2009.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Neural Information Processing Systems (NIPS), pages 2172–2180, 2016.

Christina Christakou, Spyros Vrettos, and Andreas Stafylopatis. A hybrid movie recommender system based on neural networks. International Journal on Artificial Intelligence Tools, 16(05):771–792, 2007.

Jason Chuang, John D Wilkerson, Rebecca Weiss, Dustin Tingley, Brandon M Stewart, Margaret E Roberts, Forough Poursabzi-Sangdeh, Justin Grimmer, Leah Findlater, Jordan Boyd-Graber, et al. Computer-assisted content analysis: Topic models for exploring multiple subjective interpretations. In Neural Information Processing Systems (NIPS) Workshop on Human-Propelled Machine Learning, 2014.

Gregory F Cooper, Constantin F Aliferis, Richard Ambrosino, John Aronis, Bruce G Buchanan, Richard Caruana, Michael J Fine, Clark Glymour, Geoffrey Gordon, Barbara H Hanusa, et al. An evaluation of machine-learning methods for predicting pneumonia mortality. Artificial intelligence in medicine, 9(2): 107–138, 1997.

Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. 2008.

Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G Tollis. Algorithms for drawing graphs: an annotated bibliography. Computational Geometry, 4(5):235–282, 1994.

Berkeley J. Dietvorst, Joseph P. Simmons, and Cade Massey. Algorithm aversion: People erroneously avoid algorithms after seeing them err. Journal of Experimental Psychology: General, 144(1):114–126, 2015.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608, 2017.

Jacob Eisenstein, Duen Horng Chau, Aniket Kittur, and Eric Xing. Topicviz: interactive topic exploration in document collections. In CHI'12 Extended Abstracts on Human Factors in Computing Systems, pages 2177–2182. ACM, 2012.

Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In Proceedings of the 8th international conference on Intelligent user interfaces, pages 39–45. ACM, 2003.

Manaal Faruqui and Chris Dyer. Non-distributional word vector representations. arXiv preprint arXiv:1506.05230, 2015.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. Sparse overcomplete word vector representations. arXiv preprint arXiv:1506.02004, 2015.

Paul Felt, Eric Ringger, Jordan Boyd-Graber, and Kevin Seppi. Making the most of crowdsourced document annotations: confused supervised LDA. In Proceedings of the Nineteenth Conference on Computational Natural Language Learning, pages 194–203, 2015.

Joseph L Fleiss. Measuring nominal scale agreement among many raters. Psychological bulletin, 76(5):378, 1971.

John Fox. Applied regression analysis, linear models, and related methods. Sage Publications, Inc, 1997.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1):119–139, 1997.

Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. Machine learning, 28(2):133–168, 1997.

Rocio Garcia-Retamero and Edward T Cokely. Communicating health risks with visual aids. Current Directions in Psychological Science, 22(5):392–399, 2013.

Matthew J Gardner, Joshua Lutes, Jeff Lund, Josh Hansen, Dan Walker, Eric Ringger, and Kevin Seppi. The topic browser: An interactive tool for browsing topic models. In Neural Information Processing Systems (NIPS) Workshop on Challenges of Data Visualization, volume 2, 2010.

Francesca Gino and Don A. Moore. Effects of task difficulty on use of advice. Journal of Behavioral Decision Making, 20(1):21–35, 2007.

Alyssa Glass, Deborah L McGuinness, and Michael Wolverton. Toward establishing trust in adaptive agents. In Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI), 2008.

Ben Goldacre. Make journals report clinical trials properly: there is no excuse for the shoddy practice of allowing researchers to change outcomes and goals without saying so. Nature, 530(7588):7–8, 2016.

Bryce Goodman and Seth Flaxman. European Union regulations on algorithmic decision-making and a ”right to explanation”. arXiv preprint arXiv:1606.08813, 2016.

Justin Grimmer and Brandon M Stewart. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political Analysis, page mps028, 2013.

Fangjian Guo, Charles Blundell, Hanna Wallach, and Katherine Heller. The Bayesian echo chamber: Modeling social influence via linguistic accommodation. In Artificial Intelligence and Statistics, pages 315–323, 2015.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. Revisiting embedding features for simple semi-supervised learning. In Empirical Methods in Natural Language Processing (EMNLP), pages 110–120, 2014.

Todd M Gureckis, Jay Martin, John McDonnell, Alexander S Rich, Doug Markant, Anna Coenen, David Halpern, Jessica B Hamrick, and Patricia Chan. psiTurk: An open-source framework for conducting replicable behavioral experiments online. Behavior Research Methods, 48(3):829–842, 2016.

Robbie Haertel, Eric K. Ringger, Kevin D. Seppi, James L. Carroll, and Peter McClanahan. Assessing the costs of sampling methods in active learning for annotation. 2008.

David Hall, Daniel Jurafsky, and Christopher D Manning. Studying the history of ideas using topic models. In Empirical Methods in Natural Language Processing (EMNLP), pages 363–371. Association for Computational Linguistics, 2008.

Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. Advances in psychology, 52:139–183, 1988.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. In The elements of statistical learning, pages 485–585. Springer, 2009.

M.A. Hearst and J.O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. 1996.

Dustin Hillard, Stephen Purpura, and John Wilkerson. Computer-assisted topic classification for mixed-methods social science research. Journal of Information Technology & Politics, 4(4):31–46, 2008.

Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivizing high quality crowdwork. In Proceedings of the 24th International Conference on World Wide Web, pages 419–429. International World Wide Web Conferences Steering Committee, 2015.

Ulrich Hoffrage and Gerd Gigerenzer. Using natural frequencies to improve diagnostic inferences. Academic medicine, 73(5):538–540, 1998.

Jake M Hofman, Amit Sharma, and Duncan J Watts. Prediction and explanation in social systems. Science, 355(6324):486–488, 2017.

Enamul Hoque and Giuseppe Carenini. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In Proceedings of the 20th International Conference on Intelligent User Interfaces, pages 169–180. ACM, 2015.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24(6):417, 1933.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In North American Chapter of the Association for Computational Linguistics (NAACL) Workshop on Active Learning for Natural Language Processing, HLT '09, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. Machine learning, 95(3):423–469, 2014.

Lawrence Hubert and Phipps Arabie. Comparing partitions. Journal of classification, 2(1):193–218, 1985.

Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. Unsupervised graph-based topic labelling using dbpedia. In Proceedings of the sixth ACM international conference on Web search and data mining, pages 465–474. ACM, 2013.

Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Lam Ngoc Tran, and Karl Aberer. An evaluation of aggregation techniques in crowdsourcing. In International Conference on Web Information Systems Engineering, pages 1–15. Springer, 2013.

Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. Decision Support Systems, 51(1):141–154, 2011.

Rebecca Hwa. Sample selection for statistical parsing. Computational linguistics, 30(3):253–276, 2004.

Mohit Iyyer, Peter Enns, Jordan L Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In Association for Computational Linguistics (ACL), pages 1113–1122, 2014.

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. ACM computing surveys (CSUR), 31(3):264–323, 1999.

Nitin Jindal and Bing Liu. Review spam detection. In Proceedings of the 16th international conference on World Wide Web, pages 1189–1190. ACM, 2007.

Jongbin Jung, Connor Concannon, Ravi Shro, Sharad Goel, and Daniel G. Goldstein. Simple rules for complex decisions. arXiv preprint arXiv:1702.04690, 2017.

Ece Kamar and Eric Horvitz. Incentives for truthful reporting in crowdsourcing. In Proceedings of the 11th international conference on autonomous agents and multiagent systems-volume 3, pages 1329–1330. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

Matthew Kay, Tara Kola, Jessica R Hullman, and Sean A Munson. When (ish) is my bus?: User-centered visualizations of uncertainty in everyday, mobile predictive systems. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pages 5092–5103. ACM, 2016.

Daniel A Keim. Designing pixel-oriented visualization techniques: Theory and applications. IEEE Transactions on visualization and computer graphics, 6(1):59–78, 2000.

Edward F Kelly and Philip J Stone. Computer recognition of English word senses, volume 13. North-Holland, 1975.

Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In Proceedings of the 20th international conference on Computational Linguistics, page 1367. Association for Computational Linguistics, 2004.

Yoon Kim. Convolutional neural networks for sentence classification. Empirical Methods in Natural Language Processing (EMNLP), 2014.

Hans-Dieter Klingemann, Andrea Volkens, Judith Bara, Ian Budge, Michael D McDonald, et al. Mapping policy preferences II: estimates for parties, electors, and governments in Eastern Europe, European Union, and OECD 1990-2003. Oxford University Press Oxford, 2006.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In International Conference on Machine Learning (ICML), 2017.

Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160:3–24, 2007.

Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In Proceedings of the 20th International Conference on Intelligent User Interfaces, pages 126–137. ACM, 2015.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & explorable approximations of black box models. arXiv preprint arXiv:1707.01154, 2017.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. biometrics, pages 159–174, 1977.

Ken Lang. 20 newsgroups data set, 2007. http://www.ai.mit.edu/people/jrennie/20Newsgroups/.

Ellen J Langer, Arthur Blank, and Benzion Chanowitz. The mindlessness of ostensibly thoughtful action: The role of" placebic" information in interpersonal interaction. Journal of personality and social psychology, 36(6):635, 1978.

Jey Han Lau, David Newman, Sarvnaz Karimi, and Timothy Baldwin. Best topic word selection for topic labelling. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 605–613. Association for Computational Linguistics, 2010.

Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In Association for Computational Linguistics (ACL), pages 1536–1545. Association for Computational Linguistics, 2011.

Jey Han Lau, David Newman, and Timothy Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In European Chapter of the Association for Computational Linguistics (EACL), 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436, 2015.

Tak Yeon Lee, Alison Smith, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. International Journal of Human-Computer Studies, 2017. URL docs/2017_ijhcs_human_touch.pdf.

David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 3–12. Springer-Verlag New York, Inc., 1994.

Brian Y Lim, Anind K Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), 2009.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. Unsupervised POS induction with word embeddings. arXiv preprint arXiv:1503.06760, 2015.

Zachary C Lipton. The mythos of model interpretability. arXiv preprint arXiv:1606.03490, 2016.

Jennifer M. Logg. Theory of machine: When do people rely on algorithms? Harvard Business School NOM Unit Working Paper No. 17-086, 2017.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In Knowledge Discovery and Data Mining (KDD), 2012.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In Knowledge Discovery and Data Mining (KDD), 2013.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In Neural Information Processing Systems(NIPS), 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. Journal of Machine Learning Research (JMLR), 9(Nov):2579–2605, 2008.

Davide Magatti, Silvia Calegari, Davide Ciucci, and Fabio Stella. Automatic labeling of topics. In Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on, pages 1227–1232. IEEE, 2009.

Alireza Makhzani and Brendan Frey. K-sparse autoencoders. arXiv preprint arXiv:1312.5663, 2013.

Naoki Abe Hiroshi Mamitsuka et al. Query learning strategies using boosting and bagging. In International Conference on Machine Learning (ICML), volume 1. Morgan Kaufmann Pub, 1998.

Jon D Mcauliffe and David M Blei. Supervised topic models. In Neural Information Processing Systems (NIPS), 2007.

Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://www.cs.umass.edu/ mccallum/mallet, 2002.

Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 490–499. ACM, 2007.

Marina Meilă. Comparing clusterings by the variation of information. In Learning theory and kernel machines, pages 173–187. Springer, 2003.

Prem Melville, Wojciech Gryc, and Richard D Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1275–1284. ACM, 2009.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Neural Information Processing Systems (NIPS), 2013b.

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

Fred Morstatter and Huan Liu. A novel measure for coherence in statistical topic models. In Association for Computational Linguistics (ACL), volume 2, pages 543–548, 2016.

Brian Murphy, Partha Talukdar, and Tom Mitchell. Learning effective and interpretable semantic models using non-negative sparse embedding. Proceedings of COLING 2012, pages 1933–1950, 2012.

Chris Musialek, Philip Resnik, and S. Andrew 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.

Roberto Navigli. Word sense disambiguation: A survey. ACM Computing Surveys (CSUR), 41(2):10, 2009.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In North American Chapter of the Association for Computational Linguistics (NAACL), pages 100–108. Association for Computational Linguistics, 2010.

Grace Ngai and David Yarowsky. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In Association for Computational Linguistics (ACL), pages 117–125. Association for Computational Linguistics, 2000.

Thang Nguyen, Jordan Boyd-Graber, Jeff Lund, Kevin Seppi, and Eric Ringger. Is your anchor going up or down? Fast and accurate supervised topic models. In North American Chapter of the Association for Computational Linguistics (NAACL), 2015.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. Lexical and hierarchical topic regression. In Neural Information Processing Systems (NIPS), 2013.

Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, Deborah A Cai, Jennifer E Midberry, and Yuanxin Wang. Modeling topic control to detect influence in conversations using nonparametric topic models. Machine Learning, 95(3):381–421, 2014a.

Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, and Jonathan Chang. Learning a concept hierarchy from multi-labeled documents. In Neural Information Processing Systems (NIPS), 2014b.

Dilek Önkal, Paul Goodwin, Mary Thomson, and Sinan Gönül. The relative influence of advice from human experts and statistical methods on forecast adjustments. Journal of Behavioral Decision Making, 22:390–409, 2009.

