Co-DETECT: Collaborative Discovery of Edge Cases in Text Classification

Chenfei Xiong[Z*] **Jingwei Ni**[E Z*] **Yu Fan**[E*] **Vilém Zouhar**[E]
Donya Rooein[B E] **Lorena Calvo-Bartolomé**[C] **Alexander Hoyle**[E]
Zhijing Jin[T] **Mrinmaya Sachan**[E] **Markus Leippold**[Z]
Dirk Hovy[B] **Mennatallah El-Assady**[E] **Elliott Ash**[E]
[E]ETH Zürich [Z]University of Zürich
[B]Bocconi University [T]University of Toronto [C]Universidad Carlos III
{jingni, yufan, ashe}@ethz.ch

Abstract

We introduce Co-DETECT (Collaborative Discovery of Edge cases in TExt ClassificaTion), a novel mixed-initiative annotation framework that integrates human expertise with automatic annotation guided by large language models (LLMs). Co-DETECT starts with an initial, sketch-level codebook and dataset provided by a domain expert, then leverages the LLM to annotate the data and identify edge cases that are not well described by the initial codebook. Specifically, Co-DETECT flags challenging examples, induces high-level, generalizable descriptions of edge cases, and assists user in incorporating edge case handling rules to improve the codebook. This iterative process enables more effective handling of nuanced phenomena through compact, generalizable annotation rules. Extensive user study, qualitative, and quantitative analyses prove the effectiveness of Co-DETECT.[1]

1 Introduction

Social scientists often find themselves in situations requiring data annotation based on human judgment and specific expertise (Wilkerson and Casas, 2017; Kennedy et al., 2018; Demszky et al., 2020; Drápal et al., 2023). For example, a political scientist studying hate speech on social media may need to develop a codebook that clearly defines what constitutes hate speech in the social media scenario with illustrative positive and negative examples. After developing such a codebook, the researcher may need to recruit annotators possessing sufficient domain expertise to appropriately apply the established guidelines to actual social media content.

However, both tasks—codebook development and data annotation—involve significant human effort.

Firstly, it is challenging even for domain experts to develop reliable codebook (Halterman and Keith, 2025). To start with, ambiguity and subjectivity are inherent obstacles, as interpreting complex human behaviors and communications often yields multiple valid perspectives, leading to edge cases that need specific rules to handle (Fornaciari et al., 2021; Fuchs et al., 2021; Fleisig et al., 2023; Fan et al., 2025b). It is usually infeasible for an expert to manually examine the entire target corpus in order to identify the many edge cases that arise. In addition, codebook developers may introduce biases or subjective interpretations shaped by their domain knowledge and socio-demographic background, which can limit the effectiveness of the codebook in handling difficult or ambiguous cases. Even when an edge case is identified, the expert may struggle to explicitly articulate the subtleties and intuitions that guide their judgments, a phenomenon commonly known as *Polanyi's paradox*[2] (Autor, 2014; Fügener et al., 2022). As a result, although domain experts may possess rich implicit understandings of certain social phenomena, they face significant challenges in capturing and codifying this tacit knowledge within verbal, structured annotation frameworks.

Secondly, large-scale human annotation is often infeasible in many use cases (e.g., Xie and Zhang, 2024). Employing qualified annotators is costly, especially when the task requires domain-specific expertise. Furthermore, domain-specific data frequently involve nuanced and complex contexts (e.g., Ziems et al., 2024; Fan et al., 2025a; Zhao et al., 2025), which demand greater cognitive effort and longer annotation times to ensure high quality. As a result, large-scale human annotation is often prohibitively expensive in terms of both cost and time. To address this challenge,

[*] Equal contribution.

[1] Frontend: codetect.vercel.app; code and demonstration video: github.com/EdisonNi-hku/Co-DETECT

[2] In everyday language, this phenomenon is often summarized by the phrase: "We can know more than we can tell."

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 354–364
November 4-9, 2025 ©2025 Association for Computational Linguistics

