SenseMate: An Accessible and Beginner-Friendly Human-AI Platform for Qualitative Data Analysis

SenseMate: An Accessible and Beginner-Friendly Human-AI Platform for Qualitative Data Analysis

Cassandra Overney
Massachusetts Institute of Technology
Cambridge, Massachusetts, United States
coverney@mit.edu

Belén Saldías
Massachusetts Institute of Technology
Cambridge, Massachusetts, United States

Dimitra Dimitrakopoulou
Massachusetts Institute of Technology
Cambridge, Massachusetts, United States

Deb Roy
Massachusetts Institute of Technology
Cambridge, Massachusetts, United States

ABSTRACT

Community organizations face challenges in harnessing the power of qualitative data analysis, or sensemaking, to understand the diverse perspectives and needs brought up by their constituents. One of the most time-consuming and tedious parts of sensemaking is qualitative coding, or the process of identifying themes across a large and unstructured corpus of community input. A challenge in qualitative coding is attaining high intercoder reliability, especially between expert and beginner sensemakers. In this work, we present SenseMate, a novel human-AI system designed to help with qualitative coding. SenseMate leverages rationale extraction models, a new machine learning strategy to semi-automate sensemaking, which produces theme recommendations and human-interpretable explanations. The models were trained on a dataset of people's experiences living in Boston, which was annotated for themes by expert sensemakers. We integrated rationale extraction models into SenseMate through an iterative, human-centered design process revolving around four key design principles derived from an extensive literature review. The design process consisted of three iterations with continuous feedback from seven people associated with community organizations. Through an online experiment involving 180 novice sensemakers, we aimed to determine whether AI-generated recommendations and rationales would decrease coding time, increase intercoder reliability (i.e. Cohen's kappa), and minimize differences between novice and expert coding decisions (i.e. F-score of participant answers compared to expert gold labels). We found that though the model recommendations and explanations increased coding time by 49 seconds per unit of analysis, they raised intercoder reliability by 29% and coding F-score by 10%. Regarding the effectiveness of SenseMate's design, participants reported that the platform was generally easy to use. In summary, Sensemate is (1) built for beginner sensemakers without a technical background, a user group that prior work doesn't focus on, (2) implements rationale extraction models to recommend themes and generate explanations, which has advantages over large language models in terms of user privacy and control, and (3) contains original and intuitive features created from user feedback that can be applied to future QDA systems.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **User interface design**; • **Computing methodologies** → **Natural language processing**.

KEYWORDS

Human-AI Collaboration, Explainable AI Methods, User Experiments and Studies, Content Analysis, Qualitative Coding

ACM Reference Format:
Cassandra Overney, Belén Saldías, Dimitra Dimitrakopoulou, and Deb Roy. 2024. SenseMate: An Accessible and Beginner-Friendly Human-AI Platform for Qualitative Data Analysis. In *29th International Conference on Intelligent User Interfaces (IUI '24), March 18–21, 2024, Greenville, SC, USA.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3640543.3645194

1 INTRODUCTION

Gathering qualitative feedback from constituents is essential for local entities (e.g. non-profits, municipalities, and businesses) to better understand the experiences, perspectives, and needs of the communities they serve. However, even if community-centered organizations want to engage in conversations with their constituents, they often don't have the skills or resources to analyze the rich data they gather through methods like surveys, interviews, and facilitated dialogue. As a result, organizations either outsource the analysis to individuals who are not immersed in the community, complete a superficial examination of the data, or let the data languish in obscurity and never uplift the perspectives that people share. To address this problem, we need accessible qualitative data analysis (QDA) tools that are beginner-friendly. Unfortunately, existing QDA software (e.g. NVivo[1] and ATLAS.ti[2]) has steep learning curves [49, 61], and open-source AI-assisted annotation tools (e.g. Prodigy[3]) cater to data scientists, potentially excluding beginner sensemakers without a technical background. Additionally, QDA platforms introduced by prior work [29, 57] tend to be designed for and evaluated by expert qualitative researchers.

IUI '24, March 18–21, 2024, Greenville, SC, USA

ACM ISBN 979-8-4007-0508-3/24/03
https://doi.org/10.1145/3640543.3645194

[1] https://lumivero.com/products/nvivo/.
[2] https://atlasti.com/.
[3] https://prodi.gy/.

In contrast to these other systems, our platform supports individuals working in community organizations seeking to understand constituent needs for decision-making but lacking the expertise to efficiently and reliably analyze qualitative data. We focus specifically on sensemaking, a QDA method similar to thematic analysis [9, 38, 58]. The goal of sensemaking is to systematically extract patterns, or themes, in large-scale nuanced data. After segmenting the data into units of analysis, sensemakers construct a codebook, which serves as a comprehensive guide of patterns within the data. Codebooks typically consist of a hierarchical list of codes, or themes, which need to be applied to the entire dataset through a process called qualitative coding. With large datasets, qualitative coding is typically carried out manually within groups and can feel tedious and time-consuming [7, 35, 51]. In addition, a group of sensemakers may struggle to attain high intercoder reliability, which adds to the overall analysis time. The challenge of achieving high consistency is enhanced when comparing novice and expert coding decisions. Prior work found that novice sensemakers struggle to produce coding decisions similar to those made by experts [13, 32].

Finding a balance between manual and fully automated coding can help increase efficiency and consistency while allowing human judgment and preventing systematic machine errors. Several studies have interviewed researchers to identify opportunities for human-AI collaboration in qualitative coding and found that AI needs to be modifiable and transparent about its recommendations for successful collaboration [39, 51]. Previous work that created transparent algorithms for qualitative coding mainly applied manually generated keyword-based rules, which require knowledge of pattern rules [29, 51, 57]. To increase the accessibility of semi-automated qualitative coding for non-technical users, we apply a machine learning strategy, known as rationale extraction models. Rationale extraction models are helpful in this context by generating concise, contextualized, and easily interpretable justifications for coding decisions. The explicit extraction of reasoning behind a suggestion enables researchers to concentrate on specific aspects of the data, fostering a more nuanced and accurate coding process. Notably, these models are fine-tuned on domain-specific data, enhancing their relevance for qualitative coding in specialized areas. Though rationale extraction models have been applied to other data annotation tasks [42, 45], SenseMate is the first platform that utilizes these models for qualitative coding. To facilitate bidirectional human-AI communication, we created SenseMate through a human-centered design process with input from people associated with community organizations. The platform acknowledges the ambiguous and subjective nature of qualitative coding [15] by providing affordances for users to revise and improve model behaviors.

This paper describes the process of developing SenseMate. First, we trained rationale extraction models to recommend themes and generate human-interpretable rationales for each unit of analysis in qualitative data. The models were trained using a supervised extract-to-predict pipeline on an annotated dataset containing 69 facilitated small-group conversations about people's experiences living in Boston. Annotations, or gold labels, were created by a group of highly experienced sensemakers with a plethora of background knowledge about the Boston community. We focused on classifying nine themes from the dataset, such as "Community Values" and "Housing Affordability". Next, we implemented an iterative, human-centered design process to create SenseMate. Based on our literature review and personal sensemaking experiences, we identified four design principles to guide the creation of SenseMate: 1) providing AI suggestions on demand, 2) generating model explanations that are easy to judge and non-repetitive, 3) creating user-driven and intuitive processes to collect high-quality feedback on AI suggestions, and 4) reducing model overreliance through simple design interventions. Throughout the design process, we gathered feedback through wizard-of-oz user testing from seven people connected with our target user group. Only one participant received formal training in qualitative data analysis. Six people were either actively working in community-based organizations or had helped community-based organizations analyze their data, and one person had never done sensemaking before.

After thirteen user testing sessions and three design iterations, we implemented a prototype of SenseMate and conducted a comprehensive user evaluation through an online experiment. A majority of participants had little to no sensemaking experience and were randomly assigned to one of three experiment conditions: no AI assistance, only theme recommendations, and both theme recommendations and rationales. From the user study, we aimed to determine whether AI-generated recommendations and rationales would decrease coding time (RQ-1), increase intercoder reliability (RQ-2), and minimize differences between novice and expert coding decisions (RQ-3). We refer to the gap between novice and expert coding decisions as coding performance, which is measured by calculating the accuracy, precision, recall, and F-score of participant answers compared to expert gold labels. Besides examining the impact of AI assistance on qualitative coding, we wanted to evaluate SenseMate's usability and the effectiveness of various design decisions.

The user study revealed that participants with access to AI assistance in the form of theme recommendations and rationales had higher coding times compared to participants without access to AI (RQ-1). Though the models did not make qualitative coding less time-consuming, participants who spent longer on the platform still thought their experiences were productive. Furthermore, compared to participants without AI support, those who received theme recommendations and rationales had higher intercoder reliability (RQ-2) and coding performance in terms of accuracy, precision, recall, and F-score (RQ-3). The rationale extraction models helped novice sensemakers become more aligned with each other and with experts. However, an increase in coding alignment may occur only in situations where models have high performance with respect to expert labels.

Regarding the effectiveness of SenseMate's design, participants reported that the platform was generally easy to use. One of our design principles was to generate model explanations that are easy to judge and non-repetitive. We found that the rationales were helpful and efficient to evaluate since participants chose to view, on average, a third of the model explanations and only spent a few seconds studying each one. Another design principle involved creating efficient ways of collecting high-quality feedback on AI suggestions. On average, participants with access to model recommendations and rationales only spent around four seconds correcting a rationale, while achieving more than 90% agreement. We reflect on

these results, along with qualitative feedback from our wizard-of-oz sessions and user study, to generate several design implications when creating AI-based sensemaking platforms (e.g. considering first impressions on AI assistance, developing varied and efficient feedback mechanisms, and letting users choose when to receive AI assistance). The primary contributions of our work are:

(1) **Implementing SenseMate, a novel human-AI system for qualitative coding.** SenseMate is an accessible qualitative coding platform for beginner sensemakers without a technical background, a user group that prior work doesn't focus on. Wizard-of-oz (WOz) sessions with non-researchers from community organizations ensured that all features were created based on user feedback.
(2) **Applying a new natural language processing method (i.e. rationale extraction models) for qualitative coding.** Few methods of semi-automated qualitative coding have also produced human-interpretable explanations. SenseMate attempts to address this gap with rationale extraction models, which have advantages over large language models in terms of user privacy and control.
(3) **Evaluating SenseMate through a user study with 180 participants.** After running an online experiment, we document insights into the effectiveness of our tool. We synthesize these insights into lessons and design implications for future human-AI sensemaking systems.

2 RELATED WORK

Previous work in the automated qualitative data analysis (QDA) space has explored the idea that AI can make sensemaking less time-consuming [1, 31, 53, 61] and expensive [1], while increasing its scalability [1, 2, 31, 53]. Though automation can address some of the main challenges around sensemaking, full automation of the process should be approached with caution due to possible biases from machine-generated analyses [25]. Semi-automation can mitigate the challenges of full automation while retaining the benefits of using AI-based methods. One of the most helpful areas for semi-automation is qualitative coding because it is a tedious and time-consuming part of the sensemaking process [7, 35, 51]. When creating new approaches to semi-automate qualitative coding, prior work emphasizes the importance of methods that are transparent and modifiable [15, 51], while honoring reflection and serendipity [16, 39]. Research in human-AI collaboration can shed light on ways to satisfy some of these requirements. Specifically, our work builds on the following main areas of prior work: 1) algorithm-in-the-loop decision-making, 2) explainable AI, and 3) systems that semi-automate sensemaking.