Miles Osborne and Jason Baldridge. Ensemble-based active learning for parse selection. In North American Chapter of the Association for Computational Linguistics (NAACL), pages 89–96. Citeseer, 2004.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In Empirical Methods in Natural Language Processing (EMNLP), pages 79–86. Association for Computational Linguistics, 2002.

Michael Paul and Roxana Girju. A two-dimensional topic-aspect model for discovering multi-faceted topics. In AAAI, 2010.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research (JMLR), 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), 2014.

István Pilászy and Domonkos Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In Proceedings of the third ACM conference on Recommender systems, pages 93–100. ACM, 2009.

Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Leah Findlater, and Kevin Seppi. Alto: Active learning with topic overviews for speeding label induction and document labeling. In Association for Computational Linguistics (ACL), 2016.

Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna M. Wallach. Manipulating and measuring model interpretability. In Neural Information Processing Systems (NIPS) workshop on Transparent and Interpretable Machine Learning in Safety Critical Environments, 2017.

Vadakkedathu Rajan, Mark Wegman, Richard Segal, Jason Crawford, Jeffrey Kephart, and Shlomo Hershkop. Detecting spam email using multiple spam classifiers, 2006. US Patent App. 11/029,069.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In Empirical Methods in Natural Language Processing (EMLNP), pages 248–256. Association for Computational Linguistics, 2009.

Nitin Ramrakhiyani, Sachin Pawar, Swapnil Hingmire, and Girish Palshikar. Measuring topic coherence through optimal word buckets. In European Chapter of the Association for Computational Linguistics (EACL), volume 2, pages 437–442, 2017.

William M Rand. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66(336):846–850, 1971.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In Knowledge Discovery and Data Mining (KDD), 2016.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through Monte Carlo estimation of error reduction. International Conference on Machine Learning (ICML), 2001.

S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3):660–674, 1991.

J. Saldana. The Coding Manual for Qualitative Researchers. SAGE Publications, 2012. ISBN 9781446271421. URL `https://books.google.de/books?id=V3tTG4jvgFkC`.

Lutfi Kerem Senel, Ihsan Utlu, Veysel Yucesoy, Aykut Koc, and Tolga Cukur. Semantic structure and interpretability of word embeddings. arXiv preprint arXiv:1711.00331, 2017.

Burr Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In Empirical Methods in Natural Language Processing (EMNLP), 2011.

Burr Settles. Active learning. Long Island, NY: Morgan & Clay Pool, 2012.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In Neural Information Processing Systems (NIPS), pages 1289–1296, 2008.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In Workshop on Computational learning theory, pages 287–294. ACM, 1992.

Nihar Shah and Dengyong Zhou. No oops, you wont do it again: Mechanisms for self-correction in crowdsourcing. In International Conference on Machine Learning (ICML), pages 1–10, 2016.

Claude Elwood Shannon. A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review, 5(1):3–55, 2001.

Carson Sievert and Kenneth E Shirley. Ldavis: A method for visualizing and interpreting topics. In workshop on interactive language learning, visualization, and interfaces, pages 63–70, 2014.

Alison Smith, Jason Chuang, Yuening Hu, Jordan Boyd-Graber, and Leah Findlater. Concurrent visualization of relationships between words and topics in topic models. In Workshop on Interactive Language Learning, Visualization, and Interfaces, pages 79–82, 2014.

Alison Smith, Tak Yeon Lee, Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Kevin Seppi, Niklas Elmqvist, and Leah Findlater. Human-centered and interactive: Expanding the impact of topic models. In CHI Human Centred Machine Learning Workshop, 2016.

Alison Smith, Tak Yeon Lee, Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Kevin Seppi, Niklas Elmqvist, and Leah Findlater. Evaluating visual representations for topic understanding and their effects on manually generated labels. Transactions of the Association for Computational Linguistics (TACL), 5: 1–15, 2017. URL `docs/2017_tacl_eval_tm_viz.pdf`.

Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. User-centered design and evaluation of a human-in-the-loop topic modeling system. In Intelligent User Interfaces, 2018. URL `docs/2018_iui_itm.pdf`.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. 2008.

Ji Soo Yi, Rachel Melton, John Stasko, and Julie A Jacko. Dust & magnet: multivariate information visualization using a magnet metaphor. Information Visualization, 4(4):239–256, 2005.

David Spiegelhalter, Mike Pearson, and Ian Short. Visualizing uncertainty about the future. science, 333 (6048):1393–1400, 2011.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research (JMLR), 3:583–617, 2003.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. arXiv preprint arXiv:1711.08792, 2017.

Edmund M Talley, David Newman, David Mimno, Bruce W Herr II, Hanna M Wallach, Gully APC Burns, AG Miriam Leenders, and Andrew McCallum. Database of NIH grants using machine-learned categories and graphical clustering. Nature Methods, 8(6):443, 2011.

Soon Tee Teoh and Kwan-Liu Ma. Paintingclass: interactive construction, visualization and exploration of decision trees. In Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 667–672. ACM, 2003.

Katrin Tomanek, Joachim Wermter, and Udo Hahn. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. Machine Learning Journal, 102(3):349–391, 2016.

Jan Vanthienen and Geert Wets. From decision tables to expert system shells. Data & Knowledge Engineering, 13(3):265–282, 1994.

Paul MB Vitányi, Frank J Balbach, Rudi L Cilibrasi, and Ming Li. Normalized information distance. In Information theory and statistical learning, pages 45–82. Springer, 2009.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. Hyperlex: A large-scale evaluation of graded lexical entailment. Computational Linguistics, 43(4):781–835, 2017.

Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In International Conference on Machine Learning (ICML), pages 1105–1112. ACM, 2009.

Xiaojun Wan and Tianming Wang. Automatic labeling of topic models using text summaries. In Association for Computational Linguistics (ACL), volume 1, pages 2297–2305, 2016.

Yongqiao Wang, Shouyang Wang, and Kin Keung Lai. A new fuzzy support vector machine to evaluate credit risk. IEEE Transactions on Fuzzy Systems, 13(6):820–831, 2005.

Martin Wattenberg, Fernanda Viégas, and Moritz Hardt. Attacking discrimination with smarter machine learning. 2016.

Xing Wei and W Bruce Croft. LDA-based document models for ad-hoc retrieval. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 178–185. ACM, 2006.

Kilian Q Weinberger, Fei Sha, and Lawrence K Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In Proceedings of the twenty-first international conference on Machine learning, page 106. ACM, 2004.

Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 143–146. ACM, 2011.

Ilan Yaniv. Receiving other people's advice: Influence and benefit. Organizational Behavior and Human Decision Processes, 93:1–13, 2004.

Lean Yu, Shouyang Wang, and Kin Keung Lai. Credit risk assessment with a multistage neural network ensemble learning approach. Expert systems with applications, 34(2):1434–1444, 2008.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In European conference on computer vision, pages 818–833. Springer, 2014.

Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pages 334–342. ACM, 2001.

Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, University of Minnesota, 2001.

Jun Zhu, Amr Ahmed, and Eric P. Xing. MedLDA: maximum margin supervised topic models for regression and classification. 2009. ISBN 978-1-60558-516-1. doi: http://doi.acm.org/10.1145/1553374.1553535.

Jun Zhu, Amr Ahmed, and Eric P Xing. MedLDA: maximum margin supervised topic models. Journal of Machine Learning Research (JMLR), 13(1):2237–2278, 2012.

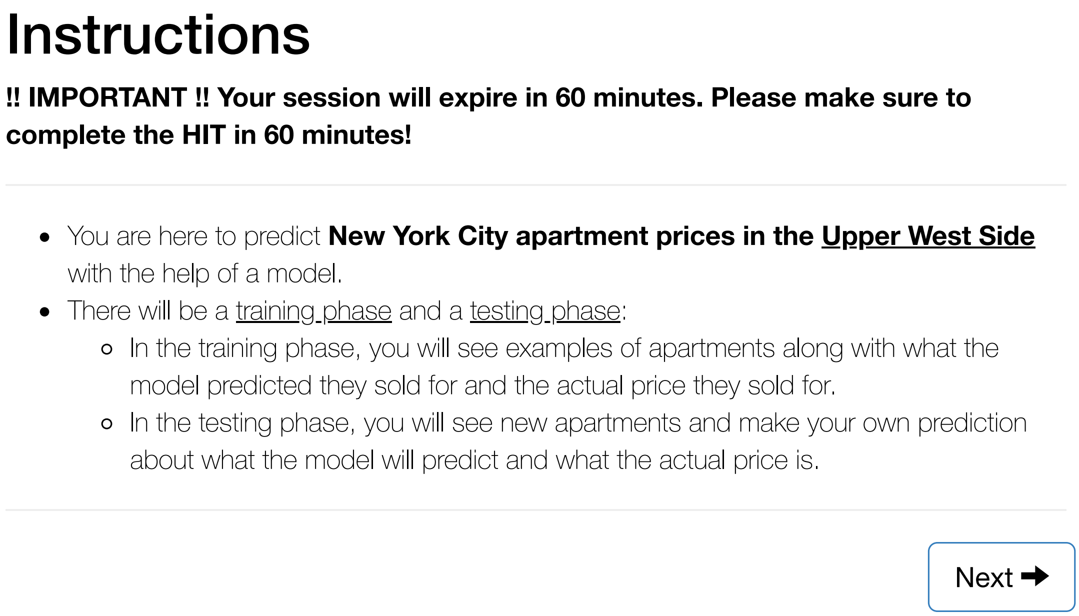
Xiaojin Zhu. Semi-supervised learning literature survey. 2005.

Appendix A

Chapter 3 Study Material

A.1 Instructions for the First Experiment

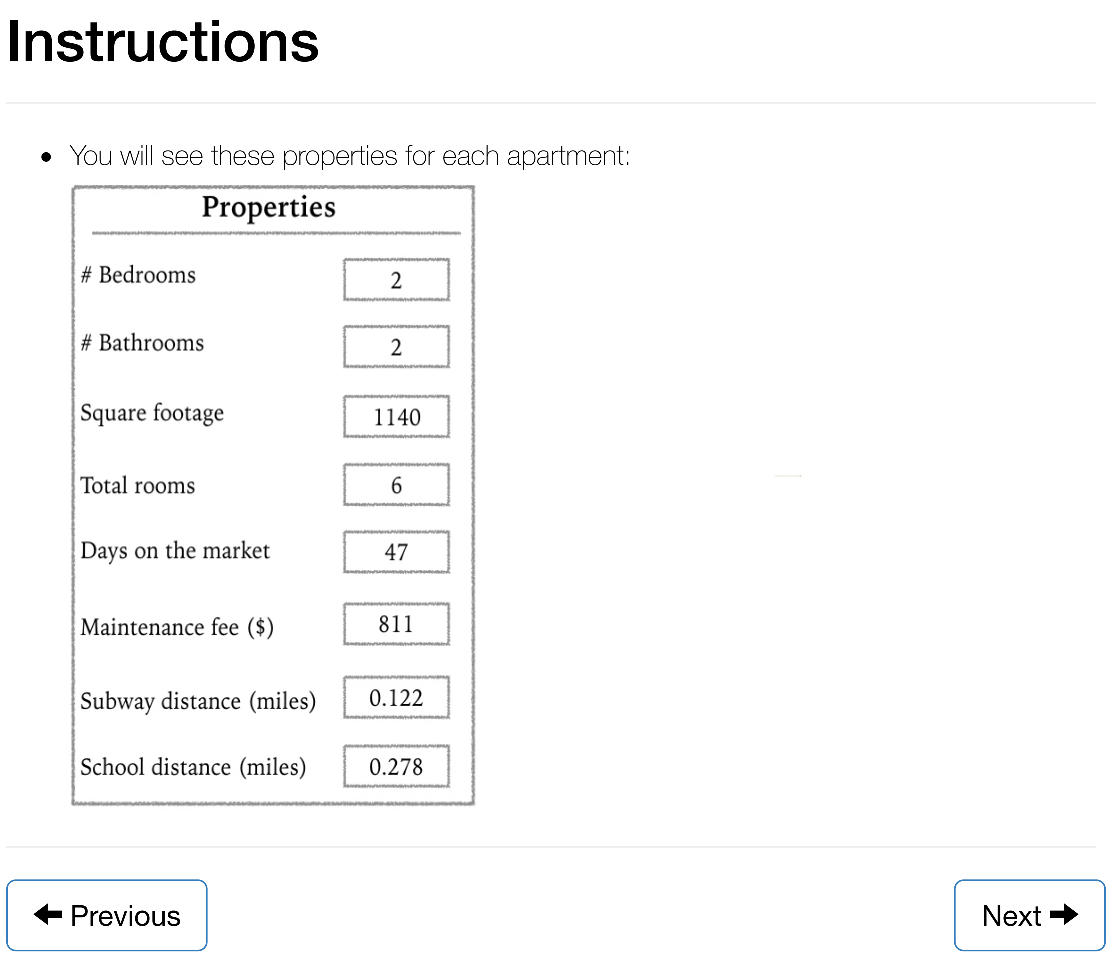
The following instructions were shown to participants assigned to the CLEAR-2 condition in our first experiment on Mechanical Turk. The instructions for other conditions and experiments were adapted from these instructions with minimal changes.



Instructions

!! IMPORTANT !! Your session will expire in 60 minutes. Please make sure to complete the HIT in 60 minutes!