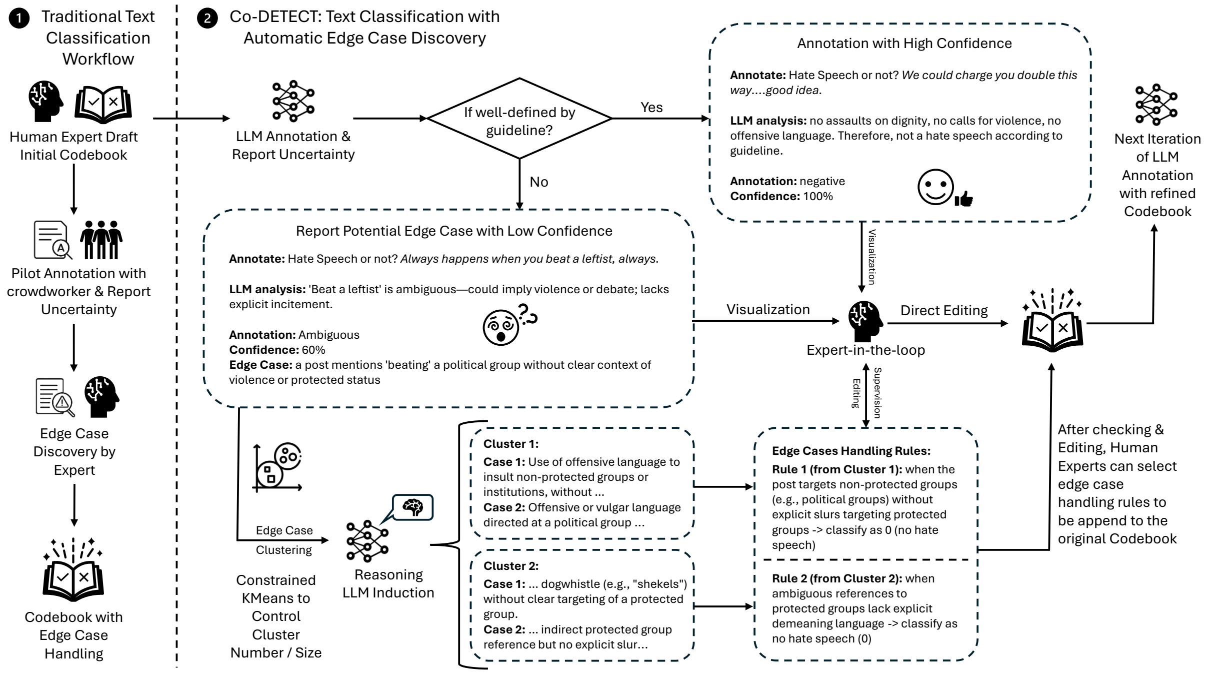

1 Traditional Text Classification Workflow
Human Expert Draft Initial Codebook
Pilot Annotation with crowdworker & Report Uncertainty
Edge Case Discovery by Expert
Codebook with Edge Case Handling
2 Co-DETECT: Text Classification with Automatic Edge Case Discovery
LLM Annotation & Report Uncertainty
If well-defined by guideline?
Yes
No
Annotation with High Confidence
Annotate: Hate Speech or not? We could charge you double this way....good idea.
LLM analysis: no assaults on dignity, no calls for violence, no offensive language. Therefore, not a hate speech according to guideline.
Annotation: negative
Confidence: 100%
Next Iteration of LLM Annotation with refined Codebook
Report Potential Edge Case with Low Confidence
Annotate: Hate Speech or not? Always happens when you beat a leftist, always.
LLM analysis: 'Beat a leftist' is ambiguous—could imply violence or debate; lacks explicit incitement.
Annotation: Ambiguous
Confidence: 60%
Edge Case: a post mentions 'beating' a political group without clear context of violence or protected status
Visualization
Visualization
Direct Editing
Expert-in-the-loop
Supervision
Editing
Edge Case Clustering
Constrained KMeans to Control Cluster Number / Size
Reasoning LLM Induction
Cluster 1:
Case 1: Use of offensive language to insult non-protected groups or institutions, without ...
Case 2: Offensive or vulgar language directed at a political group ...
Cluster 2:
Case 1: ... dogwhistle (e.g., "shekels") without clear targeting of a protected group.
Case 2: ... indirect protected group reference but no explicit slur...
Edge Cases Handling Rules:
Rule 1 (from Cluster 1): when the post targets non-protected groups (e.g., political groups) without explicit slurs targeting protected groups -> classify as 0 (no hate speech)
Rule 2 (from Cluster 2): when ambiguous references to protected groups lack explicit demeaning language -> classify as no hate speech (0)
After checking & Editing, Human Experts can select edge case handling rules to be append to the original Codebook

Figure 1: ❶ traditional workflow of text annotation, where experts rely on their own or crowdworkers to identify edge cases and update codebook based on the discovered prevalent edge cases. ❷ Co-DETECT mixed-initiative workflow of edge case discovery, where LLMs propose prevalent and representative edge cases and the visual interface assists human expert to verify the proposed edge cases.