2.1 Algorithm-in-the-Loop Decision-Making

One form of human-AI collaboration that is especially applicable to sensemaking is algorithm-in-the-loop decision-making, also known as AI-assisted decision-making. In algorithm-in-the-loop systems, AI performs an assistive role by providing recommendations, while humans are the final decision makers [33]. SenseMate supports algorithm-in-the-loop decision-making because qualitative coding is a high-stakes task, in which coding decisions determine how the data is interpreted later on in the analysis pipeline. Creators of algorithm-in-the-loop systems need to consider how automation impacts human agency [64]. Lai & Tan proposed a spectrum between full human agency and full automation with varying levels of machine assistance along the spectrum (e.g. showing machine-predicted labels with or without explanations) [43]. Green & Chen introduced three principles for algorithm-in-the-loop decision-making: accuracy, reliability, and fairness. The authors tested six different model interactions and found that while almost every treatment improved the accuracy of predictions, no treatment satisfied the criteria for reliability and fairness [33]. Consequently, there is growing interest in integrating machine learning and user interface design to improve reliability and fairness in algorithm-in-the-loop systems.

Several types of human-AI systems that integrate machine learning and design include interactive machine learning (IML) [26, 30, 59, 60], machine teaching [30], and mixed-initiative systems [14, 27, 36]. Interactive machine learning makes AI more accessible to non-experts by framing the model training process as an HCI task. A typical IML system has four components: 1) user, 2) model, 3) data, and 4) interface [26]. The machine teaching paradigm is similar to IML in that humans are tasked with helping AI models improve over time. The goal of machine teaching is to make the process of developing models as intuitive as teaching students. As a result, the emphasis in machine teaching systems is to support the teacher by helping them understand the reasoning behind a model's decisions, especially for mistakes. A machine teaching paradigm can make algorithms more transparent and interpretable during qualitative coding [15]. A slightly different human-AI system involves the mixed-initiative approach, in which machines and humans collaborate efficiently to achieve the user's goals [36]. Computationally appropriate tasks are offloaded to the machines, which enables humans to complete the other, typically more abstract, tasks. Mixed-initiative systems differ from IML and machine teaching since they don't prioritize asking users to teach the models but let users focus on higher-level analytical reasoning, which can be helpful for sensemaking. The next section describes the role of machine-generated explanations to increase the interpretability of AI suggestions, which is a key factor in promoting transparency in algorithm-in-the-loop systems.

2.2 Explainable AI

Model explanations provide a form of communication between humans and AI models in algorithm-in-the-loop systems. Explanations can be white-box or black-box (i.e. showing the internal workings of an algorithm or not), as well as static or interactive [17]. Local explanations summarize the model's rationale for a particular example, while global explanations provide a high-level understanding of how the model works [42]. Several characteristics that model explanations should strive for include interpretability, trust calibration, a low cognitive effort for users, improved understanding of the model, and help in recognizing model uncertainty [11, 30, 55, 63, 67]. Trust calibration involves providing users with the right amount of trust, such that they don't over-rely on model recommendations when they are wrong or ignore the recommendations when they are correct [11, 63, 67]. Ribeiro et al. define trust in two ways: 1) trusting a prediction, and 2) trusting the model [55]. Understanding

the reasons behind a model's predictions can help users achieve both types of trust.

Prior studies found that explanations can improve user experience [42], the understanding of AI systems [17], and human perception of AI's usefulness [48]. There are mixed results in terms of using explanations to enhance efficiency [42, 52], trust [17, 19, 30, 43, 67], and accuracy [11, 19, 43, 48, 52]. In addition, some studies found that explanations can increase model overreliance [6, 30], while others discovered the opposite effect [62]. One way to prevent overreliance is through cognitive forcing functions (e.g. adding a time delay or having someone click on a button to see the AI's recommendation). Buçinca et al. found that cognitive forcing significantly reduced overreliance but resulted in lower ratings for usability and trust [11]. Vasconcelos et al. applied a cost-benefit framework and found that overreliance decreases when 1) the task becomes more difficult compared to the explanation, 2) the explanation is easier to understand, or 3) participants receive a higher monetary reward for accuracy [62]. Vasconcelos et al.'s research provides insight into the nuanced relationship between explanations and model overreliance. As Bansal et al. and Ghai et al. observed, explanations that are difficult to verify can lead to increased overreliance. On the other hand, when explanations are easy to understand, especially in comparison to the original task difficulty, users are less likely to blindly rely on the recommendations.

Previous work on automated sensemaking emphasizes the importance of including explanations to increase trust and transparency while fostering reflection [37, 39, 56, 57]. To address these requirements, we generate explanations from rationale extraction models. Rationale extraction models apply unsupervised machine learning to identify words within a text input that is connected with a particular label [45]. Explanations from these models are easy to generate and are human-interpretable, which makes it promising to use in text classification tasks like qualitative coding.

2.3 Systems that Semi-Automate Sensemaking

Various algorithms have been applied to automate aspects of the sensemaking process, including clustering [31, 40, 46], topic modeling [1, 13], simple machine learning models [24, 32, 47, 57], and large language models [32, 46]. Nonetheless, very few methods of semi-automated qualitative coding have also produced model explanations. The main exception includes studies that applied rule-based qualitative coding, in which the presence of particular keywords determines whether a piece of data is associated with a code [21, 51, 57]. However, keyword-based rules are typically specified by users and are not machine-generated. SenseMate attempts to address the lack of machine learning algorithms that can generate human-interpretable explanations during qualitative coding. Specifically, SenseMate is the first qualitative coding platform that employs rationale extraction models.

Several studies have designed systems to semi-automate sensemaking. These systems include not only machine learning algorithms but also carefully designed features to support human-AI collaboration. QuAD is a platform that helps research teams cluster qualitative data into themes using BERT embeddings and the Girvan-Newman algorithm [31]. Machine-generated grouping suggestions can be accepted, edited, or declined, and users can also pin locations for easy and quick navigation [31]. SenseMate adapts QuAD's ability to display suggestions that users can easily approve or reject. Drouhard et al. took a unique approach to support collaborative qualitative coding by designing Aeonium, a visual analytics interface that trains a supervised machine learning model to identify disagreement between coders. The goal is to focus user attention on ambiguous parts of the data and codebook. From an experimental study, the authors discovered that Aeonium increased user understanding of the themes and helped people reflect on their coding decisions [24]. A few features within Aeonium inspired SenseMate's initial designs, including the ability to flag ambiguous data and highlight keywords to explain coding decisions. Of all the platforms we investigated, SenseMate is most similar to Cody, a system that semi-automates qualitative coding with supervised machine learning [57]. In Cody, users define code rules that AI extends to unseen data. AI suggestions are supported by explanations that highlight relevant keywords from the code rules. Through an evaluation of Cody, Rietz & Maedche found that the AI suggestions improved coding quality and not speed, and the explanations were commonly desired but rarely used [57]. Our work builds upon Cody by exploring how a new method of generating theme suggestions and explanations would impact qualitative coding. SenseMate is the first qualitative coding platform that facilitates collaboration between human sensemakers and rationale extraction models. Besides applying a novel machine learning method, SenseMate stands out from other systems described in this section by supporting beginner sensemakers without a technical background, as opposed to aiding experienced researchers.

3 THE SENSEMATE SYSTEM

SenseMate has been carefully built, with equal attention given to the modeling and design elements. The modeling side focuses on how we use rationale extraction models to recommend themes for qualitative data. The design side is primarily concerned with how users interact with the models.

3.1 Rationale Extraction Models

Rationale extraction models produce two outputs from a piece of text: recommendations for possible codes, or themes, (Figure 2) and corresponding rationales as to which words relate to the recommended themes (Figure 3). The following sections explain how we trained and evaluated SenseMate's rationale extraction models.

3.1.1 Method. To design and evaluate SenseMate, we worked with an annotated dataset containing 69 facilitated small-group conversations, in English, about people's experiences living in Boston, a United States city with over 650,000 people. These 69 conversations account for 175,899 words and 68 hours of transcribed audio. The community conversations were recorded and transcribed. From there, five people conducted sensemaking on the data. (The team consisted of researchers trained in QDA methods and community leaders with extensive knowledge of the local context around the data.) The unit of analysis was responses to conversation prompts (e.g. what is your question about the future of Boston and your place in it?). The responses were extracted from conversation transcripts to create 1,151 **snippets** that went into the sensemaking process. After four codebook iterations, the sensemaking team reached a

consensus on nine parent themes and forty sub-themes. Each snippet was labeled with one or more themes, which form the expert gold labels. To reduce the complexity of the codebook for participants in our user study, we decided to sample a diverse subset of nine themes: "Community Values", "Covid-19", "Housing Affordability", "Income", "Processes", "Quality of Education", "Race-Based Inequality", "Sense of Safety", and "Transportation". Table 3 in the Appendix contains more information about each theme.

Rationale extraction models have two components: an encoder that classifies whether a conversation snippet contains a code or not and a generator that tries to identify the subset of words within a snippet that relates to the code (i.e. the rationale) [45]. We train binary rationale extraction models for each theme, resulting in nine separate models. The input into the model is a conversation snippet, which goes through an embedding layer to convert the text into numeric data. BERT (Bidirectional Encoder Representations from Transformers) word embeddings are used to capture contextual information [22]. From there, the data goes through the generator, or rationale extraction layer, which produces a rationale, or a subset of words in the input. The generator applies a soft attention mechanism to improve rationale quality [4]. Afterward, the rationale is passed through the encoder, or classification layer, and outputs a binary prediction of whether the input contains one of the themes. Various model architectures can be used to create the encoder and generator. We apply the recurrent neural network (RNN) structures that were initially proposed by Lei et al. [45]. The encoder and generator are learned jointly based on the human-labeled classification for each training example. For each of the nine themes, the dataset is split into a training (80%), validation (10%), and test (10%) set. When training the models, we used 20 epochs, a batch size of 8, the Adam Optimizer with a learning rate of 2×10^{-5}, and the cross-entropy loss function. To evaluate the rationales produced by the models, we manually created rationales for all the validation and test examples. Each rationale extraction model produces a classification score (i.e. the probability that a conversation snippet contains a particular theme) and a rationale prediction array. The rationale prediction array contains a probability for each token, or word, in a snippet. The higher the probability, the more attention the model gave to a token when generating the final classification. The tokens with the highest probabilities would form the model's rationale.

3.1.2 Evaluation. To evaluate the models, we calculated performance metrics to determine which themes have the highest and lowest performances. The encoder is evaluated through metrics, such as accuracy, precision, recall, and F-score (Table 1). The generator is assessed by calculating the intersection between machine and human-generated rationales for all positive examples in the validation and test sets (Table 2). Table 1 displays the classification performance for each of the nine themes. The two most ambiguous themes, "Processes" and "Community Values" consistently have some of the lowest values in all metrics. "Processes" has noticeably worse performance compared to the other themes. More precisely defined themes, such as "Housing Affordability" and "Race-Based Inequality", have better performance. The rationale extraction performance is summarized in Table 2. Similar to classification performance, the two most ambiguous themes, "Processes" and "Community Values", have the lowest average rationale extraction performance in terms of accuracy, recall (how much of the human-generated rationale was detected by the model), and F-score. Precision tends to be slightly lower than recall, suggesting a high false positive rate. In general, it appears that more concrete themes (e.g. "Housing Affordability") have better-performing rationale extraction models compared to more ambiguous themes (e.g. "Processes" and "Community Values").