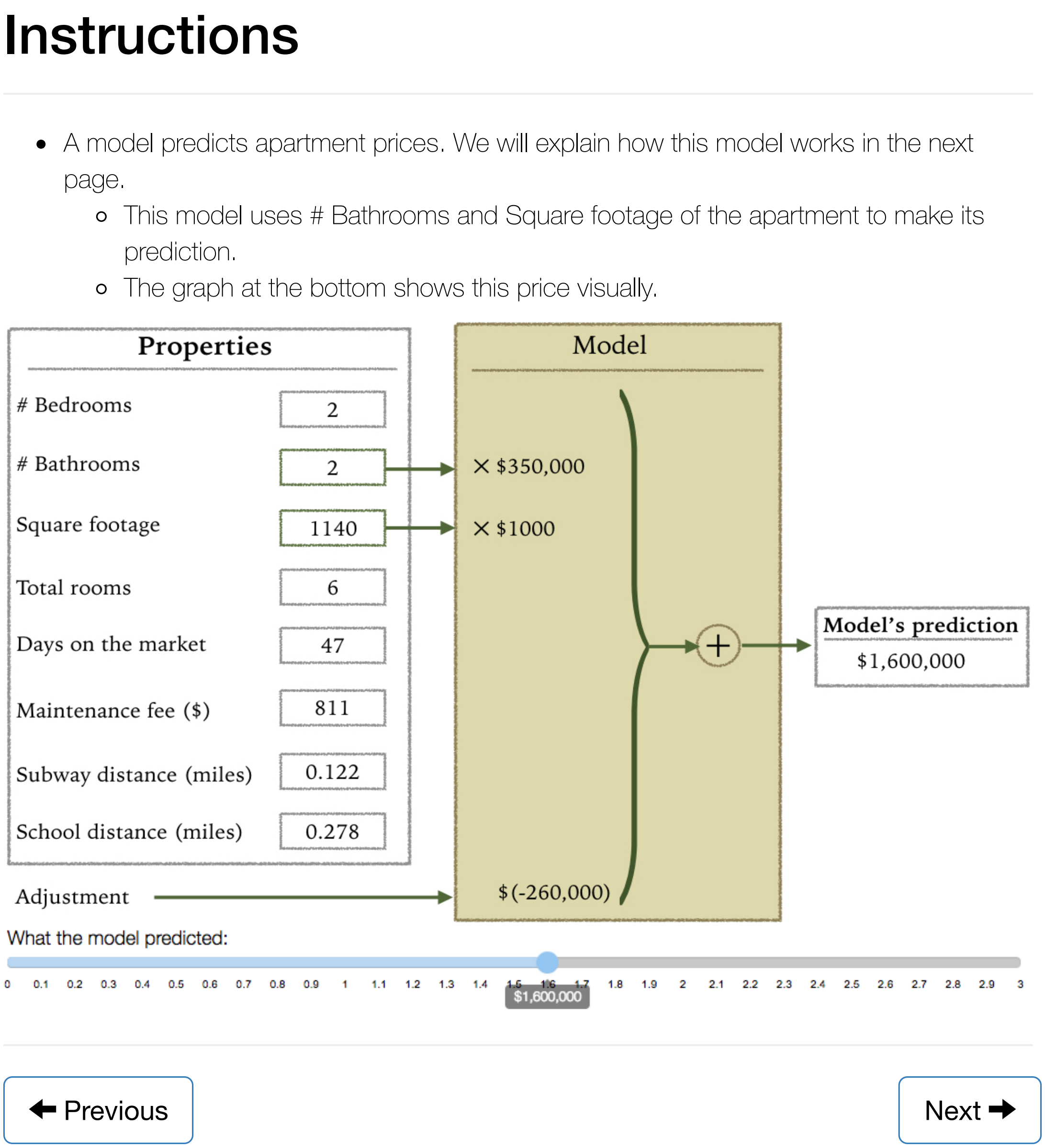
- You are here to predict **New York City apartment prices in the Upper West Side** with the help of a model.
- There will be a training phase and a testing phase:
  - In the training phase, you will see examples of apartments along with what the model predicted they sold for and the actual price they sold for.
  - In the testing phase, you will see new apartments and make your own prediction about what the model will predict and what the actual price is.



Instructions

- You will see these properties for each apartment:

Properties	
# Bedrooms	2
# Bathrooms	2
Square footage	1140
Total rooms	6
Days on the market	47
Maintenance fee ($)	811
Subway distance (miles)	0.122
School distance (miles)	0.278



Instructions

- A model predicts apartment prices. We will explain how this model works in the next page.
    - This model uses # Bathrooms and Square footage of the apartment to make its prediction.
    - The graph at the bottom shows this price visually.

Properties

# Bedrooms	2
# Bathrooms	2
Square footage	1140
Total rooms	6
Days on the market	47
Maintenance fee ($)	811
Subway distance (miles)	0.122
School distance (miles)	0.278

Adjustment

Model

× $350,000

× $1000

$(-260,000)

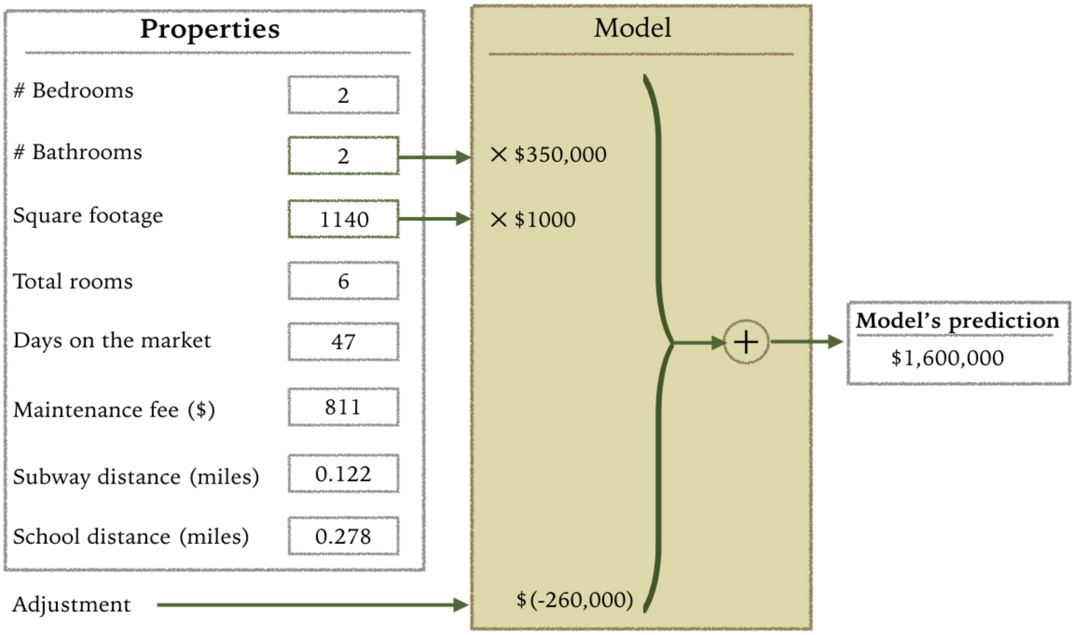
Model's prediction

$1,600,000

What the model predicted:

Instructions

- Here is how the model has made its prediction:
- Each bathroom is worth $350,000. Therefore, $350,000 is multiplied by the number of bathrooms and added to the price. This is repeated for Square footage. Finally, **the adjustment factor of $260,000 is subtracted** and a price is predicted.

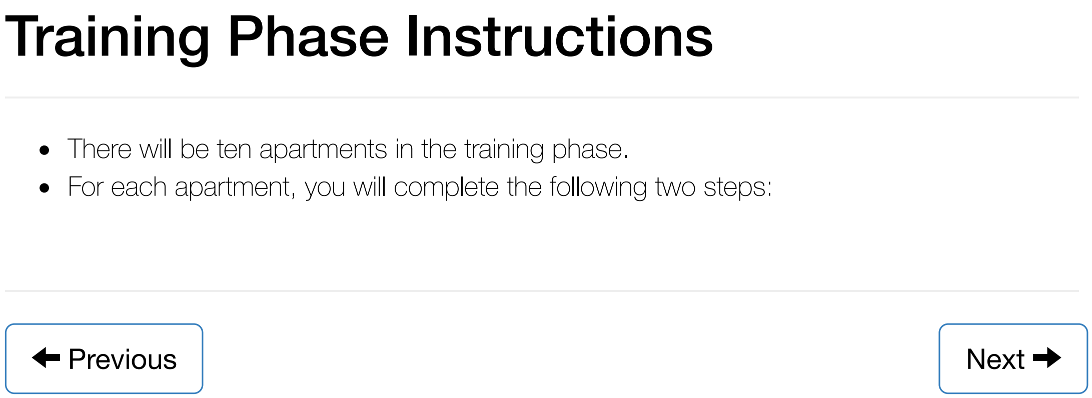


$$\underbrace{[2 \times 350{,}000]}_{700{,}000} + \underbrace{[1140 \times 1000]}_{1{,}140{,}000} + \underbrace{[-260{,}000]}_{[-260{,}000]}$$

$$\approx 1{,}600{,}000$$

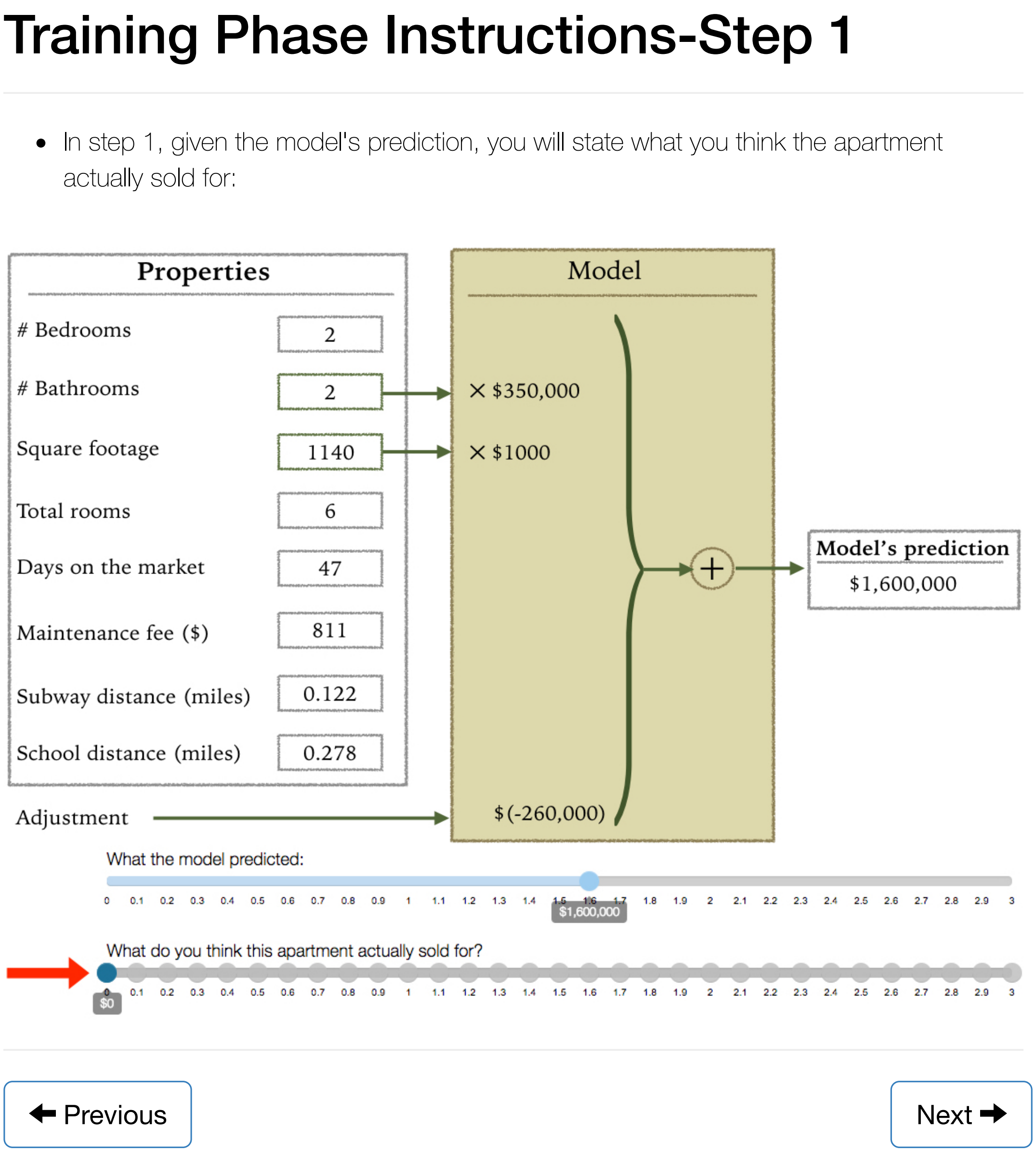
← Previous

Next →



Training Phase Instructions

- There will be ten apartments in the training phase.
- For each apartment, you will complete the following two steps:



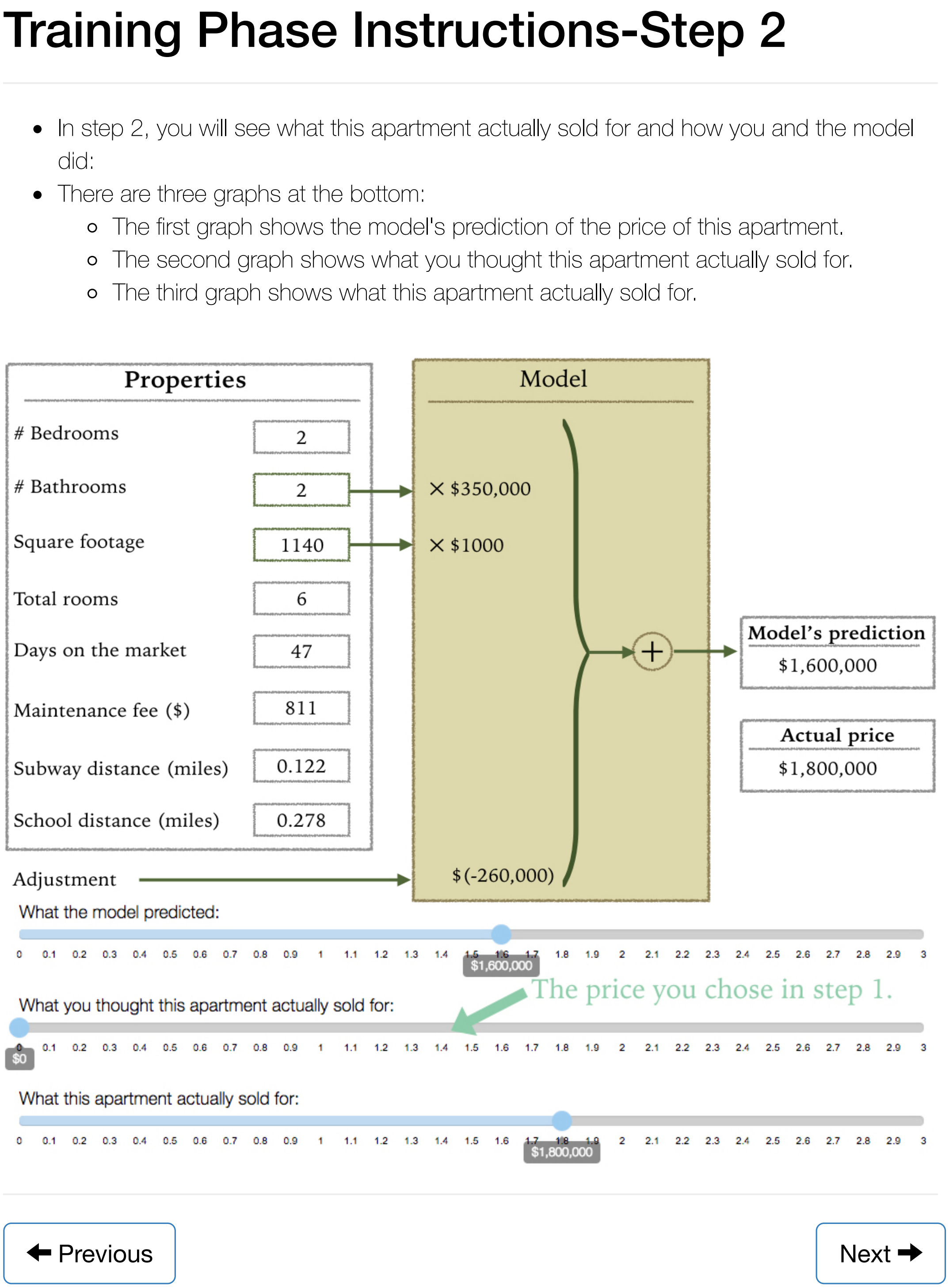
Training Phase Instructions-Step 1

- In step 1, given the model's prediction, you will state what you think the apartment actually sold for:

Properties	
# Bedrooms	2
# Bathrooms	2
Square footage	1140
Total rooms	6
Days on the market	47
Maintenance fee ($)	811
Subway distance (miles)	0.122
School distance (miles)	0.278

What the model predicted:

What do you think this apartment actually sold for?



Training Phase Instructions-Step 2

- In step 2, you will see what this apartment actually sold for and how you and the model did:
- There are three graphs at the bottom:
  - The first graph shows the model's prediction of the price of this apartment.
  - The second graph shows what you thought this apartment actually sold for.
  - The third graph shows what this apartment actually sold for.

Properties	
# Bedrooms	2
# Bathrooms	2
Square footage	1140
Total rooms	6
Days on the market	47
Maintenance fee ($)	811
Subway distance (miles)	0.122
School distance (miles)	0.278

Model

× $350,000

× $1000

Adjustment $(-260,000)

Model's prediction $1,600,000

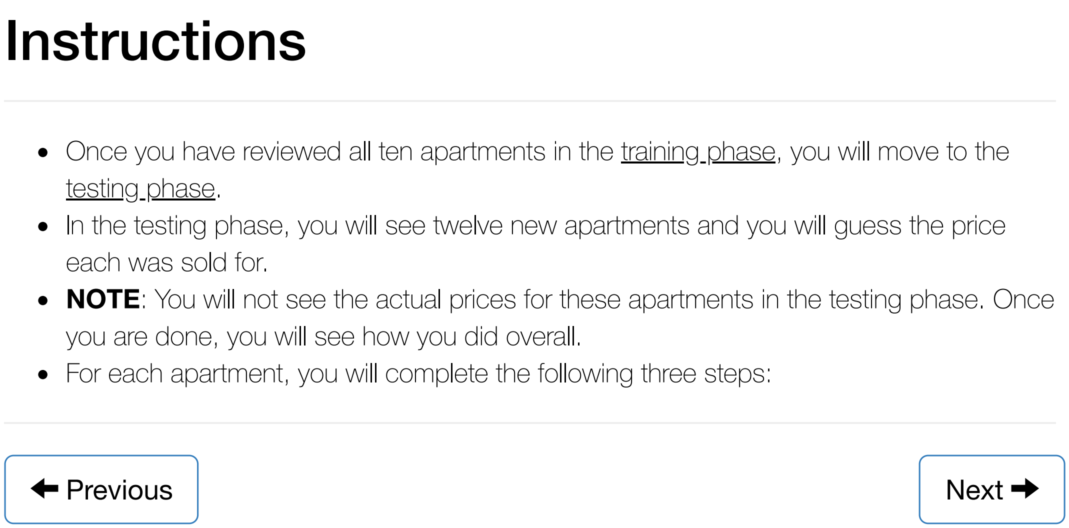
Actual price $1,800,000

What the model predicted: $1,600,000

What you thought this apartment actually sold for: $0

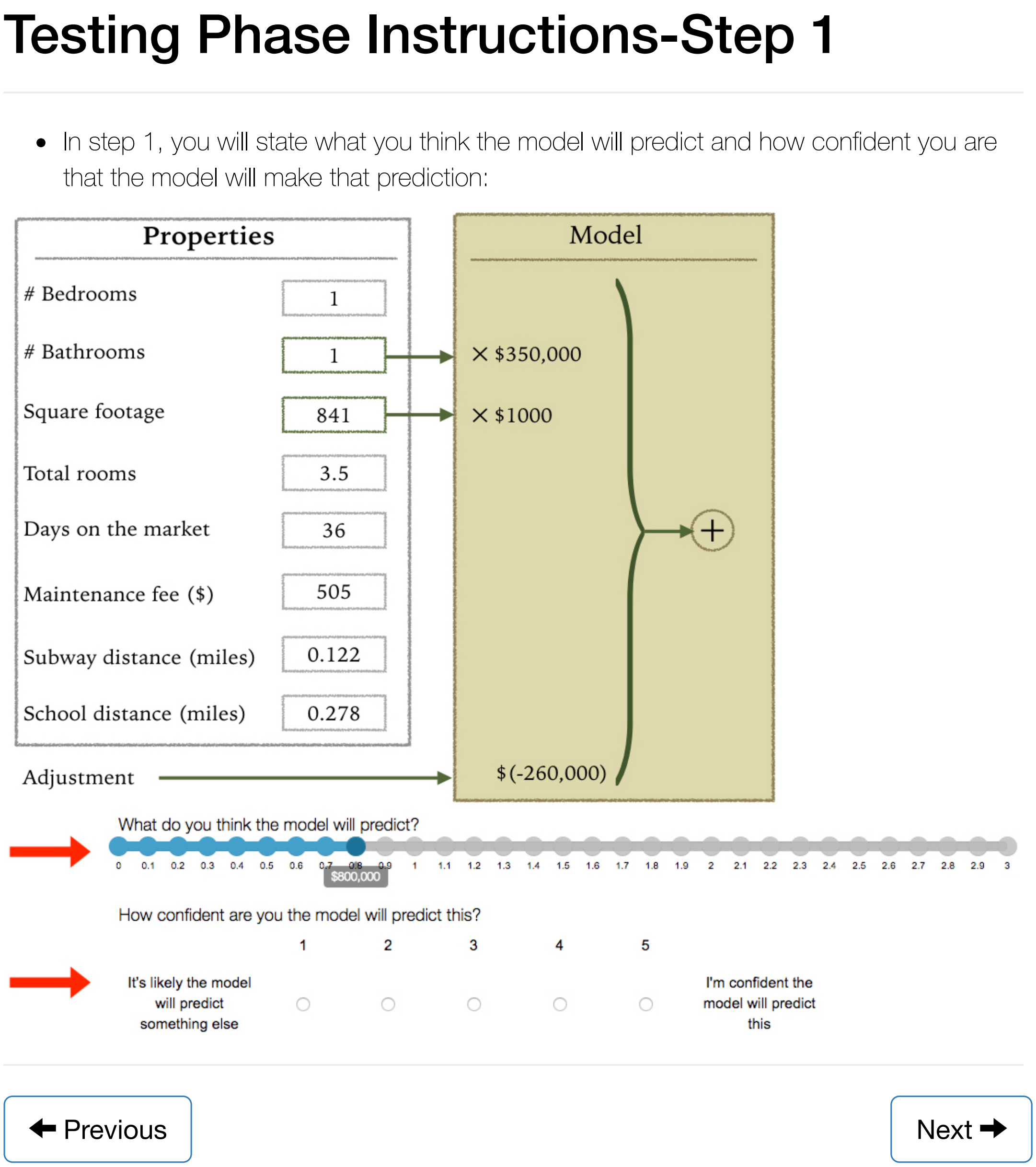
The price you chose in step 1.

What this apartment actually sold for: $1,800,000



Instructions

- Once you have reviewed all ten apartments in the training phase, you will move to the testing phase.
- In the testing phase, you will see twelve new apartments and you will guess the price each was sold for.
- **NOTE**: You will not see the actual prices for these apartments in the testing phase. Once you are done, you will see how you did overall.
- For each apartment, you will complete the following three steps:



Testing Phase Instructions-Step 1

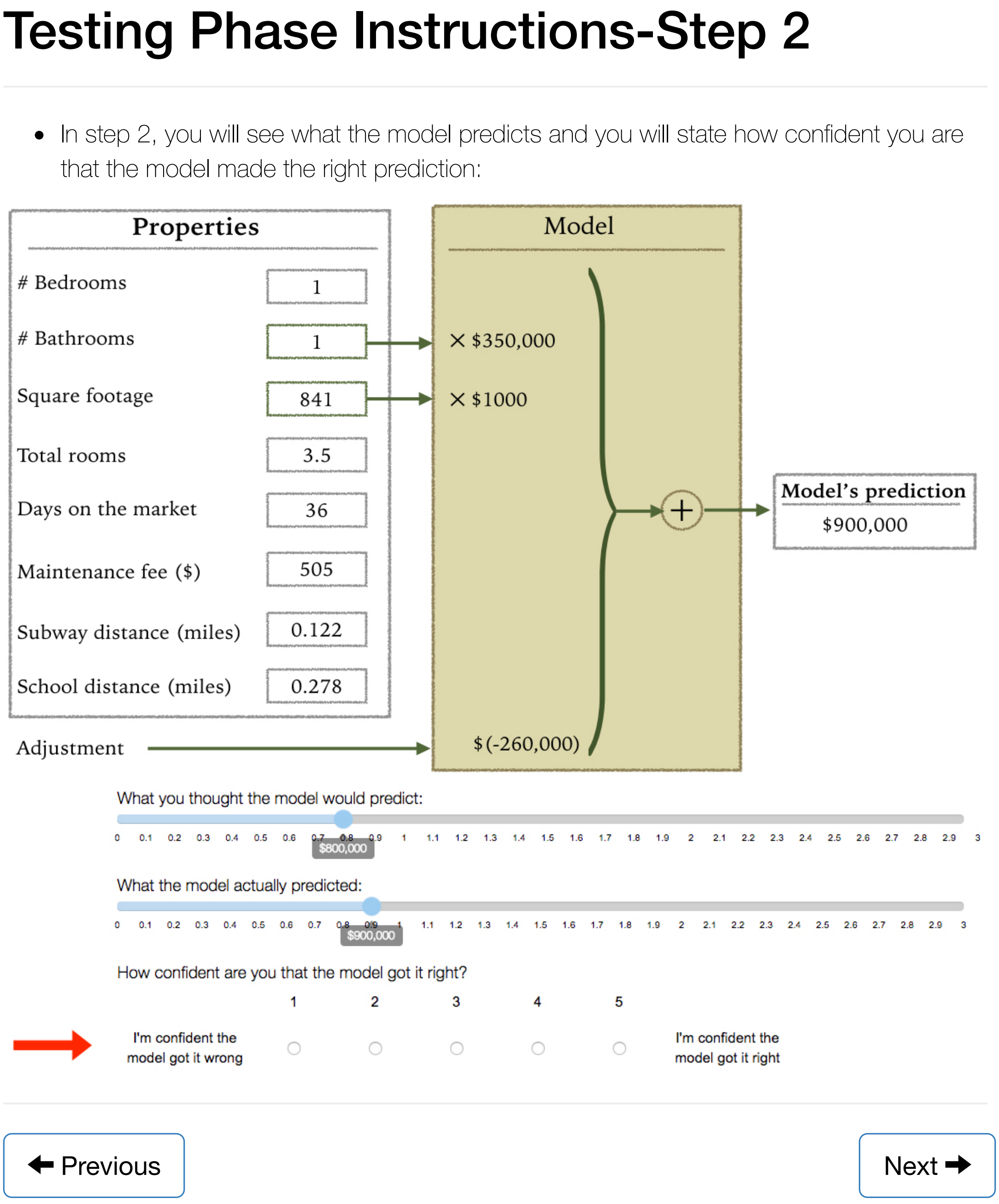
- In step 1, you will state what you think the model will predict and how confident you are that the model will make that prediction:

Properties	
# Bedrooms	1
# Bathrooms	1
Square footage	841
Total rooms	3.5
Days on the market	36
Maintenance fee ($)	505
Subway distance (miles)	0.122
School distance (miles)	0.278

What do you think the model will predict?