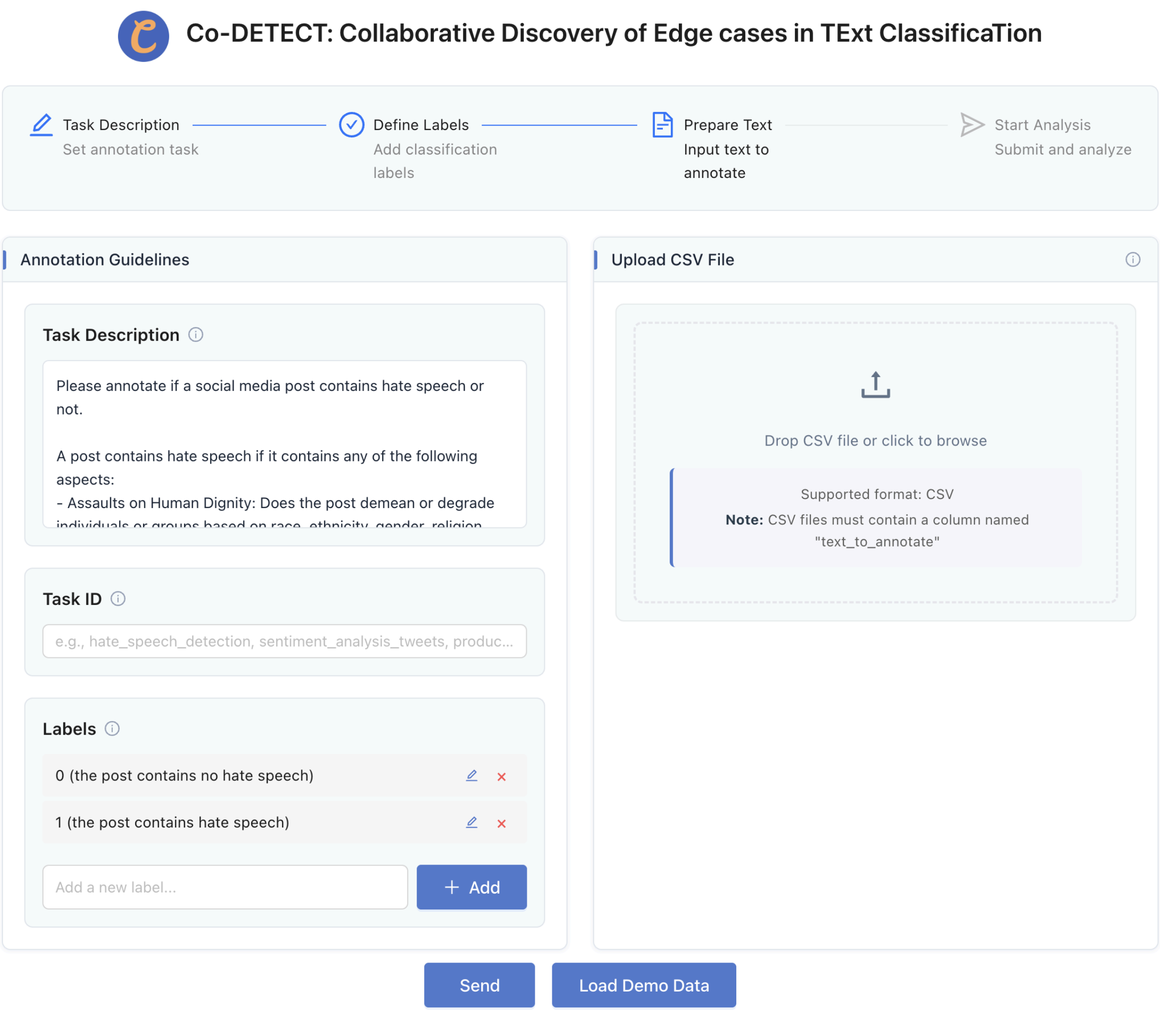
recent research explores leveraging LLMs for automatic annotation (Gilardi et al., 2023; Pangakis et al., 2023; Ding et al., 2023; He et al., 2024b,a; Dunivin, 2024; Törnberg, 2024). These approaches typically assume the availability of well-developed codebooks for LLM prompting (Halterman and Keith, 2025; Xiao et al., 2023). However, how codebook development and the annotation process interact—and, crucially, how expert knowledge shapes this interaction—remains underexplored. In practice, domain experts often iterate between the codebook and annotation results, updating the codebook based on insights gathered during annotation (e.g., Kirsten et al., 2025). This iterative process is essential for uncovering edge cases that a previous codebook may have overlooked, enabling experts to revise the codebook accordingly.

To address these gaps, we introduce Co-DETECT, a mixed-initiative text analysis tool designed to support domain experts in discovering and managing edge cases (Figure 1). Co-DETECT takes annotation guidelines (codebooks) and a target annotation corpus as input. It identifies edge cases—data points poorly defined by the provided codebook, clusters them strategically, and proposes aggregated edge categories with representative examples. The user then evaluates these suggestions, determines their validity, and updates the codebook accordingly with clear handling rules. Finally, annotation can proceed in a new iteration using the revised codebook.