3.2 User Interface

We applied an iterative human-centered design process when creating SenseMate.

3.2.1 Design Process. Our design process involved three design iterations and two rounds of user feedback. We started the design process by conducting several rounds of concept sketching, or simplified sketches, before creating slightly higher fidelity mockups. To evaluate the mockups, we gathered feedback through wizard-of-oz (WOz) user testing. Wizard-of-oz testing is a prototyping method where participants interact with a system that is controlled by the experimenter [10, 23]. This approach allows designers to explore and evaluate ideas before investing the time needed to build a working prototype. Each WOz session lasted an hour and was recorded and transcribed for further analysis. We analyzed the feedback using affinity diagramming, a popular analysis method in user interface design [50] where text data is segmented into notes and then clustered into groups.

We received feedback from seven people during the first round of WOz sessions, six of whom participated in the second user testing round. We mainly recruited non-researchers with minimal experience in qualitative data analysis. Only one person was formally trained in QDA methods. Six people were either actively working in community-based organizations or had helped community-based organizations analyze their data, and one person had never done sensemaking before.

3.2.2 Design Principles. Based on our literature review and personal sensemaking experiences, we identified four design principles (DP). We use our principles to guide the design of SenseMate and its main features.

(1) **Provide AI suggestions on demand (DP-1)**: In algorithm-in-the-loop systems, AI plays an assistive role. To promote human agency, SenseMate should provide AI recommendations on demand instead of by default.
(2) **Generate model explanations that are easy to judge and non-repetitive (DP-2):** Model explanations should be easy to judge to avoid model overreliance [62]. Prior work on mixed-initiative systems suggests presenting semantically meaningful recommendations in context to avoid repetitive explanations [20]. In addition, explanations should enable quick visual interpretation to prevent information overload [20, 44].
(3) **Create user-driven and intuitive processes to collect high-quality feedback on AI suggestions (DP-3):** The field of interactive machine learning emphasizes the importance of scaffolding the process of collecting high-quality

Table 1: Model classification performance by theme among the validation and test examples.

Theme	Accuracy	Precision	Recall	F-Score
Community Values	81.3%	0.86	0.75	0.80
Covid-19	88.9%	0.94	0.83	0.88
Housing Affordability	90.9%	0.91	0.91	0.91
Income	84.4%	0.92	0.75	0.83
Processes	65.0%	0.71	0.50	0.59
Quality of Education	88.6%	0.90	0.86	0.88
Race-Based Inequality	90.9%	0.93	0.89	0.91
Sense of Safety	84.4%	0.79	0.94	0.86
Transportation	87.5%	1.00	0.75	0.86

Table 2: Comparison between machine and human-generated rationales. The average balanced accuracy, precision, recall, and F-score are calculated across the positive validation and test examples.

Theme	Accuracy	Precision	Recall	F-Score
Community Values (n=16)	55.9 (8.0)%	0.26 (0.21)	0.2 (0.17)	0.2 (0.11)
Covid-19 (n=18)	82 (15.5)%	0.29 (0.22)	0.74 (0.3)	0.33 (0.18)
Housing Affordability (n=44)	81.9 (14.0)%	0.52 (0.27)	0.72 (0.26)	0.57 (0.22)
Income (n=16)	81.8 (15.5)%	0.39 (0.33)	0.76 (0.3)	0.48 (0.26)
Processes (n=50)	62.2 (11.6)%	0.33 (0.24)	0.33 (0.23)	0.32 (0.13)
Quality of Education (n=22)	68 (10.9)%	0.51 (0.27)	0.46 (0.26)	0.44 (0.22)
Race-Based Inequality (n=44)	83.8 (11.2)%	0.39 (0.29)	0.77 (0.22)	0.45 (0.23)
Sense of Safety (n=16)	71.8 (9.3)%	0.39 (0.2)	0.54 (0.16)	0.42 (0.17)
Transportation (n=8)	78 (14.8)%	0.39 (0.28)	0.68 (0.27)	0.41 (0.15)

and targeted user feedback in simple and intuitive ways, such as minor refinements to topic models and providing keywords to clustering algorithms [30, 59, 60]. Similarly, one of the design principles in mixed-initiative systems involves providing optional mechanisms for efficient human-machine collaboration in case users want to refine any analysis provided by the system. Previous work elaborates that these refinements should be easy and quick to do [36, 44].

(4) **Reduce model overreliance through simple design interventions (DP-4):** Model overreliance can be reduced through easy-to-understand model explanations [62]. In addition, simple cognitive forcing functions have also been found to reduce overreliance [11, 54].

3.2.3 Platform Layout and Coding Process. A major design component in SenseMate concerns the overall information layout and the coding process, or how users can assign themes to each unit of analysis. Figure 1 shows SenseMate's interface, which includes key features that make up the layout and coding process.

Grid Structure, Figure 1 (A). SenseMate is organized into two sections: the data (i.e. conversation snippets) on the left side and the codebook on the right side. When designing the overall layout of SenseMate, we took inspiration from existing QDA tools. The coding areas of these tools typically consist of multiple sections, each of which performs a certain function. We wanted to create a simpler interface that emulated an organized grid system. By separating the snippets from the codebook, both sections could be viewed alongside each other with independent scrolling, which can support multiple coding approaches.

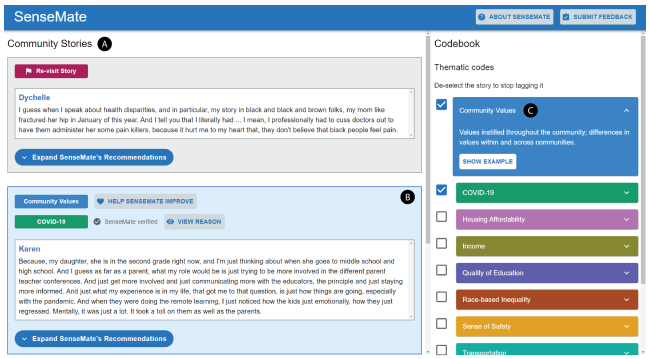


Figure 1: The overall layout of the SenseMate platform. Content is organized in two columns; the left side shows a series of community stories that users would code (A), and the right side shows the codebook that users select themes from. The second story is currently selected with the "Community Values" and "COVID-19" themes applied to it (B). Users can click on a code in the codebook to view its definition and an example (C).

Coding Action, Figure 1 (B). The coding action involves clicking on a snippet and then selecting the relevant codes in the codebook.

When a snippet is selected, the codebook section becomes a checklist. Clicking on a checkbox would assign a code to the selected snippet.

Code Definitions, Figure 1 (C). Users can click on a code in the codebook to view its definition and an example. We included this information, so people would not have to reference a separate document to access the codebook in its entirety (i.e. code names, definitions, and examples).

3.2.4 Theme Recommendations. Learning from previous work on semi-automated qualitative coding, we knew that human sensemakers want to retain total control over the coding process, which means having the agency to make the final call on all coding decisions [51]. As a result, we wanted AI models to play a supportive role, which required optional, intuitive, and fast human-AI interactions. Figure 2 depicts how people can interact with theme recommendations.

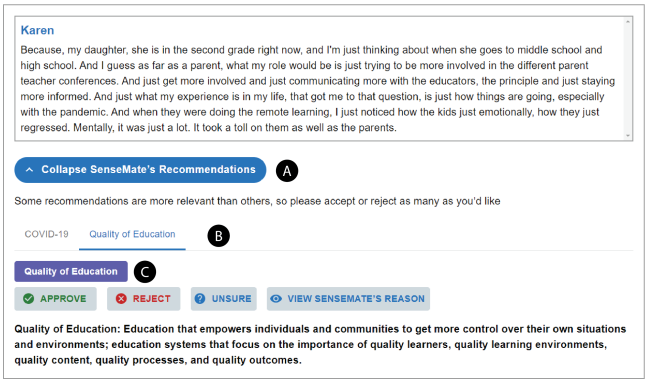


Figure 2: Users can choose to expand the "SenseMate's Recommendations" section for each story to view and interact with AI recommendations (A). From there, they can click through each recommendation via a tab bar (B). For each recommendation, users can perform various actions (C).

Expanding or Collapsing Recommendations, Figure 2 (A). We decided to move all the theme recommendations to the bottom of each snippet and have them hidden by default to provide AI suggestions when the user wants them DP-1. In addition, hiding the recommendations is an example of a "cognitive forcing function" in human-AI decision-making literature, which can reduce model overreliance [11, 54] DP-4. In general, participants from the user testing sessions would generally read and code the stories before viewing the recommendations, even when working with hundreds of stories. P7 would *"always read [the story] first, and then [she] would think about what [she] would do, and then look at the recommendations and see if they match up."* The ability to expand the recommendations allows users to think on their own before accessing the AI suggestions.

Recommendation Tab Bar, Figure 2 (B). Only one recommendation is shown at a time to prevent users from quickly approving all of the suggestions DP-4. Throughout the design process, we explored the pros and cons of showing one recommendation at a time versus showing all of them in a list. Participants appreciated how seeing one recommendation at a time was less overwhelming, but more effort was required to navigate between the AI suggestions. On the other hand, seeing all the recommendations at once made it easier to quickly act on each one, though the mockup appeared cluttered. We decided to combine the strengths of both designs by creating a tab bar with all the recommendations listed but only showing the details for one recommendation at a time to prevent information overload.

Recommendation Actions, Figure 2 (C). For each recommendation, users can complete the following actions: 1) approve the recommendation, 2) reject the recommendation, 3) mark the recommendation as unsure, or 4) view SenseMate's reason for the recommendation. The "view reason" button displays the model's explanations on demand DP-1. In general, participants from WOz testing appreciated the breadth of possible interactions with the recommendations. The quick actions provide intuitive ways of responding to a recommendation DP-3.

3.2.5 Model Explanations. An important aspect of SenseMate is the ability to view and interact with explanations for each theme recommendation. Figure 3 shows how model rationales are displayed within SenseMate, and Figures 4 and 5 detail the different ways participants can give feedback on model explanations.

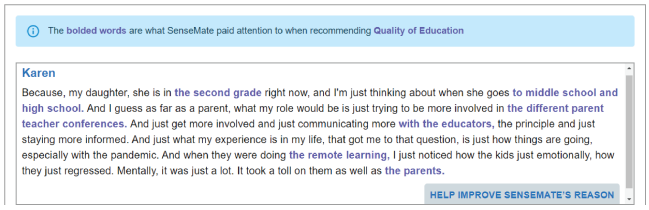


Figure 3: An example of how model rationales are displayed in SenseMate. The rationale is bolded within the story. Users can give feedback on the rationales by clicking on the "HELP IMPROVE SENSEMATE'S REASON" button.

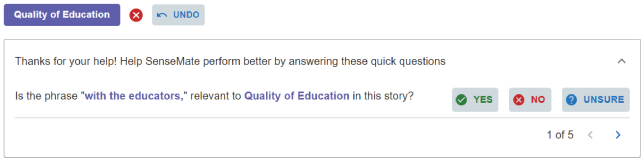


Figure 4: An example of the quick yes/no questions that get asked when a user rejects a recommendation. At most five questions are asked for each recommendation.