How confident are you the model will predict this?

	1	2	3	4	5	
It's likely the model will predict something else	○	○	○	○	○	I'm confident the model will predict this



Testing Phase Instructions-Step 2

- In step 2, you will see what the model predicts and you will state how confident you are that the model made the right prediction:

Properties	
# Bedrooms	1
# Bathrooms	1
Square footage	841
Total rooms	3.5
Days on the market	36
Maintenance fee ($)	505
Subway distance (miles)	0.122
School distance (miles)	0.278

Adjustment

Model

× $350,000

× $1000

$(-260,000)

+

Model's prediction

$900,000

What you thought the model would predict:

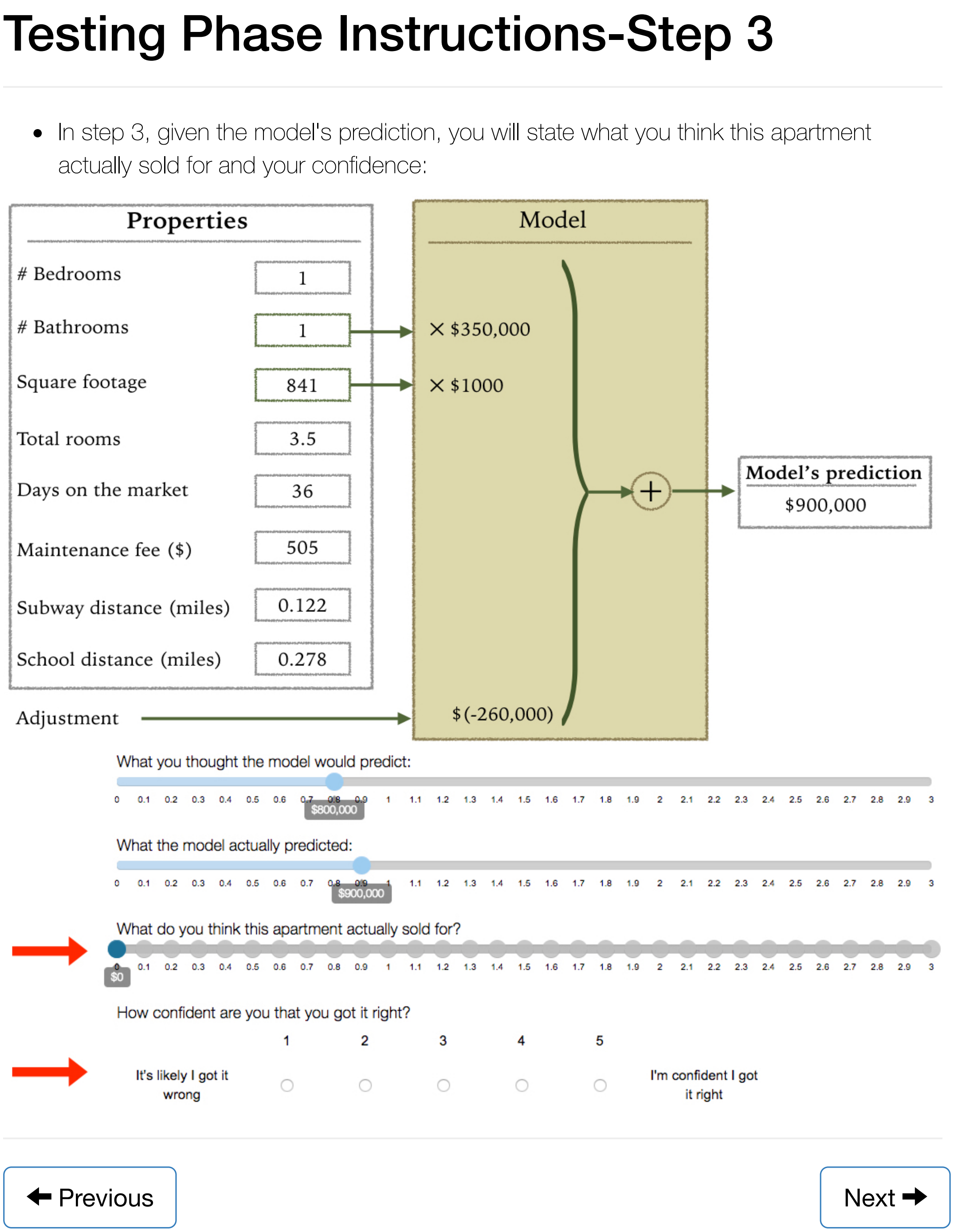
$800,000

What the model actually predicted:

$900,000

How confident are you that the model got it right?

	1	2	3	4	5	
I'm confident the model got it wrong	○	○	○	○	○	I'm confident the model got it right



Testing Phase Instructions-Step 3

- In step 3, given the model's prediction, you will state what you think this apartment actually sold for and your confidence:

Properties	
# Bedrooms	1
# Bathrooms	1
Square footage	841
Total rooms	3.5
Days on the market	36
Maintenance fee ($)	505
Subway distance (miles)	0.122
School distance (miles)	0.278

Adjustment

Model

× $350,000

× $1000

$(-260,000)

Model's prediction

$900,000

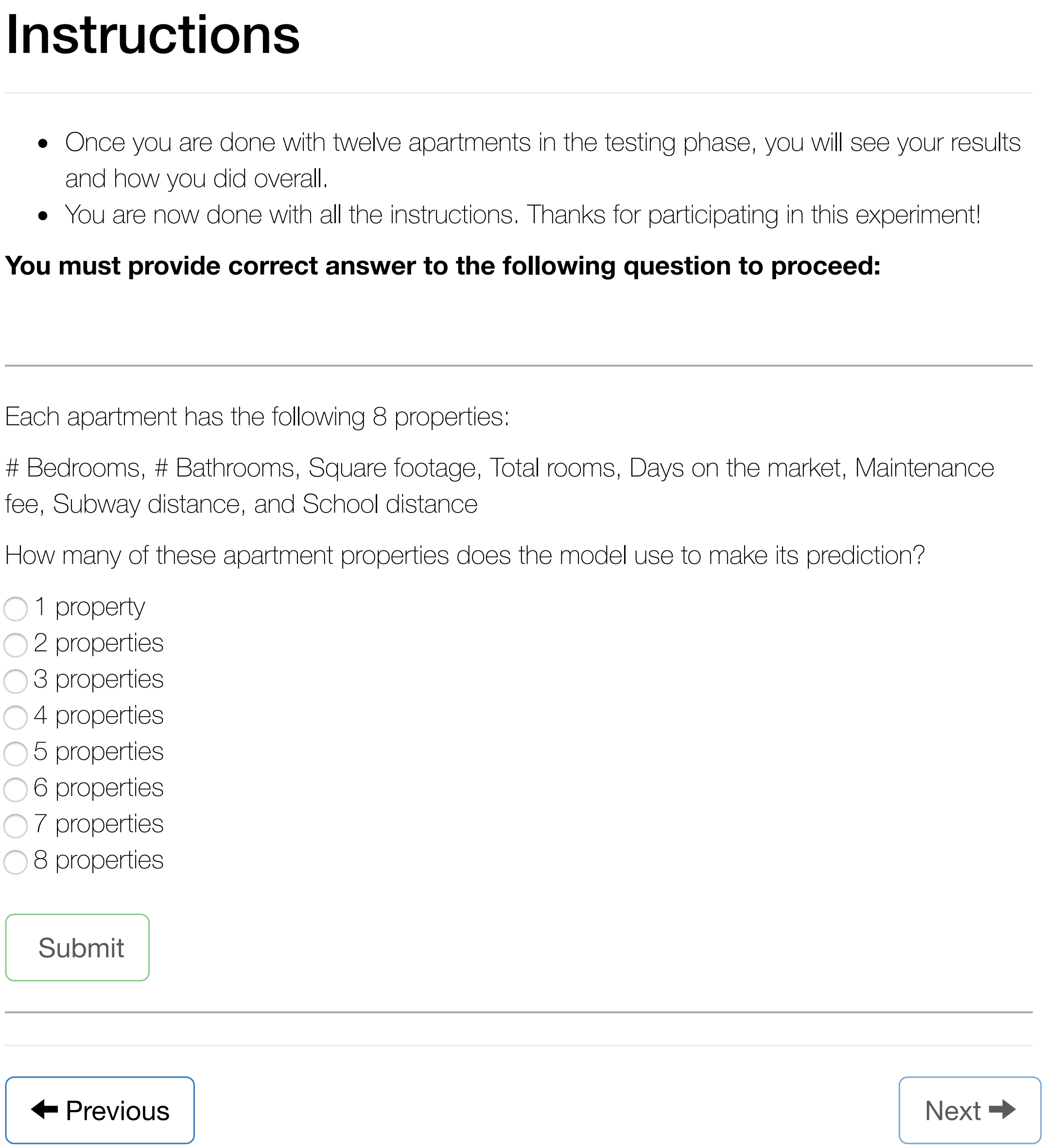
What you thought the model would predict: $800,000

What the model actually predicted: $900,000

What do you think this apartment actually sold for? $0

How confident are you that you got it right?

	1	2	3	4	5	
It's likely I got it wrong	○	○	○	○	○	I'm confident I got it right



Instructions

- Once you are done with twelve apartments in the testing phase, you will see your results and how you did overall.
- You are now done with all the instructions. Thanks for participating in this experiment!

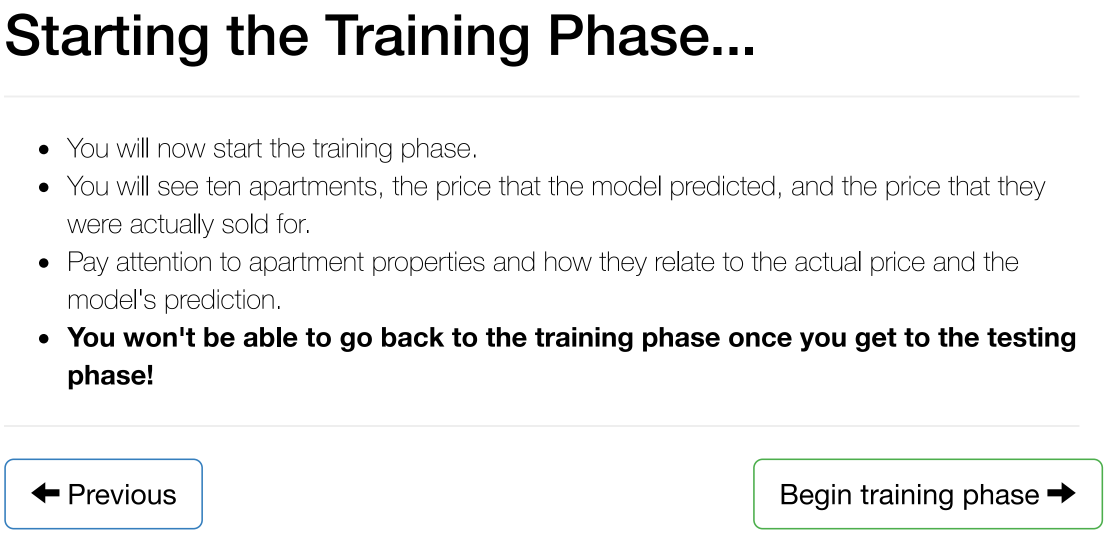
You must provide correct answer to the following question to proceed:

Each apartment has the following 8 properties:

# Bedrooms, # Bathrooms, Square footage, Total rooms, Days on the market, Maintenance fee, Subway distance, and School distance

How many of these apartment properties does the model use to make its prediction?

○ 1 property
○ 2 properties
○ 3 properties
○ 4 properties
○ 5 properties
○ 6 properties
○ 7 properties
○ 8 properties



Starting the Training Phase...