This expert-in-the-loop approach enables humans and AI to complement one another by leveraging their respective strengths in collaboration. Prior work shows that humans often struggle to identify edge cases due to limited *metaknowledge*—the ability to assess the scope and boundaries of their own knowledge (Fügener et al., 2022; Evans and Foster, 2011). By contrast, AI systems are better at uncovering edge cases, though their ability to address them remains limited (Ni et al., 2025a). Accordingly, incorporating AI into an expert-in-the-loop framework is advisable: AI can surface edge cases and enrich the codebook, while human oversight ensures appropriate interpretation and handling. In summary, our contributions are:

1. We develop Co-DETECT for domain experts that iteratively updates the codebook under human supervision. Consists of an LLM-based induction algorithm suggesting representative edge cases and a user-friendly interface enabling domain experts to more effectively handle edge cases.

2. We conduct user studies with domain experts from diverse backgrounds, shedding light on the effectiveness of Co-DETECT and directions for



Please annotate if a social media post contains hate speech or not.

A post contains hate speech if it contains any of the following aspects:
- Assaults on Human Dignity: Does the post demean or degrade individuals or groups based on race, ethnicity, gender, religion

e.g., hate_speech_detection, sentiment_analysis_tweets, produc...

0 (the post contains no hate speech)

1 (the post contains hate speech)

Supported format: CSV
Note: CSV files must contain a column named "text_to_annotate"

Figure 2: User Interface – Input Page

future work.

2 Frontend and User Work Flow

In this section, we provide a detailed introduction to the user workflow (illustrated in fig. 1), including a preparation stage section 2.1 and a dashboard analysis stage section 2.2.

2.1 Preparation Stage

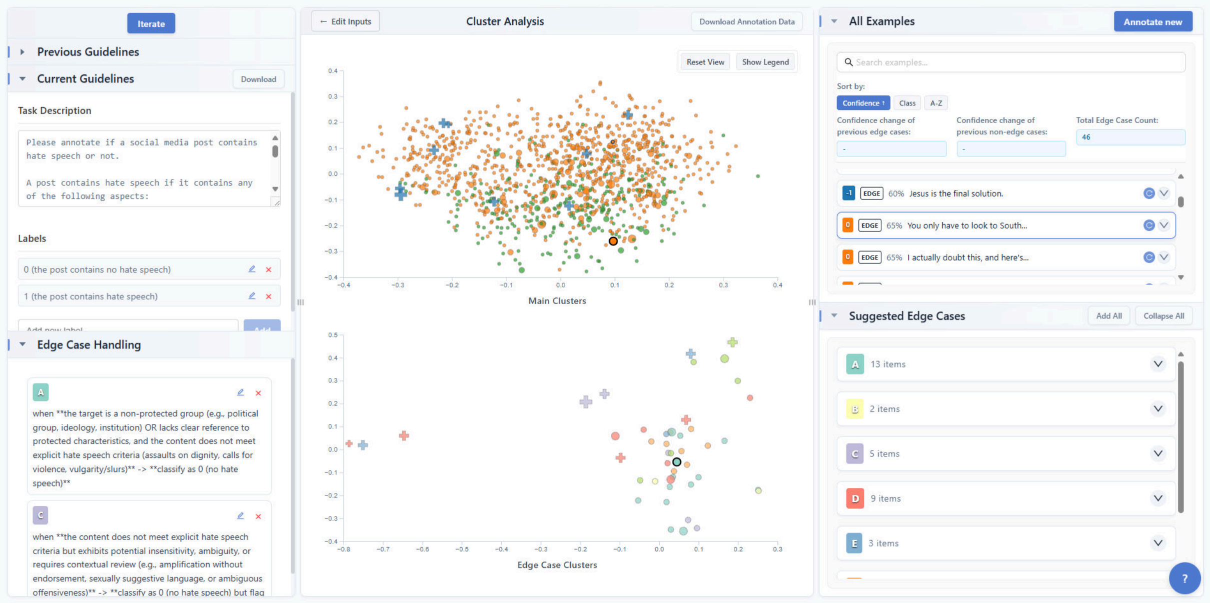
Onboarding. To ensure a smooth onboarding experience, the first launch of Co-DETECT triggers an intro.js tour that guides users through the input and dashboard pages. Furthermore, users can also click "Load Demo Data" to explore a sample usecase in annotating hate speech from social media.