Bolded Rationales, Figure 3. Providing rationales can slow down the coding process by encouraging people to think carefully about whether to accept a recommendation or not. P3 emphasized that *"having the reasons as to why is a very needed part."* We bold the rationale within the story, so users can quickly view the explanation

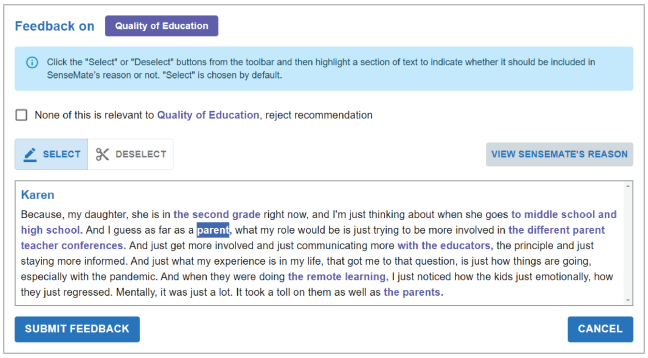


Figure 5: More nuanced feedback on the rationales can be provided through a highlighting interface. Users can add and remove words via the "SELECT" and "DESELECT" buttons.

without having to scroll somewhere else or read extra content. As a result, model explanations are easy to read and integrated into each snippet, making them non-repetitive DP-2. P4 confirms our design by stating, *"bolding is really essential. To link the color in the highlight to the actual color of the code is important."*

Quick Questions, Figure 4. When designing how users give feedback on the rationales, we decided not to ask for feedback through open-ended questions because they may be distracting and require a lot of energy to answer. We wanted to gather feedback through very simple questions since people would be more likely to answer them for multiple stories. In addition, several WOz participants wanted to provide feedback through multiple-choice questions. P6 said it would be *"better to have more narrowed down questions [because] it feels like [the platform] is more likely to listen to your feedback."* As a result, we decided to display optional yes/no questions after users reject a recommendation. The questions would ask if several phrases in the rationale relate to the recommended theme. An example quick question is in Figure 4. The questions provide another way for users to quickly view and evaluate model explanations DP-2. In addition, WOz participants appreciated how the questions were straightforward and specific. By answering yes or no, users can give feedback on what parts of the rationale make sense, which can help the models generate better explanations over time DP-3. Since the quick questions *"add an extra layer of thinking"*, participants preferred answering them for rejected recommendations. As P6 said, they *"give me an opportunity to think about [a recommendation] again."*

Highlighting Interface, Figure 5. The quick questions provide preliminary feedback on the rationales, but they have a few limitations. First, the questions don't allow users to add new phrases to rationales. Second, no feedback can be collected for themes that the models missed. Third, the questions don't allow for more nuanced feedback (e.g. this part of the phrase is relevant but not this part). To address these limitations, we constructed another way of providing feedback on the rationales through a highlighting interface. Within this interface, users can selectively include or exclude any word in a theme's rationale. We took inspiration from text editing tools to make the user experience as familiar as possible. In addition, we added signifiers in the form of colored text to indicate the current and modified rationales. The colored text helped WOz participants *"more efficiently give feedback on the rationales without having to read the whole story again"* DP-3. From user feedback, we observed that the highlighting interface is better for more relevant recommendations. WOz participants also preferred to use the highlighting interface for new themes, or themes that the models missed. P6 liked highlighting the new themes to *"verify what [she was] thinking."* P4 viewed the interaction as *"an investment of [her] analysis back into [SenseMate]"* DP-3.

3.3 Implementation Details

We implemented a prototype of SenseMate based on our third design iteration. The frontend was created using React[4], Redux[5], and Material UI[6]. The backend consisted of a Flask web framework[7], which was served with Nginx[8]. SenseMate was deployed to an EC2 instance. While implementing SenseMate, we strived to develop a minimal viable product (MVP). One implementation decision we made involved not re-training the models based on user feedback. Though closing the human-AI feedback loop will be critical in future versions of SenseMate, participants in the user study would not have spent enough time with the models to change their behavior. Only two to three snippets were associated with each theme, which is a very small sample size to fine-tune the models. Another major implementation decision we made was to carefully order the snippets and recommendations. Stories would be ordered by the number of recommendations and then by the story length, such that shorter stories with fewer recommendations would be at the top and more complex stories would be at the bottom. Within each story, theme recommendations would be arranged in descending order by model classification confidence. We selected these ordering strategies to scaffold the interactions between human sensemakers and the models.

4 USER STUDY

We conducted a summative evaluation of SenseMate through an online experiment with 180 participants. Participants were randomly assigned to one of three experiment conditions: receiving no AI assistance, receiving only theme recommendations, and receiving both theme recommendations and rationales. We aimed to address the following research questions:

(1) **How do varying levels of AI assistance impact coding efficiency? (RQ-1)**: We hypothesize that people who receive any AI assistance in the form of theme recommendations and rationales will spend less time on qualitative coding compared to those without AI support.
(2) **How do varying levels of AI assistance impact intercoder reliability? (RQ-2)**: We hypothesize that people who receive any AI assistance will have higher rates of intercoder reliability compared to those who don't receive AI assistance. In addition, people with the maximum level of AI assistance

[4]https://react.dev/.
[5]https://redux.js.org/.
[6]https://mui.com/.
[7]https://flask.palletsprojects.com/en/2.2.x/.
[8]https://www.nginx.com/.

in the form of theme recommendations and rationales will have the highest rates of intercoder reliability.

(3) **How do varying levels of AI assistance impact coding performance? (RQ-3)**: We define coding performance as the accuracy, precision, recall, and F-score of participant answers compared to expert gold labels. A high coding performance indicates a small difference between novice and expert coding decisions. Similar to intercoder reliability, we hypothesize that people who receive any AI assistance will have higher coding performances compared to those without AI support. We also predict that people with the maximum level of AI assistance will have higher coding performances than people who receive only theme recommendations.

In addition to the research questions, we strived to evaluate SenseMate's overall usability and the effectiveness of various human-AI interactions, such as viewing and giving feedback on theme recommendations and rationales.

4.1 Study Design

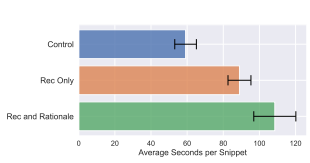
4.1.1 Participants. We recruited 180 participants (82 male, 92 female, 5 non-binary, and 1 transgender) from Prolific, a crowdsourcing platform. Section B.1 describes how we calculated the sample size. Most participants (N = 58) were 25 to 34 years old, followed by the 35 to 44 age range (N = 42). 69 participants completed high school as their highest level of education, and one-third of participants obtained a bachelor's degree. All participants resided in the United States and were able to communicate clearly in written and spoken English. Participants had varying levels of tech-savviness with 70% having little to no programming knowledge. Over 97% of participants felt confident using computers. Most participants had little (N = 89) to no (N = 49) prior experience in qualitative data analysis.

4.1.2 Procedure. Participants were randomly assigned to one of three conditions: 1) using SenseMate without any theme recommendations and rationales [**Control**], 2) using SenseMate with only theme recommendations [**Rec Only**], and 3) using SenseMate with theme recommendations and corresponding rationales [**Rec and Rationale**]. Participants were first presented with some general instructions and provided informed consent. From there, they were shown a short video tutorial on how to use the platform, which was tailored for each experiment condition. Once participants answered 12 comprehension check questions, they obtained access to the SenseMate platform. Everyone was asked to code the same collection of snippets, displayed in a particular order. We curated a sample of 10 snippets to cover different combinations of themes. The snippets ranged in length and number of themes. We excluded the "Processes" theme because even after including a definition and two examples, participants during a pilot study struggled to conceptualize the theme. "Processes" also has, by far, the worst-performing rationale extraction model among the validation and test examples. All other themes were represented by at least one snippet. After coding the 10 snippets, participants were directed to a Qualtrics post-survey, where they were asked to answer a series of usability and background questions. On average, participants took 35 minutes to complete the entire task and were paid $16.53 per hour.

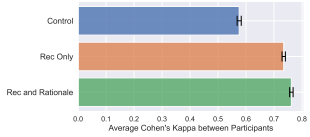
4.1.3 Analysis. To answer our research questions, we measured the following quantitative variables: **Time** (number of seconds it takes a participant to assign themes to each snippet), **Performance** (the accuracy, precision, recall, and F-score of a participant's answer for each snippet with respect to the gold labels), and **Reliability** (the Cohen's kappa between every unique pair of participants in each experiment condition based on their answers across all snippets). We also collected metrics on platform usage for each participant. These metrics include how often participants used various features in SenseMate, and the feedback participants provided for each recommendation and rationale. In the post-survey, we asked participants a series of Likert-scale questions (1 = strongly disagree and 7 = strongly agree) relating to the overall user experience. A few questions were adapted from the system usability scale (SUS) [5]. The Appendix (Section B.2) contains a list of all the Likert-scale questions.

Part of our analysis is pre-registered at https://aspredicted.org/P2C_41F. We created linear mixed-effect models for the "Time" and "Performance" metrics using the `lmer` function in the `lme4` R package [8]. The experiment condition was the fixed effect, and the user and snippet ids were random intercepts. (We logged the coding time outcome measure since it was heavily skewed toward higher values.) We created a linear regression model for "Reliability" with the experiment condition as the independent variable. We controlled for both users in each pair by including their demographic data and performed F-tests to determine any differences across the experiment conditions. In addition, we conducted an exploratory analysis of usage metrics to understand how participants interacted with the platform. This analysis provided insight into how our design decisions impacted people's coding experiences.

4.2 Results



(a) Average seconds spent on a snippet for each experiment condition.



(b) Average reliability, measured using Cohen's kappa, between pairs of participants from each experiment condition.

Figure 6: Impact of varying levels of AI assistance on coding time and intercoder reliability.

4.2.1 Impact of AI Assistance on Coding Time (RQ1). Figure 6a compares the time spent on each snippet across the experiment conditions. Participants in the control condition spent on average 59.1 seconds (σ = 74.8) per snippet, which is significantly less time compared to the 89.0 seconds (σ = 79.3) spent in the "Rec Only" condition and 108.5 seconds (σ = 145.5) spent in the "Rec and Rationale" condition. For context, participants would spend around 10 minutes coding 10 snippets compared to 18 minutes with access to the theme recommendations and rationales. The difference in

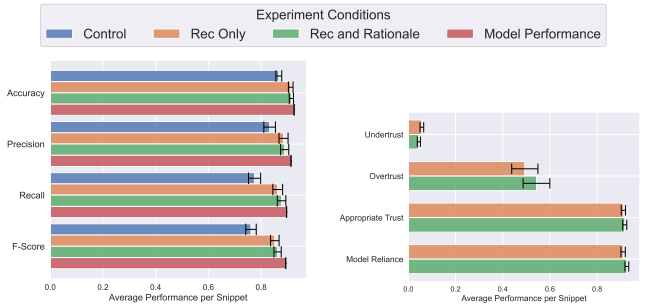
coding time is statistically significant according to the linear mixed-effect models ("Rec Only": $\hat{\beta} = 0.48, SE = 0.11, t = 4.45$; "Rec and Rationale": $\hat{\beta} = 0.59, SE = 0.11, t = 5.42$). After accounting for fixed and random effects, participants with the maximum level of AI assistance had an 80% higher geometric mean in coding time compared to those without any AI assistance. Participants who only had access to theme recommendations had a 62% increase in geometric mean for coding time. Though AI assistance in SenseMate did not make qualitative coding less time-consuming, participants in the treatment conditions still thought their experiences were productive. At the end of the task, participants were asked to rate from 1 to 7 whether they felt productive quickly and whether they were able to complete the task quickly. We did not observe any significant differences in **productivity** ("Control": $\mu = 6.2, \sigma = 0.9$; "Rec Only": $\mu = 6.2, \sigma = 0.9$; "Rec and Rationale": $\mu = 6.0, \sigma = 0.9$) and **working speed** ("Control": $\mu = 6.3, \sigma = 0.8$; "Rec Only": $\mu = 6.1, \sigma = 1.0$; "Rec and Rationale": $\mu = 5.9, \sigma = 1.1$) between the experiment conditions. In general, participants gave high ratings to both Likert-scale questions.