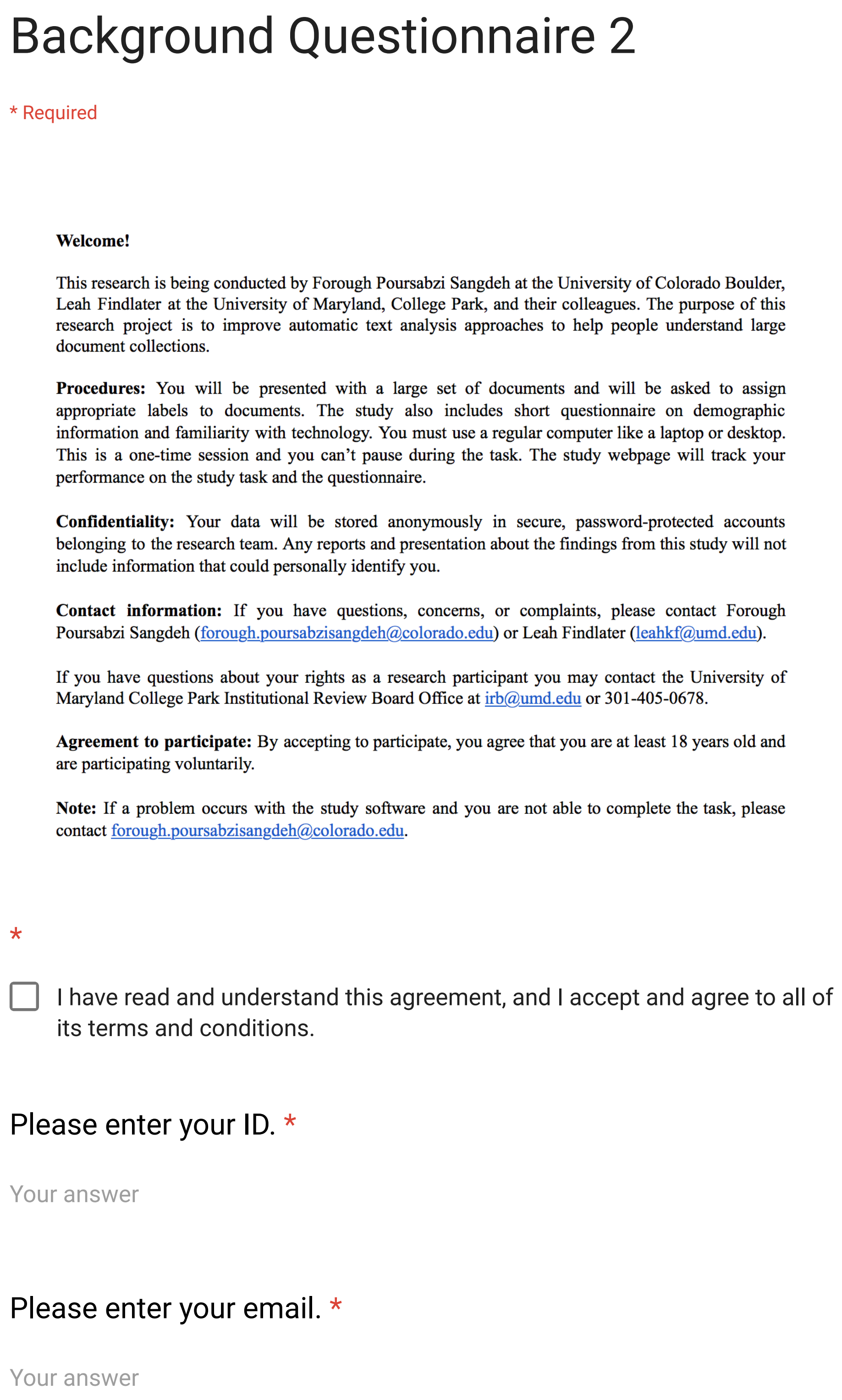
- You will now start the training phase.
- You will see ten apartments, the price that the model predicted, and the price that they were actually sold for.
- Pay attention to apartment properties and how they relate to the actual price and the model's prediction.
- **You won't be able to go back to the training phase once you get to the testing phase!**

Appendix B

ALTO Study (Chapter 4) Material

B.1 Background Questionnaire

Contains the survey that participants filled before starting the task in the TA and TR conditions. For participants in the LA and LR conditions, the question about familiarity with topic model softwares was omitted.



Background Questionnaire 2

* Required

Welcome!

This research is being conducted by Forough Poursabzi Sangdeh at the University of Colorado Boulder, Leah Findlater at the University of Maryland, College Park, and their colleagues. The purpose of this research project is to improve automatic text analysis approaches to help people understand large document collections.

Procedures: You will be presented with a large set of documents and will be asked to assign appropriate labels to documents. The study also includes short questionnaire on demographic information and familiarity with technology. You must use a regular computer like a laptop or desktop. This is a one-time session and you can't pause during the task. The study webpage will track your performance on the study task and the questionnaire.

Confidentiality: Your data will be stored anonymously in secure, password-protected accounts belonging to the research team. Any reports and presentation about the findings from this study will not include information that could personally identify you.

Contact information: If you have questions, concerns, or complaints, please contact Forough Poursabzi Sangdeh (forough.poursabzisangdeh@colorado.edu) or Leah Findlater (leahkf@umd.edu).

If you have questions about your rights as a research participant you may contact the University of Maryland College Park Institutional Review Board Office at irb@umd.edu or 301-405-0678.

Agreement to participate: By accepting to participate, you agree that you are at least 18 years old and are participating voluntarily.

Note: If a problem occurs with the study software and you are not able to complete the task, please contact forough.poursabzisangdeh@colorado.edu.

*

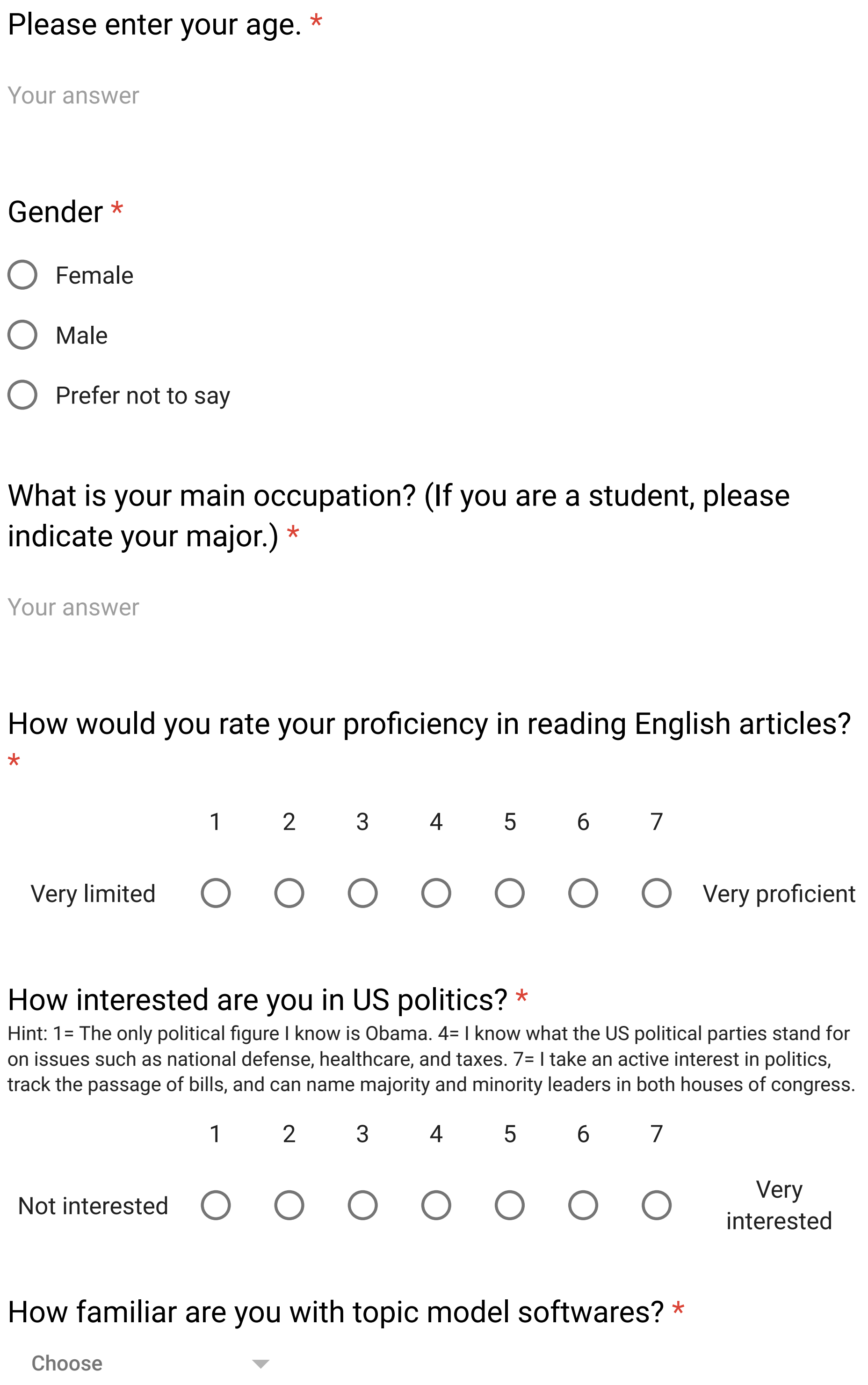
□ I have read and understand this agreement, and I accept and agree to all of its terms and conditions.

Please enter your ID. *

Your answer

Please enter your email. *

Your answer



Please enter your age. *

Your answer

Gender *

○ Female

○ Male

○ Prefer not to say

What is your main occupation? (If you are a student, please indicate your major.) *

Your answer

How would you rate your proficiency in reading English articles? *

	1	2	3	4	5	6	7	
Very limited	○	○	○	○	○	○	○	Very proficient

How interested are you in US politics? *

Hint: 1= The only political figure I know is Obama. 4= I know what the US political parties stand for on issues such as national defense, healthcare, and taxes. 7= I take an active interest in politics, track the passage of bills, and can name majority and minority leaders in both houses of congress.

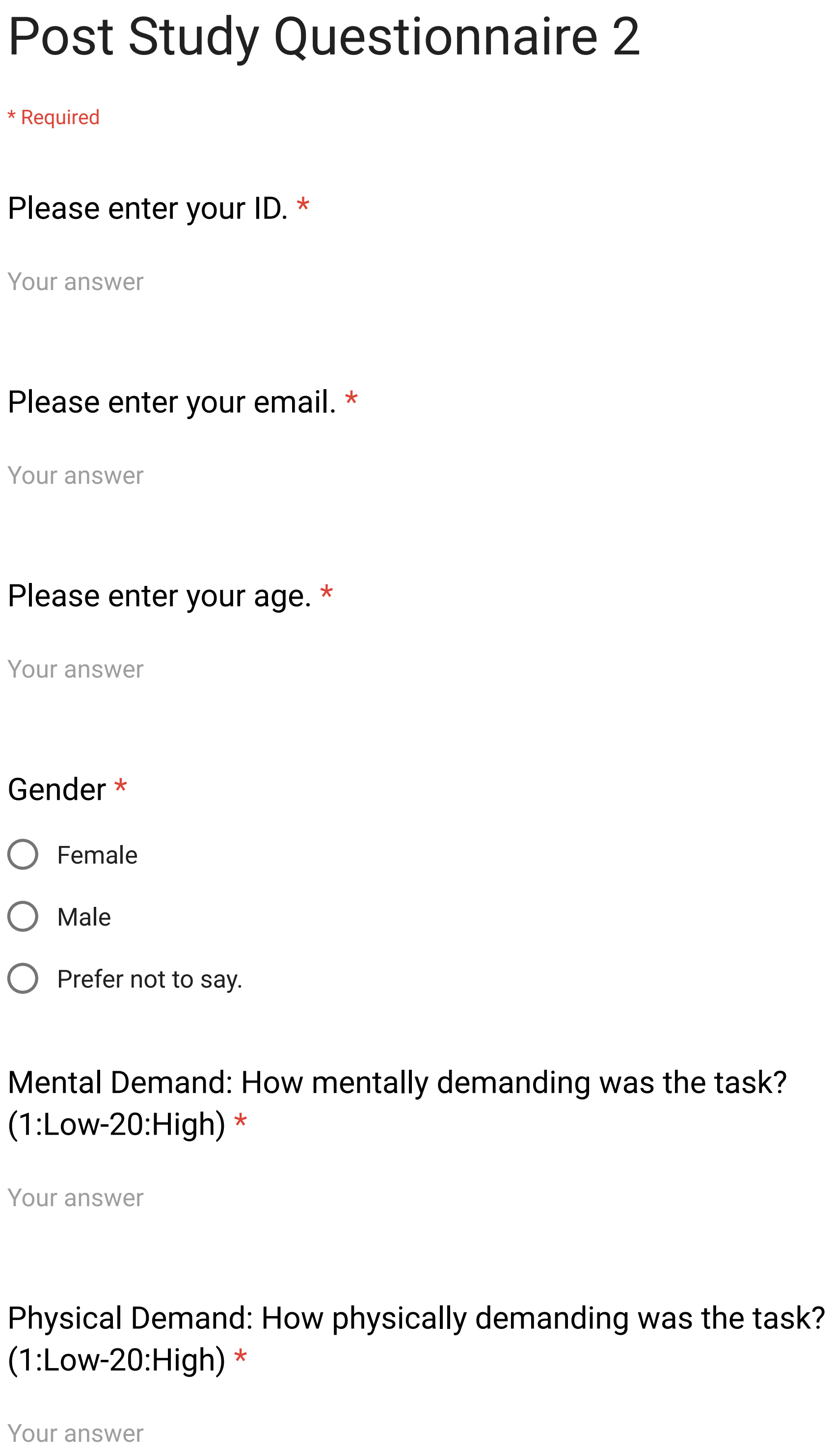
	1	2	3	4	5	6	7	
Not interested	○	○	○	○	○	○	○	Very interested

How familiar are you with topic model softwares? *

Choose ▾

B.2 Post Study Questionnaire

Contains the survey that participants filled after doing the task in the TA and TR conditions. For participants in the LA and LR conditions, the question about topics was omitted.