Input Page. After familiarized with Co-DETECT, the user can start from the input page (Figure 2), where they need to provide a initial draft of the codebook, including (1) a task definition (e.g., a post contains hate speech if it contains assaults on human dignity, calls for violence, or vulgarity.); (2) classification labels (e.g., 1 for hate speech and 0 for no hate speech); and (3) a task ID which is useful for saving all annotation outcomes, edge cases, and codebooks to the backend. Besides the codebook, the user also need to provide a csv file containing 500 to 1000 target texts to be annotated. We suggest this number of texts to ensure the representativeness of edge cases with reasonable budget. With the input prepared, the user can click "send" to pass the inputs to the LLM analysts.

2.2 Dashboard Analysis Stage

Current Guidelines. At the upper left-hand side of the dashboard, the user-provided codebook is displayed, allowing users to optimize their annotation task descriptions and manage labels.

Exploring Annotation Results. At the upper middle of the dashboard, we provide a scatter plot



Please annotate if a social media post contains hate speech or not.

A post contains hate speech if it contains any of the following aspects:

0 (the post contains no hate speech)

1 (the post contains hate speech)

when **the target is a non-protected group (e.g., political group, ideology, institution) OR lacks clear reference to protected characteristics, and the content does not meet explicit hate speech criteria (assaults on dignity, calls for violence, vulgarity/slurs)** -> **classify as 0 (no hate speech)**

when **the content does not meet explicit hate speech criteria but exhibits potential insensitivity, ambiguity, or requires contextual review (e.g., amplification without endorsement, sexually suggestive language, or ambiguous offensiveness)** -> **classify as 0 (no hate speech) but flag

Confidence change of previous edge cases: -

Confidence change of previous non-edge cases: -

Total Edge Case Count: 46

-1 EDGE 60% Jesus is the final solution.

0 EDGE 65% You only have to look to South...

0 EDGE 65% I actually doubt this, and here's...

A 13 items

B 2 items

C 5 items

D 9 items

E 3 items

Figure 3: User Interface – Analysis Dashboard

showing all annotated text samples. Each point represents an example, clustered according to embedding similarity. Different colors indicate different annotation labels, and point size denotes annotation uncertainty—how likely the samples belong certain edge cases. Clicking on points in the scatter plot, the annotation details of the corresponding items will pop out on the upper right "All Examples" list, including LLM analysis, annotation confidence, and edge case suggestions.

Analyzing Edge Cases. Either clicking large points (uncertain annotations) in the upper scatter plot or "Edge" items in the upper right list will connect the user to the lower middle scatter plot and lower right list. The lower middle plot presents the clusters of potential edge cases that may require attention. These samples are automatically identified by the system as challenging or requiring more precise annotation guidelines. The panel named "Suggested Edge Cases" on the lower right outlines high-level descriptions of each edge case cluster, and examples in each cluster.

Edge Case Handling and Iterative Optimization. Once users find any cluster in "Suggested Edge Cases" reasonable, they can add the corresponding edge case handling rule to the lower left panel "Edge Case Handling". For example, clusters A and C are added in Figure 3. Users can also edit the edge case handling rules freely. Once they are satisfied with the added rules, they can click "Iterate" on the top left to re-annotate the corpus with the codebook augmented with "Edge Case Handling". Codebook of previous iterations will be saved in the top left panel—"Previous Guidelines".

3 Backend Algorithm for Edge Case Discovery

Problem Formulation. As illustrated in fig. 1, either traditional text annotation or Co-DETECT requires a target corpus and task definition (i.e., initial codebook) as inputs. At this stage, the user may lack insights about the corpus, including limited edge case handlings in the codebook. Co-DETECT aims at discovering edge cases that are ambiguously defined by the codebook. The edge cases proposed by Co-DETECT should be:

- **Descriptive**: capture the core features of exact edge case samples and the reason why they are ambiguous.
- **High-Level**: while being descriptive, the edge case descriptions should not over specifically describe certain samples. Only then can they be added to the codebook and generalized to unseen data points.

To fulfill these desiderata, the edge case discovery algorithm of Co-DETECT details as follows:

Step 1: Item-Level Edge Cases. We start from using a non-reasoning LLM[3] (e.g., in our case GPT-4.1) to quickly annotate all data points. We prompt

[3] We use reasoning LLMs referring to LLMs with test-time long CoT reasoning (e.g., DeepSeek-R1 (DeepSeek-AI, 2025) and OpenAI O3 (OpenAI, 2025b)); and non-reasoning LLMs referring to those answer immediately (e.g., GPT-4.1 (OpenAI, 2025a)).