4.2.2 Impact of AI Assistance on Intercoder Reliability (RQ2). Figure 6b depicts a large difference in intercoder reliability between the control and treatment conditions. The average Cohen's kappa between pairs of participants in the control condition is 0.58 ($\sigma = 0.16$), which suggests moderate agreement. In comparison, the average Cohen's kappa is 0.73 ($\sigma = 0.13$) in the "Rec Only" condition and 0.76 ($\sigma = 0.13$) in the "Rec and Rationale" condition, which corresponds to substantial agreement. According to a linear regression model, intercoder reliability is on average an estimated 26% higher in the "Rec Only" condition ($\hat{\beta} = 0.15, SE = 0.005, t = 31$) and 29% higher in the "Rec and Rationale" condition ($\hat{\beta} = 0.17, SE = 0.005, t = 33$) relative to the control. The differences are significant (F(22, 5287) = 107.4, $p < 2.2 \times 10^{-16}$). As hypothesized, participants with the maximum level of AI assistance in the form of recommendations and rationales had the highest rates of intercoder reliability. Nonetheless, there is a small difference between the treatment conditions, which suggests that the theme recommendations had a larger impact on intercoder reliability compared to the rationales.

4.2.3 Impact of AI Assistance on Coding Performance (RQ3). As shown in Figure 7a, receiving assistance from the models had an observable effect on coding performance in terms of accuracy, precision, recall, and F-score (Table 5). Participants who received any form of AI assistance tended to select themes that were closer to expert-level decisions compared to those without the models. Accuracy and precision have smaller differences between the control and treatment conditions compared to recall and F-score. Participants with access to recommendations and rationales had slightly higher averages for F-score, recall, and precision than those who received only theme recommendations, but the differences are not significant.

Our observations in Figure 7a are confirmed by the linear mixed-effect models. Figure 9 in the Appendix displays the regression coefficient values, which are all statistically significant. F-score, a combination of precision and recall, is on average an estimated 10% higher in both treatment conditions relative to the control condition ("Rec Only": $\hat{\beta} = 0.09, SE = 0.02, t = 5.72$; "Rec and Rationale": $\hat{\beta} = 0.10, SE = 0.02, t = 6.39$). The treatment conditions have nearly twice as large coefficients for recall ("Rec Only": $\hat{\beta} = 0.09, SE = 0.02, t = 4.33$; "Rec and Rationale": $\hat{\beta} = 0.10, SE = 0.02, t = 5.06$) compared to precision ("Rec Only": $\hat{\beta} = 0.05, SE = 0.02, t = 3.30$; "Rec and Rationale": $\hat{\beta} = 0.06, SE = 0.02, t = 3.57$). On average, the AI assistance within SenseMate is more effective in helping people select the correct themes than avoiding incorrect themes. When examining whether humans with AI support perform better than AI alone, Figure 7a shows that the models have higher values compared to the averages in each experiment condition. For example, the models have the highest average F-score ($\mu = 0.89$) across the snippets, followed by the "Rec and Rationale" condition ($\mu = 0.86, \sigma = 0.17$), "Rec Only" condition ($\mu = 0.85, \sigma = 0.20$), and then the "Control" condition ($\mu = 0.76, \sigma = 0.25$). Since the model performance is already quite high, participants tended to follow the models' suggestions and human-AI collaboration did not add anything more to coding performance.

To better understand how participants utilized the models in SenseMate, we explored the dynamics of trust (appropriate trust, undertrust, overtrust) and model reliance across the treatment conditions, which is shown in Figure 7b. Section C.2 of the Appendix contains details on how these metrics were calculated. In general, model reliance is over 0.9 for both treatment conditions ("Rec Only": $\mu = 0.91, \sigma = 0.11$; "Rec and Rationale": $\mu = 0.93, \sigma = 0.10$). Model reliance does not reach 1.0, which suggests that there is some disagreement between the participants and models. Undertrust ("Rec Only": $\mu = 0.06, \sigma = 0.10$; "Rec and Rationale": $\mu = 0.04, \sigma = 0.08$) accounts for some of the disagreement. A value of 0.05 means that, on average, participants disagreed with 5% of the correct model predictions. Alternatively, overtrust ("Rec Only": $\mu = 0.49, \sigma = 0.44$; "Rec and Rationale": $\mu = 0.54, \sigma = 0.43$) accounts for the number of false positives and negatives from the models that people agreed with. A value of 0.5 implies that, on average, participants agreed with incorrect model predictions 50% of the time. Overtrust is significantly higher than undertrust, suggesting that users are more

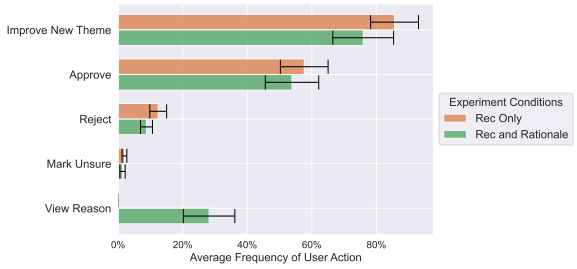


(a) Average accuracy, precision, recall, and F-score at a snippet-level for each experiment condition and when looking at the model performance by itself.

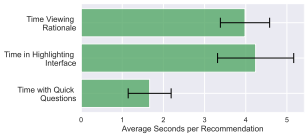
(b) Average model reliance and trust at a snippet-level for the "Rec Only" and "Rec and Rationale" conditions with 95% confidence intervals.

Figure 7: Impact of varying levels of AI assistance on coding performance.

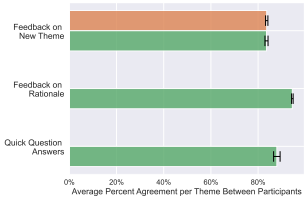
likely to follow an incorrect recommendation than ignore a correct one.



(a) Average frequency of various human-AI interactions among participants in the "Rec Only" and "Rec and Rationale" treatment conditions.



(b) Average amount of time participants in the "Rec and Rationale" condition spent viewing and giving feedback on rationales.



(c) Average agreement between pairs of participants on feedback provided at a theme level.

Figure 8: A summary of SenseMate's human-AI interaction patterns.

4.2.4 Usability and Interaction Patterns. In addition to answering the main research questions, we were curious about how participants viewed SenseMate's usability and interacted with different elements of the platform. These interactions served as another way of evaluating our design choices in addition to the user testing sessions. After interacting with SenseMate, participants reported high ratings (out of 7) on whether the platform was **easy to use** ("Control": $\mu = 6.6, \sigma = 0.6$; "Rec Only": $\mu = 6.3, \sigma = 0.8$; "Rec and Rationale": $\mu = 6.1, \sigma = 1.1$), **well integrated** ("Control": $\mu = 6.3, \sigma = 0.8$; "Rec Only": $\mu = 6.2, \sigma = 0.9$; "Rec and Rationale": $\mu = 6.1, \sigma = 1.0$), and **contained effective information** ("Control": $\mu = 6.4, \sigma = 0.7$; "Rec Only": $\mu = 6.4, \sigma = 0.7$; "Rec and Rationale": $\mu = 6.2, \sigma = 1.0$). In general, 98% of participants agreed that SenseMate was easy to use. We did notice a difference in rating between the control and treatment conditions, in which fewer participants in the "Rec and Rationale" condition thought SenseMate was easy to use ("Control": 100% agreement; "Rec Only": 98% agreement; "Rec and Rationale": 95% agreement). The addition of various ways to interact with the AI models might have made SenseMate more complicated for beginners.

Figure 8a shows the frequency of different actions that participants applied to the recommendations and new themes. "Improve New Theme" refers to moments when participants were willing to give feedback on themes that the models missed. On average, participants in the treatment conditions chose to give feedback on over 75% of new themes they selected ("Rec Only": $\mu = 85.6\%, \sigma = 28.8\%$; "Rec and Rationale": $\mu = 75.9\%, \sigma = 36.9\%$). Regarding the frequency of actions applied to AI suggestions, approving a recommendation had by far the highest rates ("Rec Only": $\mu = 57.7\%, \sigma = 29.2\%$; "Rec and Rationale": $\mu = 53.9\%, \sigma = 32.7\%$) compared to rejecting a recommendation ("Rec Only": $\mu = 12.4\%, \sigma = 10.3\%$; "Rec and Rationale": $\mu = 8.8\%, \sigma = 7.3\%$) or marking a recommendation as unsure ("Rec Only": $\mu = 2.0\%, \sigma = 2.9\%$; "Rec and Rationale": $\mu = 1.4\%, \sigma = 3.2\%$). In addition, participants in the "Rec and Rationale" condition chose to view the rationale for a recommendation around 28.2% of the time ($\sigma = 31.5\%$).

Figure 8b shows the average amount of time, in seconds, participants in the "Rec and Rationale" condition spent viewing and giving feedback on a rationale. On average, participants spent around 4.0 seconds ($\sigma = 11.1$) viewing a rationale (Figure 3), 4.2 seconds ($\sigma = 17.1$) using the highlighting interface to give feedback on the rationales (Figure 5), and 1.7 seconds ($\sigma = 9.7$) answering the quick questions (Figure 4). All of these interactions are quite short. The quick questions are a more efficient way of providing feedback compared to the highlighting interface. Assuming a snippet has five recommendations and a user chooses to view and give feedback on each recommendation, they would only spend around 2 extra minutes on the snippet. In terms of feedback quality, Figure 8c illustrates that participants obtained high agreement in their feedback on the rationales. All of the average agreements are above 80%. The average agreement on feedback for themes that the models missed (i.e. new themes) is around 83%. The average agreement on the rationale feedback is 94.5% ($\sigma = 8.1\%$), which is higher than the average agreement between user feedback and the original rationales ($\mu = 92.9\%, \sigma = 7.5\%$). Participants have higher agreement among each other than with the original machine-generated rationales, suggesting that participants can collectively improve the rationales instead of confusing the models with conflicting feedback.

5 DISCUSSION AND FUTURE WORK

While building and evaluating SenseMate, we gained a thorough understanding of what a semi-automated qualitative coding platform could and should support. We reflect on our findings and suggest several design implications for human-AI collaboration in sensemaking efforts.