Post Study Questionnaire 2

* Required

Please enter your ID. *

Your answer

Please enter your email. *

Your answer

Please enter your age. *

Your answer

Gender *

○ Female

○ Male

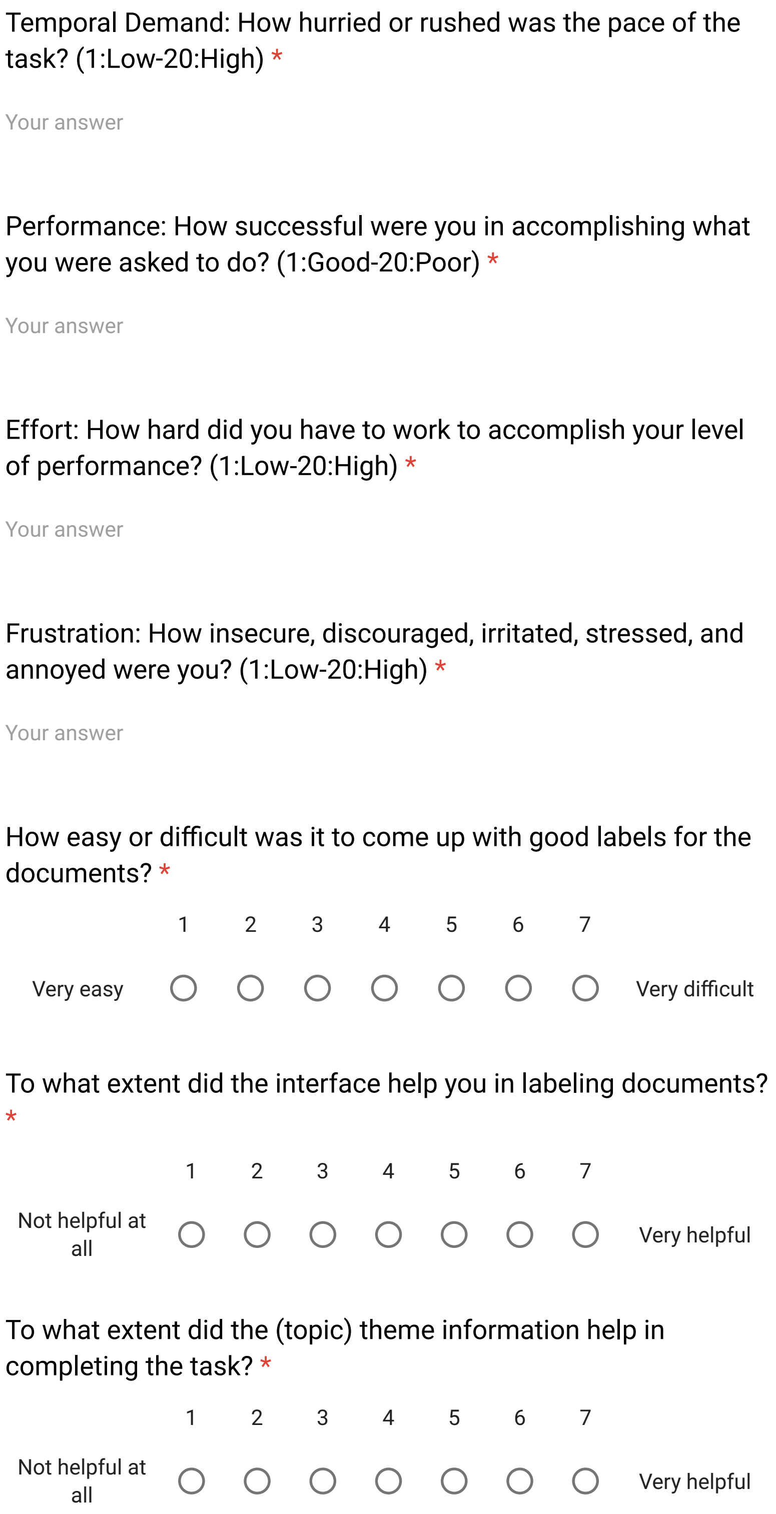
○ Prefer not to say.

Mental Demand: How mentally demanding was the task? (1:Low-20:High) *

Your answer

Physical Demand: How physically demanding was the task? (1:Low-20:High) *

Your answer



Temporal Demand: How hurried or rushed was the pace of the task? (1:Low-20:High) *

Your answer

Performance: How successful were you in accomplishing what you were asked to do? (1:Good-20:Poor) *

Your answer

Effort: How hard did you have to work to accomplish your level of performance? (1:Low-20:High) *

Your answer

Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you? (1:Low-20:High) *

Your answer

How easy or difficult was it to come up with good labels for the documents? *

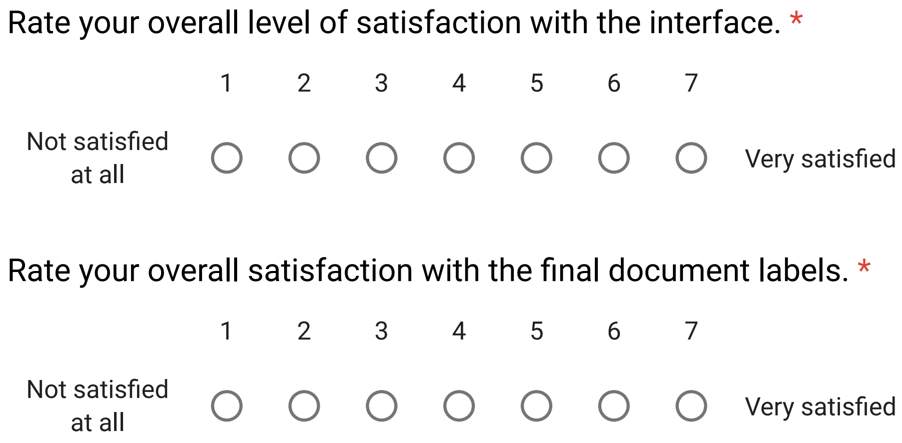
	1	2	3	4	5	6	7	
Very easy	○	○	○	○	○	○	○	Very difficult

To what extent did the interface help you in labeling documents? *

	1	2	3	4	5	6	7	
Not helpful at all	○	○	○	○	○	○	○	Very helpful

To what extent did the (topic) theme information help in completing the task? *

	1	2	3	4	5	6	7	
Not helpful at all	○	○	○	○	○	○	○	Very helpful



Rate your overall level of satisfaction with the interface. *

	1	2	3	4	5	6	7	
Not satisfied at all	○	○	○	○	○	○	○	Very satisfied

Rate your overall satisfaction with the final document labels. *

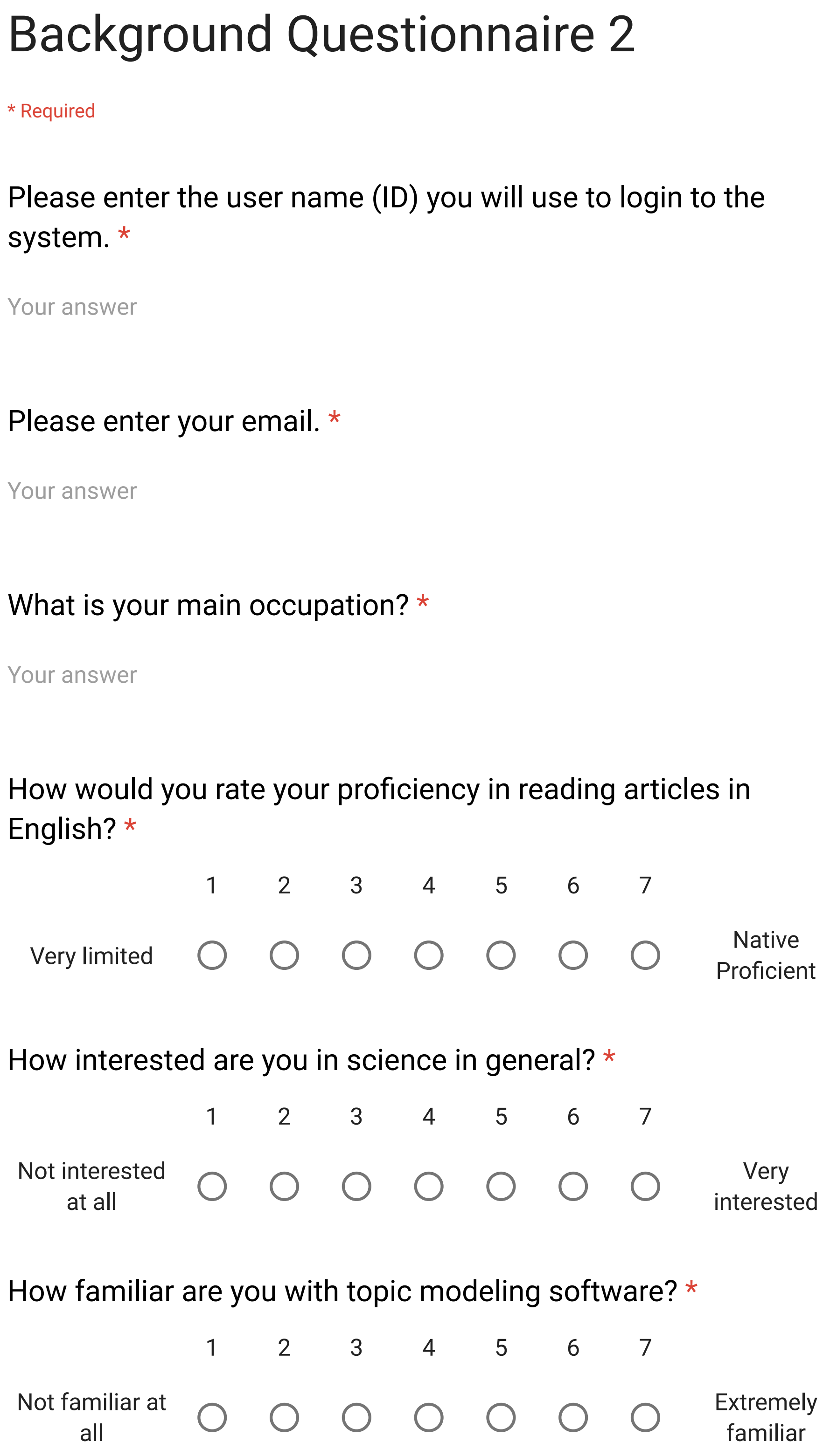
	1	2	3	4	5	6	7	
Not satisfied at all	○	○	○	○	○	○	○	Very satisfied

Appendix C

Science Policy Study (Chapter 5) Material

C.1 Background Questionnaire

Contains the survey that participants filled before starting the task in the TOPIC condition. For participants in the LIST condition, the question about familiarity with topic model software was omitted.