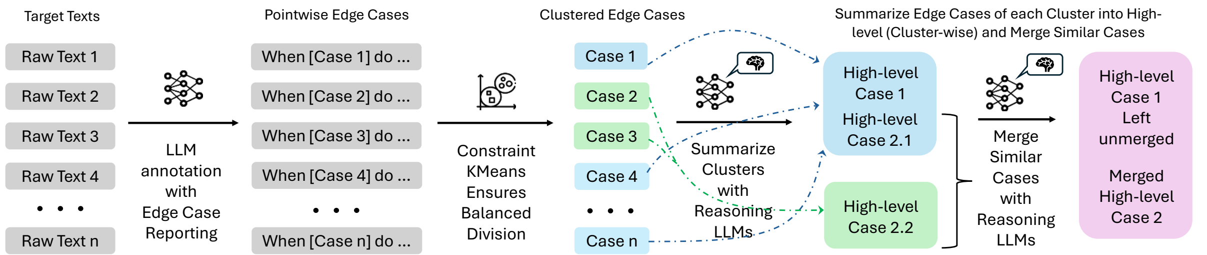


Figure 4: Co-DETECT's backend algorithm for automatically discover representative edge cases. Firstly, an LLM annotator report pointwise edge cases. Secondly, a reasoning LLM aggregates item-wise edge cases into more representative high-level edge cases, with the help of clustering algorithms.

the LLM to (1) annotate, (2) provide a confidence score reflecting the annotation correctness (following Tian et al. (2023b)), and (3) explain why the case is an edge case if the confidence is low. The explanations are in a form of edge case handling rules like "when `[Case Description]`, do `[Action]`". `[Case Description]` describes why the sample is ambiguously defined, so it might be too specific and low-level. `[Action]` is an LLM suggested handling for the edge case.

Step 2: Cluster-Level Edge Cases. To avoid over specific `[Case Description]` that fails to generalize to other samples, we need to aggregate item-level edge cases with similar ambiguity and describe them in a higher level. This is a challenging task requiring (1) covering all item-level edge cases; and (2) strategically finding logical similarities between reasons for ambiguity. Therefore, we employ a SOTA reasoning LLM—DeepSeek-R1 (DeepSeek-AI, 2025) to cluster `[Case Description]` and generate high-level edge cases and handling rules. Specifically, we extract the item-level `[Case Description]`, embed them with semantic embedding models[4], and cluster them with constrained KMeans (Levy-Kramer, 2018). Each cluster of `[Case Description]` and corresponding `[Action]` are fed to DeepSeek-R1 to generate Cluster-wise Edge Cases. Constrained KMeans ensures that all clusters have 10 to 20 samples, so that the input (i.e., each cluster) to DeepSeek-R1 will not be too large or small, as we empirically find that large clusters (>20) may increase the reasoning burden and lead to hallucination, while small clusters (<5) may generate over-specific edge cases. Cluster-level edge cases are also companied with `[Action]` to handling them.

Step 3: Merge Cluster-Level Edge Cases. Since each cluster may have overlapped edge cases, we finally call DeepSeek-R1 again to merge cluster-level edge cases and their handling rules. This also ensures that similar edge cases are not handled with different rules.

4 User Study

To evaluate Co-DETECT's effectiveness and collect feedback for further improvement, we conduct a systematic user study with domain experts. Prior to the study, participants were asked to prepare a text annotation task and an accompanying corpus from their own research domains (i.e., areas where they possess domain expertise). At the beginning of the user study, participants first complete a pre-interaction survey gathering their background information. Then, they interact with the system for approximately 45 minutes, including the LLM response time. Finally, they complete a post-interaction survey, collecting comprehensive user feedback. We recruit 10 users in total. 5 of them are not involved in the design of Co-DETECT. The remaining five are co-authors of the paper but were not familiarized with the workflow before the user study.

4.1 Pre-Interaction Survey Takeaways

We summarize the key findings from the pre-interaction survey below. For the full survey form, please refer to Appendix A.

Diverse Experience and Background of the Participants. Our participants have a broad academic background in social science, computational linguistics, and interdisciplinary training. They also exhibit diverse experience in both social science qualitative coding and LLM-assisted annotation, from no experience to expert level.

Heavy Reliance on Manual Effort for Edge Case Discovery. Concerning common workflows for identifying edge cases, 80% of participants manually review subsets of data to detect potential edge

[4] In our project, we use OpenAI text-embedding-3-large (OpenAI, 2024) for convenience.

cases. Some also report employing other human (e.g., crowdworkers) or AI annotators to do pilot annotation and flag potential edge cases.

Moderate Prior Knowledge of Edge Cases. 70% of participants report knowing certain edge cases in their intended datasets. Therefore, it would be valuable to check if Co-DETECT can mine already-known edge cases or discover new edge cases.