5.1 Lessons Learned from Applying Rationale Extraction Models to Qualitative Coding

5.1.1 Reflections on Coding Time (RQ1). We found that participants with access to machine-generated recommendations and rationales spent the most time on qualitative coding followed by those who only received theme recommendations and then participants without AI support. Contrary to our original hypothesis, the introduction of rationale extraction models did not make qualitative coding less time-consuming. Limitations in the user study may have impacted how AI support affects coding time. Due to time and cost constraints, we only asked participants to analyze 10 snippets, which does not represent a realistic qualitative coding experience. We attempted to mitigate this limitation through careful sampling of the snippets. Even so, the small sample of snippets may explain our finding of increased coding time between the treatment and control conditions. Participants in the treatment conditions could

have spent more time on the task while learning how to interact with the AI models, and the task may have ended just as they were becoming more familiar with the platform. Thus, an increase in efficiency may be observed when users interact with SenseMate to code more data. The creators of Cody, another qualitative coding platform, similarly found that AI suggestions benefit coding quality rather than speed. The authors mentioned that improving quality can reduce the workload in the long run because users would have to spend less time correcting coding errors [57]. Though we did not find an immediate increase in coding efficiency, it is worth exploring whether any time is saved when using SenseMate in the field. In addition, we observed a difference between objective and subjective measures of coding efficiency. Though participants with AI assistance spent more time on the task, they still reported high ratings in terms of feeling productive and being able to complete the task quickly. When evaluating the effectiveness of semi-automated sensemaking, researchers should not only consider whether the presence of AI models reduces the absolute coding time but also if the models encourage people to be productive and spend more time in the analytical zone.

5.1.2 Reflections on Intercoder Reliability (RQ2). Intercoder reliability is a measure of how consistently a codebook is applied to the data. As hypothesized, we found that participants who received any AI assistance had higher rates of intercoder reliability compared to those without access to the models. In addition, participants with the maximum level of AI assistance in the form of theme recommendations and rationales obtained the highest rates of intercoder reliability. Without obtaining high reliability, patterns or insights derived from the data may be flawed, which could negatively impact downstream decision-making. High rates of intercoder reliability are especially helpful during deductive coding where sensemakers start with a predefined set of thematic codes, which are assigned to qualitative data. The AI support in SenseMate can help users stay grounded in the codebook and minimize coding errors, which can speed up the analysis. Though the rationale extraction models can increase intercoder reliability, there could be unintended consequences, such as high model overreliance. When AI dominates the decision-making process, the reliability will naturally increase while harming human agency. This pattern could be harmful to inductive coding, in which thematic codes are determined using a bottom-up approach and disagreements between sensemakers can highlight the most interesting areas of the dataset. It is important to consider the potential impacts of model overreliance on intercoder reliability. While designing SenseMate, we intentionally created several features to reduce model overreliance, including cognitive forcing functions DP-4. From the user study, we detected high rates of appropriate trust and lower rates of undertrust and overtrust (Figure 7b). Though participants tended to agree with AI suggestions, we did not observe signs of high model overreliance.

5.1.3 Reflections on Coding Performance (RQ3). We found that participants with access to the rationale extraction models had higher coding performance in terms of accuracy, precision, recall, and F-score compared to those without AI support, which supports our initial hypothesis. Participants who received the maximum level of AI assistance had slightly higher coding performance than participants who only had access to AI recommendations. Accuracy, precision, recall, and F-score were measured relative to expert gold labels, so a high coding performance indicates a small difference between novice and expert coding decisions. Rationale extraction models can help novice sensemakers more efficiently reach a higher quality standard. However, this finding may depend on the model performance, in which models need to achieve high coding performance on their own to help novices. The rationale extraction models had high classification performance with an average F-score of 0.89 on the snippets from the user study. When comparing model-only coding performance with the average coding performance among participants in each experiment condition, model-only performance remained the highest. Novices from the user study were not able to surpass model-only performance after getting access to the theme recommendations. Zhang et al. suggest that humans need to bring in unique knowledge to complement AI errors [67]. Consequently, beginner sensemakers with a wealth of information about the community represented in the data may provide more complementary knowledge compared to individuals without any context.

5.2 Design Implications for Human-AI Sensemaking Systems

5.2.1 First Impressions on AI Assistance. Through our wizard-of-oz (WOz) user testing, we found that initial impressions on AI assistance matter, especially when users have to analyze a large dataset. People would interact with the recommendations less if the first few were not very accurate. For example, P5, from the WOz sessions, shared that if the recommendations were *"taking things too literally,"* then she would not look at them. P6 would stop viewing recommended themes *"if there was always one outlandish recommendation coming up."* P7 mentioned the importance of trust. If the recommendations were *"pretty accurate after the first few,"* she would trust them more and hence use them *"earlier on in the decision-making process."* With less trust, she would *"just use [them] more as confirmation."* The importance of positive first impressions motivated the careful ordering of snippets within SenseMate. We observed that participants usually code in sequential order, so we placed shorter stories with fewer recommendations at the top to help users more quickly evaluate the models. We also ordered theme recommendations at the snippet level according to model confidence to increase the likelihood of having a positive first impression of model suggestions. Additionally, we strived to create rationales that are easy to judge and non-repetitive DP-2, so users can more effectively evaluate AI suggestions. From the user study, we observed that participants in the "Rec and Rationale" condition viewed, on average, 28% of the rationales, which contrasts with results from prior studies where explanations are desired but rarely used because they take too much time to evaluate [57]. When designing human-AI collaborative tools, we recommend factoring in possible first impressions. Intentional scaffolding can promote positive initial interactions with AI, which can result in more collaborative human-AI relationships.

5.2.2 Varied and Efficient Feedback Mechanisms. DP-3 involves creating user-driven and intuitive processes to collect feedback on

AI suggestions. To accomplish this design principle, we learned that it is important to provide different ways of giving feedback. The highlighting interface allows users to provide nuanced feedback on model rationales, which is helpful for approved recommendations and new themes. As one participant in the user study reflected, *"the process of highlighting words that matched with new theme recommendations that I made was great."* On the other hand, the quick questions are more helpful for rejected recommendations. Besides having access to a variety of feedback mechanisms, users wanted to provide feedback without significantly increasing their workload, which meant answering simple closed-ended questions. P4 from the WOz sessions did not want the feedback mechanisms to feel like she was interacting with a human sensemaker because it would take *"too much emotional energy"* to *"attend to it like a person."* Participants appreciated how straightforward and efficient the quick questions and highlighting interface were. From the user study, we found that the feedback mechanisms are straightforward and fast to utilize. Participants only spent a few seconds at a time giving feedback on the rationales. In addition, we observed high agreement in user feedback, which contrasts with Ghai et al.'s concern that user feedback on AI explanations can have high variance, which can negatively impact how much a model improves [30]. The carefully designed highlighting interface and quick questions may have resulted in higher-quality feedback because they are less open-ended. As a result, we recommend designing a variety of feedback mechanisms in human-AI systems that prioritize ease of use and quick interactions. Effective feedback mechanisms not only improve the models but also promote human learning, an important outcome of sensemaking [29]. Features such as the quick questions create valuable learning opportunities for users to reflect on their decisions, which can inform the design of future QDA platforms.

5.2.3 Choosing When to Receive AI Assistance. A major design principle we followed was DP-1, or providing AI suggestions on demand. While talking with potential users, we noticed that optional AI support is critical to provide flexibility during sensemaking. The ability to decide when to view AI suggestions supports different types of users. During the WOz sessions, we observed that some participants were more willing to use the recommendations than others. P6 mentioned that since she has a habit of second-guessing herself, she *"would probably view recommended themes every time."* P3 also finds the recommendations helpful for similar reasons: *"sometimes I will just ruminate on one highlight for a long time, so having the recommended tags is also another way of me getting that support of knowing there's something here that's helping me analyze this... ultimately the decision is mine of what to tag it with, but at least having that extra support is really helpful."* On the other hand, experienced sensemakers, like P4, may be less likely to interact with the recommendations. P4 described the differences between coding independently and coding with SenseMate's help, highlighting that both forms of coding are important. Coding independently involves *"focusing on the text and [one's] own thoughts."* It is similar to working with *"Play-Doh"*, in which everything is collapsed and you can make whatever you want. Coding with SenseMate is *"working with a system,"* and is comparable to *"Legos"*, in which there is a constant back-and-forth between the user and instructions. A system that supports qualitative coding should allow the user to freely pivot between these two modes.

5.3 Ethical Considerations

The use of a system such as SenseMate raises several ethical concerns. First, model misclassifications can result in misinterpretation and bad decision-making, so it's critical for humans to stay in control of the sensemaking process. Second, the long-term effects of using SenseMate remain unclear. Users may become overly dependent on AI recommendations, especially since the models are intended to improve over time. Cognitive forcing functions and generally slowing down the process of interacting with AI models can reduce overreliance. Interventions we explored during the design process include: letting sensemakers code without seeing the recommendations, encouraging users to interact with one recommendation at a time, and asking people who approve recommendations too quickly to confirm their actions. Third, there is the risk of missing important patterns in the data, especially as users start to trust the models more. It will be important to evaluate what information SenseMate is ignoring before deploying it.

5.4 Opportunities for Future Research

SenseMate offers a range of exciting possibilities for future research in the automated sensemaking space.

5.4.1 Applying Large Language Models (LLMs). One area of future work is to experiment with ways to apply more powerful language models to semi-automate the entire sensemaking process. At the tail end of our work, ChatGPT and then GPT-4 were released. Both models have proven to be tremendously powerful in many analytical tasks. To get a sense of how useful LLMs would be for qualitative coding, we conducted a preliminary investigation into the feasibility of using ChatGPT for rationale extraction. We gave ChatGPT examples of stories with ambiguous codes like "Processes" and "Community Values" and asked the model to find concise and verbatim parts of the story that are related to a particular theme. We found that ChatGPT was able to explain the connection between stories and themes that were false negatives according to the rationale extraction models. Though ChatGPT may perform better with ambiguous themes and snippets, it isn't clear how the model produces these rationales and how users can impact ChatGPT's responses. Prior work highlights concerns with utilizing LLMs for qualitative coding, citing issues such as lack of transparency, consistency challenges, and data privacy risks [65, 66]. Users grapple with a lack of control over the outputs, making it challenging to achieve proper alignment, particularly on a handful of examples. In contrast, rationale extraction models can be trained locally without significant runtime costs, and user feedback can contribute to the fine-tuning of model parameters. Importantly, SenseMate's contribution extends beyond model choice: its design integrates feedback mechanisms directly into the coding process, eliminating the need for prompt engineering, which requires experimentation and expertise. Explanations are seamlessly displayed within the text without requiring additional content reading. As aforementioned concerns get addressed, SenseMate can incorporate more powerful LLMs,

while retaining the effective features identified through our design process and user study.

5.4.2 Designing Collaborative Sensemaking Platforms. We built SenseMate to support collaboration between AI and an individual user. Ideally, the AI model would improve over time based on user feedback. However, if the model only takes in feedback from one person, then it would start to emulate the person's coding behaviors, including their biases, which can skew the analysis. One way of incorporating multiple perspectives when analyzing qualitative data is through collaborative sensemaking [18, 24, 28]. An exciting area of future work is to design and evaluate collaborative sensemaking platforms that integrate AI assistance and group deliberation. For example, AI models can strategically delegate work using deferral systems [41] or flag ambiguous data for future group discussions [15, 24]. It is important to consider the impacts of implicit social incentives when designing group-AI systems for sensemaking [12]. For example, prior research has found that informing a human user that AI assistance has been utilized by others can increase the user's adherence to AI's suggestions [3]. Intentional design decisions would have to be made around how information is shared between AI and a group of sensemakers.