Background Questionnaire 2

* Required

Please enter the user name (ID) you will use to login to the system. *

Your answer

Please enter your email. *

Your answer

What is your main occupation? *

Your answer

How would you rate your proficiency in reading articles in English? *

	1	2	3	4	5	6	7	
Very limited	○	○	○	○	○	○	○	Native Proficient

How interested are you in science in general? *

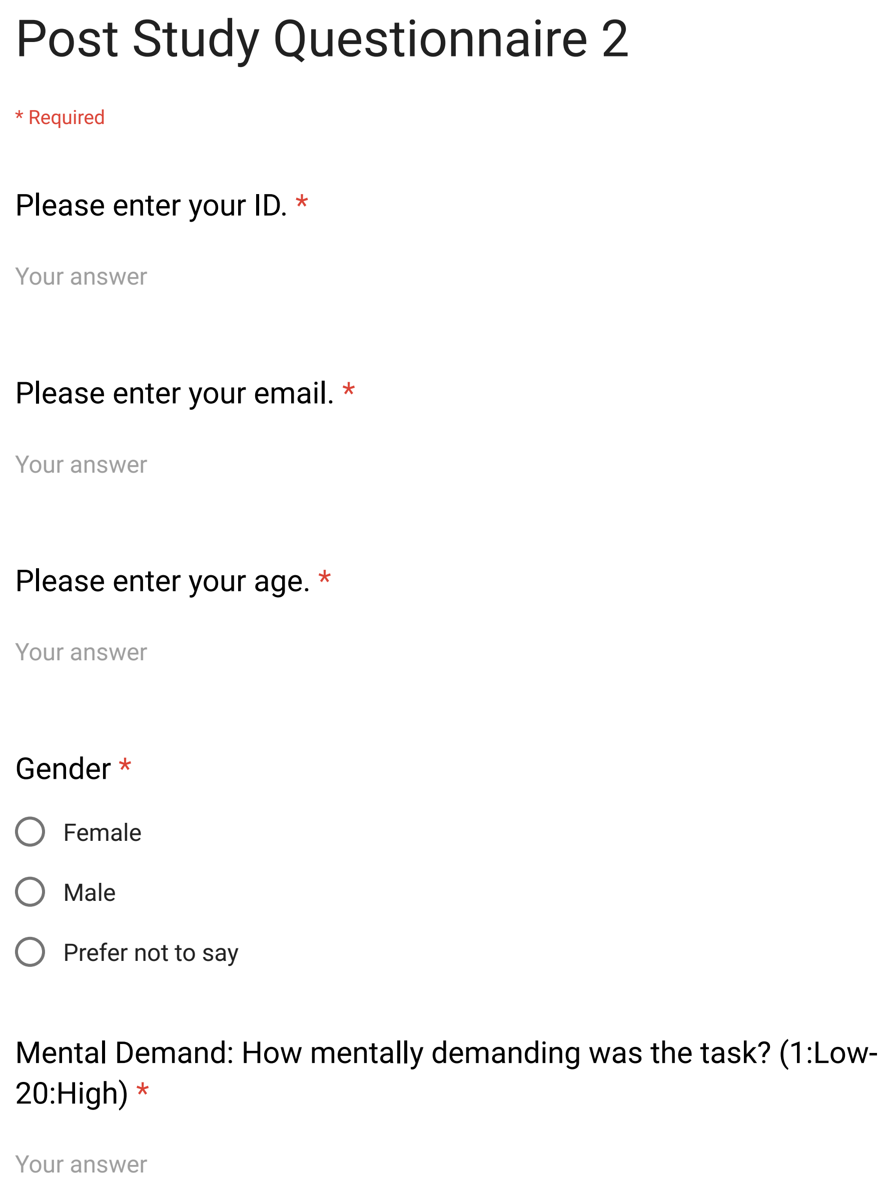
	1	2	3	4	5	6	7	
Not interested at all	○	○	○	○	○	○	○	Very interested

How familiar are you with topic modeling software? *

	1	2	3	4	5	6	7	
Not familiar at all	○	○	○	○	○	○	○	Extremely familiar

C.2 Post Study Questionnaire

Contains the survey that participants filled after doing the task in the TOPIC condition. For participants in the LIST condition, the question about topics was omitted.



Post Study Questionnaire 2

* Required

Please enter your ID. *

Your answer

Please enter your email. *

Your answer

Please enter your age. *

Your answer

Gender *

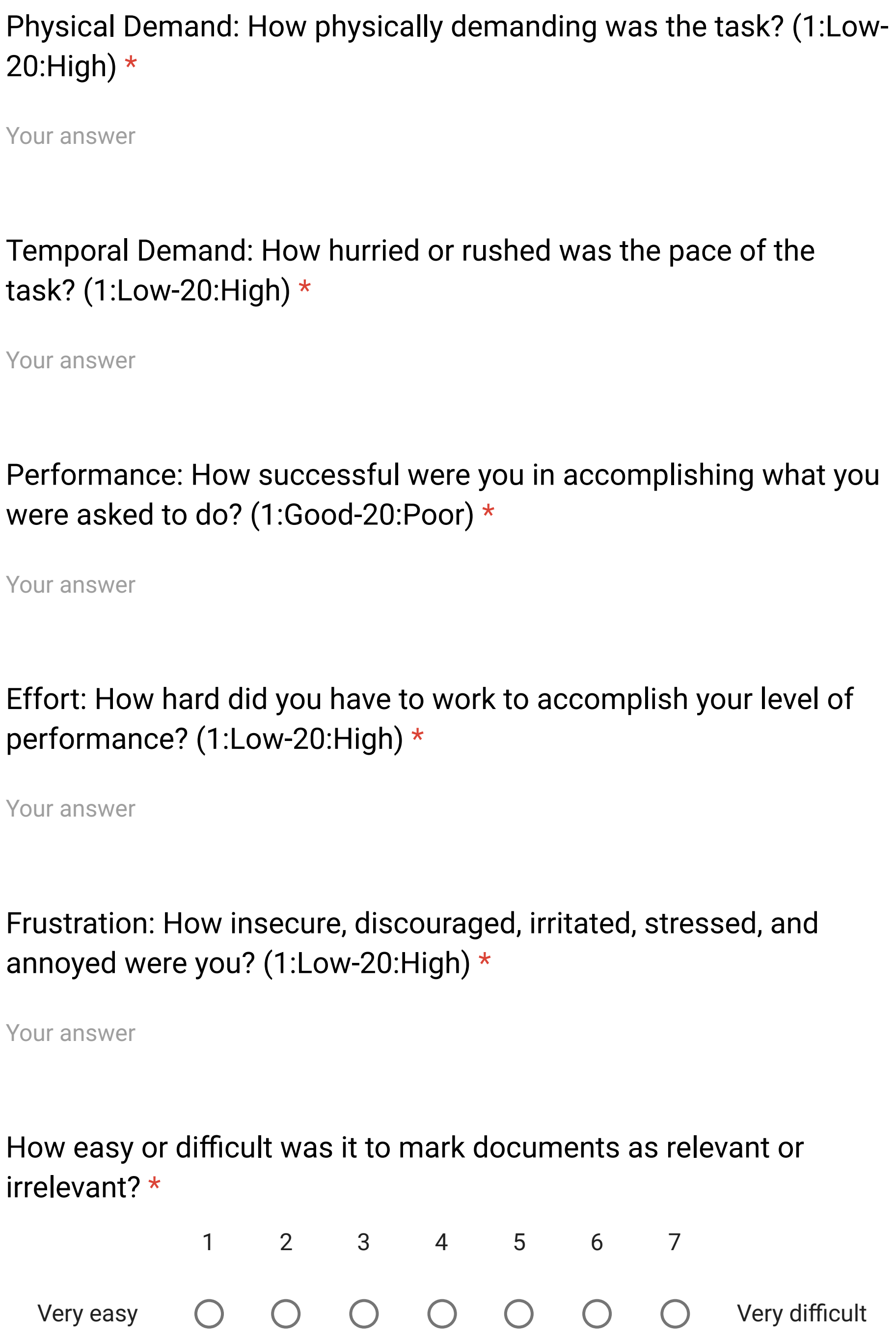
○ Female

○ Male

○ Prefer not to say

Mental Demand: How mentally demanding was the task? (1:Low-20:High) *

Your answer



Physical Demand: How physically demanding was the task? (1:Low-20:High) *

Your answer

Temporal Demand: How hurried or rushed was the pace of the task? (1:Low-20:High) *

Your answer

Performance: How successful were you in accomplishing what you were asked to do? (1:Good-20:Poor) *

Your answer

Effort: How hard did you have to work to accomplish your level of performance? (1:Low-20:High) *

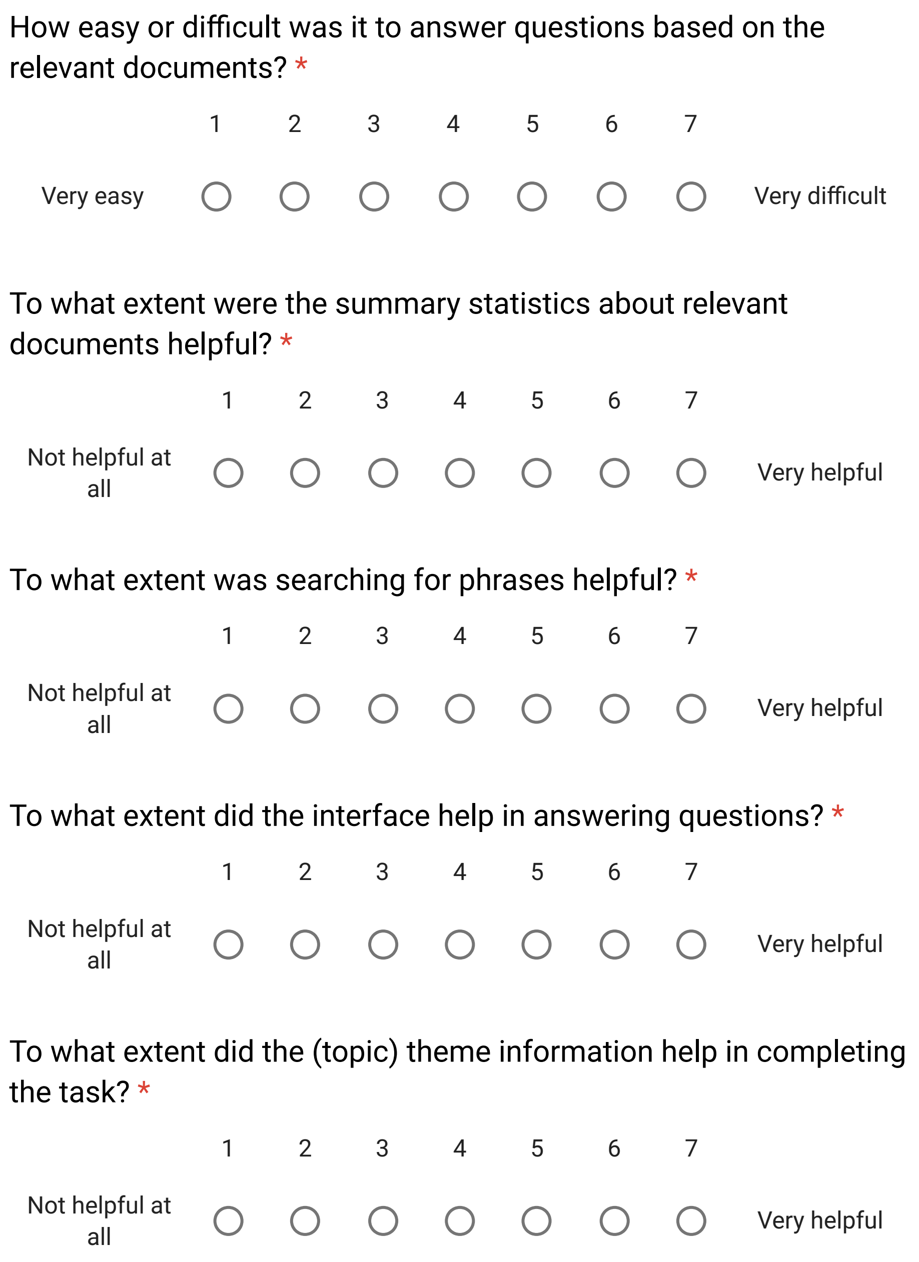
Your answer

Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you? (1:Low-20:High) *

Your answer

How easy or difficult was it to mark documents as relevant or irrelevant? *

	1	2	3	4	5	6	7	
Very easy	○	○	○	○	○	○	○	Very difficult



How easy or difficult was it to answer questions based on the relevant documents? *

	1	2	3	4	5	6	7	
Very easy	○	○	○	○	○	○	○	Very difficult

To what extent were the summary statistics about relevant documents helpful? *

	1	2	3	4	5	6	7	
Not helpful at all	○	○	○	○	○	○	○	Very helpful

To what extent was searching for phrases helpful? *

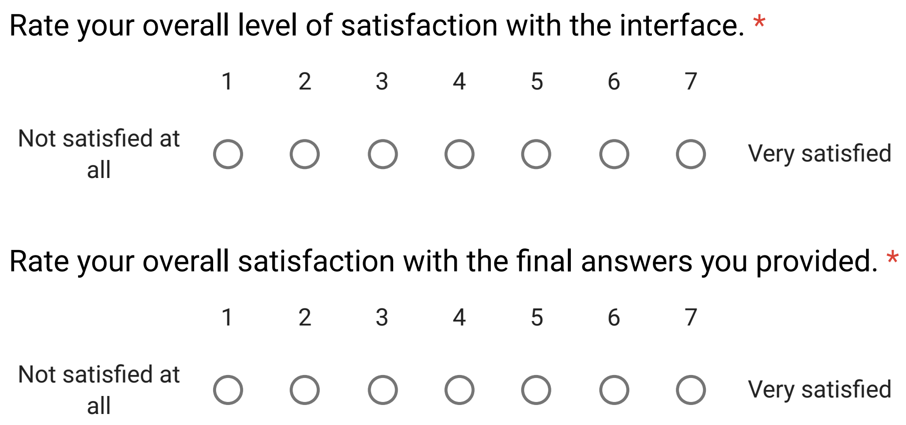
	1	2	3	4	5	6	7	
Not helpful at all	○	○	○	○	○	○	○	Very helpful

To what extent did the interface help in answering questions? *

	1	2	3	4	5	6	7	
Not helpful at all	○	○	○	○	○	○	○	Very helpful

To what extent did the (topic) theme information help in completing the task? *

	1	2	3	4	5	6	7	
Not helpful at all	○	○	○	○	○	○	○	Very helpful



Rate your overall level of satisfaction with the interface. *

	1	2	3	4	5	6	7	
Not satisfied at all	○	○	○	○	○	○	○	Very satisfied

Rate your overall satisfaction with the final answers you provided. *

	1	2	3	4	5	6	7	
Not satisfied at all	○	○	○	○	○	○	○	Very satisfied