4.2 Post-Interaction Survey Takeaways

Below, we highlight the main insights from the post-interaction survey, which center on four key aspects of user experience with Co-DETECT: ease of use, interpretability of visualizations, validity of edge cases, and overall satisfaction and feedback. For the full survey form, please refer to Appendix B.

The Majority Finds Co-DETECT Workflow Easy to Follow. The survey results reveal generally positive feedback on interface ease of use and task clarity, with most participants (80%) finding navigation intuitive and interaction straightforward. Some requested additional visualizations (e.g., density distribution of the confidence scores) or export features for enhanced usability.

Co-DETECT Can Identify Relevant Edge Cases. 60% of participants approved Co-DETECT's ability to clearly identify relevant edge cases. For example, one participant reported that the edge case handling rules suggested are "clear, realistic, and concise", and directly inform their acceptance or rejection decisions, pointing out ways to improve the precision and coverage of these suggestions. 90% of participants report that Co-DETECT may help discover new edge cases beyond their prior knowledge of the dataset.

Useful Iterative Workflow and Overall Satisfaction. 80% of participants find the iterative feature of Co-DETECT useful for refining their annotation guidelines. All participants were satisfied with the Co-DETECT system's support in generating annotation guidelines and identifying new edge cases.

Constructive Critiques. Besides the generally positive feedback on user experience of Co-DETECT, the post-interaction survey also gives us valuable critiques, highlighting areas for future improvement of Co-DETECT. 40% of participants express concern that Co-DETECT may overlook potential edge cases although the identified edge cases seem reasonable. For instance, one participant also indicated that there was room to enhance the coherence and descriptive clarity of the edge cases.

Dataset	1st Iter.	2nd Iter.
GabHateCorpus	0.2144	**0.2523**
GoEmotions-Positive	0.0300	**0.3297**
GoEmotions-Negative	0.2823	**0.3046**

Table 1: Classification F1 Scores using the original codebook (from the 1st iteration) and the improved codebook after one Co-DETECT iteration (from the 2nd iteration).

4.3 Quantitative Human Evaluation on Edge Case Validity

We further conducted a quantitative human evaluation with three participants[5]. Each participant was asked to randomly select 1 to 2 samples from each edge case cluster and manually assess how many were accurately captured by the Co-DETECT-suggested edge case descriptions. Among 41 randomly selected samples, 33 (**80.5%**) were reported as well-described by the suggested edge case descriptions. The edge case descriptions are also found to be sufficiently high-level to cover more than one samples. We further find that it often takes less than 5 seconds for an expert to identify if a sample is covered by an edge case description or not, indicating that Co-DETECT may not impose a heavy cognitive load on users when supervising suggested edge case clusters.

5 Can Improved Codebook Benefit Automatic Annotation?

The main goal of Co-DETECT is to help experts improve codebooks, but does a better codebook actually enhance automatic annotation? To investigate this, we provide GPT-4.1 with codebooks before and after Co-DETECT enhancement and compare its classification F1, varying only the codebook. We strategically pick a hate speech detection dataset—GabHateCorpus (Kennedy et al., 2021) and an emotion classification—GoEmotions (Demszky et al., 2020; Positive / Negative Emotion Detection) for this evaluation, because these tasks are highly subjective (Davani et al., 2022; Ni et al., 2025a) and thus challenging for codebook drafting. They are therefore challenging for advanced LLMs like GPT-4.1 that are smart enough to understand

[5] Due to the original user study is already time-intensive, participation in the quantitative evaluation was optional.

the nuanced perturbations within the codebook. It is also challenging for human experts to manually improve codebook as it is hard for individuals to capture various subjectivity.

The results are exhibited in table 1, where we observe an increase in F1 scores across different datasets. Notably, Co-DETECT only augments codebook by appending edge case handling rules. The initial codebook for GoEmotion-Positive has very low F1 score due to an extremely low recall—the model rarely predicts positive emotions that are not explicitly stated by the raw codebook. Thereby, we showcase that Co-DETECT can improve classification outcomes with improved codebook, even for subjective tasks that are both challenging for LLMs and individual experts.

6 Related Work

Annotation with LLM Assistance. Both NLP (Kim et al., 2024; Ni et al., 2024, 2025b) and HCI (He et al., 2024b; Törnberg, 2023) community have widely explored human-AI collaborations for text classification. Tian et al. (2023a) find that the verbalized confidence of LLMs indicates classification quality, and annotations where LLM reports high annotation confidence may outperform human annotator (Ni et al., 2024, 2025b; Törnberg, 2023). We follow this stream of work to calibrate the quality of LLM annotation using verbalized confidence scores. In Human-AI interaction, Wang et al. (2024) and Kim et al. (2024) develop mixed-initiative tools to enhance automatic annotation with minimal human supervision. However, these methods focus on annotation accuracy and assume a predefined codebook. In contrast, our work targets efficiency in codebook development and edge-case discovery, critical steps especially in the initial stages of text classification (Törnberg, 2024).