6 CONCLUSION

In summary, our research tackles a pressing challenge faced by community-centered organizations in gathering and analyzing qualitative data to understand constituent needs and perspectives. One of the most tedious and time-consuming parts of qualitative data analysis is qualitative coding, or identifying themes in the entire dataset. A challenge in qualitative coding involves attaining high intercoder reliability. Additionally, beginner sensemakers may struggle to produce similar quality coding decisions as experts who are formally trained in qualitative data analysis. By harnessing rationale extraction models, a new machine learning method to semi-automate qualitative coding, and an iterative, human-centered design process, we have developed SenseMate, a novel human-AI system for people with minimal sensemaking experience. Through a comprehensive user study, we demonstrate that access to AI assistance in the form of theme recommendations and rationales significantly improves intercoder reliability by 29% while increasing the alignment between novice and expert coding decisions. However, we found that the models increased coding time by 49 seconds per unit of analysis, which may have been caused by having to learn a new system while only coding a small number of snippets. Though users with AI support had higher coding times, they still thought their experiences were productive. In addition, participants reported that SenseMate was generally easy to use. As the AI community tackles challenges in privacy and consistency when utilizing LLMs to analyze qualitative community data, SenseMate can adapt these models, while maintaining the user-driven and intuitive feedback mechanisms verified through our iterative design process and user study. We hope the design explorations and lessons in our paper will inspire future endeavors to make qualitative data analysis more accessible to non-researchers.

ACKNOWLEDGMENTS

Thanks to all who tested out and provided feedback on SenseMate. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2141064. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Marissa D Abram, Karen T Mancini, and R David Parker. 2020. Methods to integrate natural language processing into qualitative research. *International Journal of Qualitative Methods* 19 (2020), 1609406920984608.
[2] Corey M Abramson, Jacqueline Joslyn, Katharine A Rendle, Sarah B Garrett, and Daniel Dohan. 2018. The promises of computational ethnography: Improving transparency, replicability, and validity for realist approaches to ethnographic analysis. *Ethnography* 19, 2 (2018), 254–284.
[3] Veronika Alexander, Collin Blinder, and Paul J Zak. 2018. Why trust an algorithm? Performance, cognition, and neurophysiology. *Computers in Human Behavior* 89 (2018), 279–288.
[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[5] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.
[6] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.
[7] Tehmina Basit. 2003. Manual or electronic? The role of coding in qualitative data analysis. *Educational research* 45, 2 (2003), 143–154.
[8] Douglas Bates, Martin Maechler, Ben Bolker, Steven Walker, Rune Haubo Bojesen Christensen, Henrik Singmann, Bin Dai, Fabian Scheipl, and Gabor Grothendieck. 2009. Package 'lme4'. *URL http://lme4. r-forge. r-project. org* (2009).
[9] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
[10] Jacob T Browne. 2019. Wizard of oz prototyping for machine learning experiences. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.
[11] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z Gajos. 2021. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–21.
[12] Jason W Burton, Mari-Klara Stein, and Tina Blegind Jensen. 2023. Beyond Algorithm Aversion in Human-Machine Decision-Making. In *Judgment in Predictive Analytics*. Springer, 3–26.
[13] Senthil Chandrasegaran, Sriram Karthik Badam, Lorraine Kisselburgh, Karthik Ramani, and Niklas Elmqvist. 2017. Integrating visual analytics support for grounded theory practice in qualitative text analysis. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 201–212.
[14] Eshwar Chandrasekharan, Chaitrali Gandhi, Matthew Wortley Mustelier, and Eric Gilbert. 2019. Crossmod: A cross-community learning-based system to assist reddit moderators. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–30.
[15] Nan-Chen Chen, Margaret Drouhard, Rafal Kocielnik, Jina Suh, and Cecilia R Aragon. 2018. Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 8, 2 (2018), 1–20.
[16] Nan-chen Chen, Rafal Kocielnik, Margaret Drouhard, Vanessa Peña-Araya, Jina Suh, Keting Cen, Xiangyi Zheng, and Cecilia R Aragon. 2016. Challenges of applying machine learning to qualitative coding. In *ACM SIGCHI Workshop on Human-Centered Machine Learning*.
[17] Hao-Fei Cheng, Ruotong Wang, Zheng Zhang, Fiona O'Connell, Terrance Gray, F Maxwell Harper, and Haiyi Zhu. 2019. Explaining decision-making algorithms through UI: Strategies to help non-expert stakeholders. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–12.
[18] Chun-Wei Chiang, Zhuoran Lu, Zhuoyan Li, and Ming Yin. 2023. Are Two Heads Better Than One in AI-Assisted Decision Making? Comparing the Behavior and Performance of Groups and Individuals in Human-AI Collaborative Recidivism Risk Assessment. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.

[19] Eric Chu, Deb Roy, and Jacob Andreas. 2020. Are visual explanations useful? a case study in model-in-the-loop prediction. *arXiv preprint arXiv:2007.12248* (2020).

[20] Kristin Cook, Nick Cramer, David Israel, Michael Wolverton, Joe Bruce, Russ Burtner, and Alex Endert. 2015. Mixed-initiative visual analytics using task-driven recommendations. In *2015 IEEE conference on visual analytics science and technology (VAST)*. IEEE, 9–16.

[21] Kevin Crowston, Xiaozhong Liu, and Eileen E Allen. 2010. Machine learning and rule-based automated coding of qualitative data. *proceedings of the American Society for Information Science and Technology* 47, 1 (2010), 1–2.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[23] Steven Dow, Blair MacIntyre, Jaemin Lee, Christopher Oezbek, Jay David Bolter, and Maribeth Gandy. 2005. Wizard of Oz support throughout an iterative design process. *IEEE Pervasive Computing* 4, 4 (2005), 18–26.

[24] Margaret Drouhard, Nan-Chen Chen, Jina Suh, Rafal Kocielnik, Vanessa Pena-Araya, Keting Cen, Xiangyi Zheng, and Cecilia R Aragon. 2017. Aeonium: Visual analytics to support collaborative qualitative coding. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 220–229.

[25] Steven M Drucker, Danyel Fisher, and Sumit Basu. 2011. Helping users sort faster with adaptive machine learning recommendations. In *Human-Computer Interaction–INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings, Part III 13*. Springer, 187–203.

[26] John J Dudley and Per Ola Kristensson. 2018. A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 8, 2 (2018), 1–37.

[27] Cristian Felix, Aritra Dasgupta, and Enrico Bertini. 2018. The exploratory labeling assistant: Mixed-initiative label curation with large document collections. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 153–164.

[28] Abbas Ganji, Mania Orand, and David W McDonald. 2018. Ease on Down the Code: Complex Collaborative Qualitative Coding Simplified with'Code Wizard'. *Proceedings of the ACM on human-computer interaction* 2, CSCW (2018), 1–24.

[29] Simret Araya Gebreegziabher, Zheng Zhang, Xiaohang Tang, Yihao Meng, Elena L Glassman, and Toby Jia-Jun Li. 2023. Patat: Human-ai collaborative qualitative coding with explainable interactive rule synthesis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[30] Bhavya Ghai, Q. Vera Liao, Yunfeng Zhang, Rachel Bellamy, and Klaus Mueller. 2021. Explainable Active Learning (XAL): Toward AI Explanations as Interfaces for Machine Teachers. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW3, Article 235 (jan 2021), 28 pages. https://doi.org/10.1145/3432934

[31] Ariel Goldman, Cindy Espinosa, Shivani Patel, Francesca Cavuoti, Jade Chen, Alexandra Cheng, Sabrina Meng, Aditi Patil, Lydia B Chilton, and Sarah Morrison-Smith. 2022. QuAD: Deep-Learning Assisted Qualitative Data Analysis with Affinity Diagrams. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–7.

[32] Philipp Grandeit, Carolyn Haberkern, Maximiliane Lang, Jens Albrecht, and Robert Lehmann. 2020. Using BERT for qualitative content analysis in psychosocial online counseling. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*. 11–23.

[33] Ben Green and Yiling Chen. 2019. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–24.

[34] Peter Green and Catriona J MacLeod. 2016. SIMR: An R package for power analysis of generalized linear mixed models by simulation. *Methods in Ecology and Evolution* 7, 4 (2016), 493–498.

[35] Timothy C Guetterman, Tammy Chang, Melissa DeJonckheere, Tanmay Basu, Elizabeth Scruggs, and VG Vinod Vydiswaran. 2018. Augmenting qualitative text analysis with natural language processing: methodological study. *Journal of medical Internet research* 20, 6 (2018), e231.

[36] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166.

[37] Annette Hoxtell. 2019. Automation of qualitative content analysis: A proposal. In *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, Vol. 20.

[38] Maggie Hughes, Dimitra Dimitrakopoulou, Maridena Rojas, and Somala Diby. 2023. Digital Civic Sensemaking: Computer-Supported Participatory Sensemaking of Nuanced, Experience-Based Dialogue. (2023).

[39] Jialun Aaron Jiang, Kandrea Wade, Casey Fiesler, and Jed R Brubaker. 2021. Supporting serendipity: Opportunities and challenges for Human-AI Collaboration in qualitative analysis. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–23.

[40] Alexander Keller and Hans Achatz. 2019. Reading Between the Lines of Qualitative Data–How to Detect Hidden Structure Based on Codes. (2019).

[41] Vijay Keswani, Matthew Lease, and Krishnaram Kenthapadi. 2021. Towards unbiased and accurate deferral to multiple experts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 154–165.

[42] Vivian Lai, Samuel Carton, Rajat Bhatnagar, Q Vera Liao, Yunfeng Zhang, and Chenhao Tan. 2022. Human-AI Collaboration via Conditional Delegation: A Case Study of Content Moderation. In *CHI Conference on Human Factors in Computing Systems*. 1–18.

[43] Vivian Lai and Chenhao Tan. 2019. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the conference on fairness, accountability, and transparency*. 29–38.

[44] Tak Yeon Lee, Alison Smith, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. 2017. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies* 105 (2017), 28–42.

[45] Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 107–117. https://doi.org/10.18653/v1/D16-1011

[46] Robert P Lennon, Robbie Fraleigh, Lauren J Van Scoy, Aparna Keshaviah, Xindi C Hu, Bethany L Snyder, Erin L Miller, William A Calo, Aleksandra E Zgierska, and Christopher Griffin. 2021. Developing and testing an automated qualitative assistant (AQUA) to support qualitative analysis. *Family Medicine and Community Health* 9, Suppl 1 (2021).

[47] Jasy Suet Yan Liew, Nancy McCracken, Shichun Zhou, and Kevin Crowston. 2014. Optimizing features in active machine learning for complex qualitative content analysis. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. 44–48.

[48] Han Liu, Vivian Lai, and Chenhao Tan. 2021. Understanding the effect of out-of-distribution examples and interactive explanations on human-ai decision making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–45.

[49] Chi-Jung Lu and Stuart W Shulman. 2008. Rigor and flexibility in computer-based qualitative research: Introducing the Coding Analysis Toolkit. *International Journal of Multiple Research Approaches* 2, 1 (2008), 105–117.

[50] Andrés Lucero. 2015. Using affinity diagrams to evaluate interactive prototypes. In *Human-Computer Interaction–INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, September 14-18, 2015, Proceedings, Part II 15*. Springer, 231–248.

[51] Megh Marathe and Kentaro Toyama. 2018. Semi-automated coding for qualitative research: A user-centered inquiry and initial prototypes. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.

[52] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2018. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682* (2018).

[53] Laura K Nelson. 2020. Computational grounded theory: A methodological framework. *Sociological Methods & Research* 49, 1 (2020), 3–42.

[54] Joon Sung Park, Rick Barber, Alex Kirlik, and Karrie Karahalios. 2019. A slow algorithm improves users' assessments of the algorithm's accuracy. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–15.

[55] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[56] Tim Rietz and Alexander Maedche. 2020. Towards the Design of an Interactive Machine Learning System for Qualitative Coding.. In *ICIS*.

[57] Tim Rietz and Alexander Maedche. 2021. Cody: An AI-based system to semi-automate coding for qualitative research. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[58] Johnny Saldaña. 2021. The coding manual for qualitative researchers. *The coding manual for qualitative researchers* (2021), 1–440.

[59] Ehsan Sherkat, Seyednaser Nourashrafeddin, Evangelos E Milios, and Rosane Minghim. 2018. Interactive document clustering revisited: A visual analytics approach. In *23rd International Conference on Intelligent User Interfaces*. 281–292.

[60] Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. 2018. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *23rd International Conference on Intelligent User Interfaces*. 293–304.

[61] Patrick J Tierney. 2020. A qualitative analysis framework using natural language processing and graph theory. *International Review of Research in Open and Distributed Learning* 13, 5 (2020), 173–189.

[62] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael Bernstein, and Ranjay Krishna. 2022. Explanations Can Reduce Overreliance on AI Systems During Decision-Making. *arXiv preprint arXiv:2212.06823* (2022).

[63] Xinru Wang and Ming Yin. 2021. Are explanations helpful? a comparative study of the effects of explanations in ai-assisted decision-making. In *26th International Conference on Intelligent User Interfaces*. 318–328.

[64] John Wenskovitch, Corey Fallon, Kate Miller, and Aritra Dasgupta. 2021. Beyond Visual Analytics: Human-Machine Teaming for AI-Driven Data Sensemaking. In *2021 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*. IEEE, 40–44.
[65] He Zhang, Chuhao Wu, Jingyi Xie, ChanMin Kim, and John M Carroll. 2023. QualiGPT: GPT as an easy-to-use tool for qualitative coding. *arXiv preprint arXiv:2310.07061* (2023).
[66] He Zhang, Chuhao Wu, Jingyi Xie, Yao Lyu, Jie Cai, and John M Carroll. 2023. Redefining qualitative analysis in the AI era: Utilizing ChatGPT for efficient thematic analysis. *arXiv preprint arXiv:2309.10771* (2023).
[67] Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 295–305.
[68] Zhiyong Zhang, Yujiao Mai, Miao Yang, and Maintainer Zhiyong Zhang. 2018. Package 'WebPower'. *Basic and Advanced Statistical Power Analysis Version* 72 (2018).

A DESCRIPTION OF THEMATIC CODES

Table 3 contains a description of the themes we classified.

B USER STUDY DESIGN

B.1 Power Analysis

To determine the sample size for the user study, we conducted a pilot study with 20 people. We applied the `WebPower` and `simr` packages in R [34, 68]- and specifically the `kanova` and `powerCurve` functions- to compute sample size requirements for a given type I error rate (α) and power level (β). We set α and β at conventional levels of 0.05 and 0.8, respectively. The `kanova` function conducts power analyses for k-way ANOVAs. The function requires specifying the numerator degrees of freedom `ndf` (2), effect size `f` (Cohen's f), total number of groups `ng` (3), `alpha` (0.05), and `power` (0.8). The `powerCurve` function conducts power analyses for linear mixed-effect models by running multiple simulations at varying sample sizes. From the power analyses, we discovered that we would need at least 40 participants per condition to achieve a power of 80% for coding performance. We decided to increase our sample size to 180 participants, or 60 per condition.

B.2 Usability Questions

Table 4 lists all the Likert-scale questions we asked participants after they completed the qualitative coding task. Participants in the treatment conditions ("Rec Only" and "Rec and Rationale") received additional Likert-scale questions about how much they understood and trusted the recommendations, how much control they had over the recommendations, and whether they thought the recommendations (and rationales) were accurate.

C ADDITIONAL USER STUDY RESULTS

C.1 Coding Performance Metrics Across Experiment Conditions

Table 5 contains the means and standard deviations of coding accuracy, precision, recall, and F-score for each experiment condition. These values complement the patterns shown in Figure 7a. Figure 9 displays the regression coefficient values for coding accuracy, precision, recall, and F-score, which are all statistically significant.

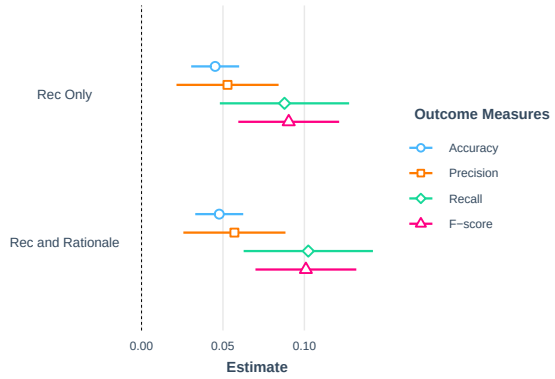


Figure 9: Associations between coding performance and treatment conditions, relative to the control condition. Circles represent average associations (i.e. regression coefficient values), and lines represent 95% confidence intervals. Intervals that do not intersect zero indicate statistically significant associations.

C.2 Analysis on Model Reliance and Trust

To better understand how participants utilized the models in SenseMate, we explored the dynamics of trust and model reliance across the treatment conditions, which is shown in Figure 7b. We first filtered out 64 cases where participants chose not to expand the recommendations for a snippet. Only 14 out of 120 participants (9 from "Rec and Rationale" and 5 from "Rec Only") did not view the recommendations for all 10 snippets. From there, we calculated various metrics relating to trust, taking inspiration from prior work in human-AI collaboration [63]. Undertrust is the proportion of themes where participants chose not to apply a correct recommendation relative to the number of correct model predictions. Overtrust represents the proportion of themes where participants applied an incorrect recommendation relative to the number of incorrect model predictions. Appropriate trust is the proportion of themes where participants applied correct model predictions and ignored incorrect ones. Model reliance is the agreement between participant and model answers for a snippet.

To complement the trust metrics in Figure 7b, we asked participants in the treatment conditions to rate (from 1 to 7) how much they understood, trusted, and felt in control of the models. Participants gave high ratings on all three factors: **understanding** ("Rec Only": $\mu = 6.3, \sigma = 0.8$; "Rec and Rationale": $\mu = 6.1, \sigma = 0.8$), **trust** ("Rec Only": $\mu = 5.7, \sigma = 1.0$; "Rec and Rationale": $\mu = 5.7, \sigma = 1.1$), and **agency** ("Rec Only": $\mu = 6.0, \sigma = 0.8$; "Rec and Rationale": $\mu = 5.8, \sigma = 1.0$). The ratings for understanding tended to be slightly higher compared to those for trust and agency. Trust has the lowest average ratings, yet 92% of participants agreed to some extent that they trusted the theme recommendations.

Table 3: A description of the 9 themes we focus on, including their names and definitions. In addition, the second column depicts the number of unique snippets that contain each theme.

Theme Name and Definition	# Examples
Community Values: values instilled throughout the community and differences in values within and across communities	85
Covid-19: COVID-19, vaccines, masks, COVID tests, boosters, and the impacts of COVID-19, such as working from home, school closures, and jobs lost	93
Housing Affordability: cost of housing and how affordable that cost is to residents, regardless of tenure (tenant/owner) and subsidy (e.g. workforce housing, public housing)	222
Income: references to income/wages and wealth. This can include discussions about: one's personal income; satisfaction with their income; in/ability to increase their income; in/ability to build wealth; income inequality; the income/wage levels to be able to afford the cost of living in Boston	81
Processes: references to processes through which the public interfaces with government, such as voting, community engagement, campaigning, electoral processes, and other decision-making processes	257
Quality of Education: education that empowers individuals and communities to get more control over their own situations and environments; education systems that focus on the importance of quality learners, quality learning environments, quality content, quality processes, and quality outcomes	118
Race-Based Inequality: defined as lack of jobs, services, and goods based on skin color, ethnicity, and language	224
Sense of Safety: refers to feeling unsafe at home, in one's neighborhood, and throughout the city	87
Transportation: references to public transportation— like the MBTA, buses, and trains. This can include discussions about: the quality, affordability, accessibility, and safety of transportation	49

Table 4: Description of Likert-scale questions asked during the user study. Participants were asked to rate how much they agreed with several statements (1 = strongly disagree and 7 = strongly agree).

Metric Name	Description
Productivity	Level of agreement towards the statement: "I became productive quickly while using SenseMate." [5]
Working speed	Level of agreement towards the statement: "I was able to complete the task quickly using SenseMate." [5]
Ease of use	Level of agreement towards the statement: "I thought the SenseMate platform was easy to use." [5]
Well integrated	Level of agreement towards the statement: "I found the various functions in SenseMate were well integrated." [5]
Effective information	Level of agreement towards the statement: "The information provided in SenseMate was effective in helping me complete my work." [5]
Confidence	Level of agreement towards the statement: "I'm confident about the themes I selected for the stories"
Want to use in future	Level of agreement towards the statement: "I would like to use SenseMate for similar labeling tasks." [5]
Would recommend	Level of agreement towards the statement: "I would recommend SenseMate to people who do similar labeling tasks." [5]
Understanding	Level of agreement towards the statement: "I understand why SenseMate gave particular theme recommendations." ["Rec Only" and "Rec and Rationale" conditions]
Trust	Level of agreement towards the statement: "I trust the theme recommendations made by SenseMate." ["Rec Only" and "Rec and Rationale" conditions]
Agency	Level of agreement towards the statement: "I was able to shape SenseMate's theme recommendations with my feedback and actions." ["Rec Only" and "Rec and Rationale" conditions]
Perceived accuracy	Level of agreement towards the statements: "I thought SenseMate's theme recommendations were accurate overall." ["Rec Only" and "Rec and Rationale" conditions] + "I thought SenseMate's reasons for theme recommendations were accurate overall." ["Rec and Rationale" conditions]

Table 5: Mean and standard deviation of coding accuracy, precision, recall, and F-score for each experiment condition.

Experiment Condition	Accuracy	Precision	Recall	F-Score
Control	0.87 (0.14)	0.83 (0.28)	0.76 (0.29)	0.76 (0.25)
Rec Only	0.91 (0.11)	0.86 (0.21)	0.86 (0.23)	0.85 (0.20)
Rec and Rationale	0.91 (0.10)	0.89 (0.19)	0.88 (0.20)	0.86 (0.17)