Goal-Driven Clustering in NLP. One critical step of our edge case discovery algorithm is to cluster low-level specific edge cases into high-level representative edge cases. Such goal-driven clustering (Wang et al., 2023) is essentially relevant to many NLP sub-fields, such as topic modeling (Pham et al., 2024), inductive reasoning (Lam et al., 2024), corpus comparison (Zhong et al., 2023), information retrieval (Ni et al., 2025b) etc. In such tasks, LLM plays an important role in understanding users' goal and steering / interpreting the clustering accordingly (Zhang et al., 2023; Viswanathan et al., 2024; Movva et al., 2025). Our work contributes to adapting goal-driven clustering to edge case discovery, leveraging analytical skills of reasoning models (DeepSeek-AI, 2025).

7 Conclusion

We developed Co-DETECT to systematically identifies descriptive and generalizable edge cases and collaboratively improve codebook with human expert. To achieve this, Co-DETECT induces representative edge cases leveraging multi-step clustering and reasoning LLMs. Then the user can supervise the quality of suggested edge cases and decide whether to include them into the codebook or not. Comprehensive user study, and other qualitative and quanititative evaluations prove the effectiveness of Co-DETECT.

8 Limitations and Future Work

While our user study and both qualitative and quantitative analyses demonstrate the effectiveness of Co-DETECT, it also has limitations that we plan to address in future work.

The primary limitation lies in the overreliance on LLM-reported confidence levels, which may introduce significant biases. For example, models may depend on superficial features or spurious correlations learned during pretraining, resulting in unfaithfully high confidence scores and thus the neglect of important edge cases. Moreover, human annotators may rely too heavily on the model's suggestions, potentially overlooking relevant edge cases or alternative interpretations.

To address this, we plan to incorporate sparse autoencoders in future work to provide interpretable features for edge case detection. This will allow users to assess whether a detection is driven by spurious features or by genuine factors that warrant further refinement and specification in the codebook.

Ethics Statement

This research involved voluntary participation in user studies, during which participants provided professional background information and evaluated interface functionality for annotation and edge-case identification tasks. Participants were clearly informed about the study objectives, tasks, and their right to withdraw at any time. Collected data were securely stored, anonymized, and analyzed collectively to ensure confidentiality and privacy. The study posed minimal risks to participants, aligned with standard professional activities, and adhered closely to ethical guidelines for human-centered research.

Broader Impact Statement

Our system pipeline emphasizes an interactive and iterative approach designed to enhance annotation accuracy and generalization through the systematic management of challenging edge cases. This approach is based on the assumption that improved confidence metrics in a model correlate with enhanced annotation performance. Nevertheless, analogous to the *Clever Hans* phenomenon—where an intelligent system identifies unintended cues instead of genuinely learning underlying knowledge—it is crucial to critically assess the robustness of this pipeline against potential biases and unintended shortcuts that may result from repeated feedback loops and rule-induction processes.

One potential concern involves deriving edge-case rules primarily from model confidence metrics and automatically suggested edge-case instances. If the model's selection and clustering of these edge cases rely predominantly upon internally generated confidence measures, there is a risk that inductively derived rules may reinforce model-specific biases rather than reflect genuinely generalizable conceptual regularities. For example, the model might inadvertently identify clusters based on spurious correlations between input texts and target labels instead of addressing genuine annotation challenges.

Therefore, such risks necessitate increased caution. Systems optimized exclusively within internal frameworks may achieve superficially impressive performance improvements without truly acquiring underlying domain expertise. Consequently, we emphasize the critical importance of human domain expertise. Users should ensure that only rules verified by domain experts are included when updating the codebook.

Acknowledgment

We would like to express our gratitude to *Cong Jiang, Felix Riechmann, Yang Tian, Tingyu Yu,* and *Darya Zare* for their generous support throughout this study.

References

David Autor. 2014. Polanyi's paradox and the shape of employment growth. Technical report, National Bureau of Economic Research.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Jakub Drápal, Hannes Westermann, Jaromir Savelka, et al. 2023. Using large language models to support thematic analysis in empirical legal studies. In *JURIX*, pages 197–206.

Zackary Okun Dunivin. 2024. Scalable qualitative coding with llms: Chain-of-thought reasoning matches human performance in some hermeneutic tasks. *Preprint*, arXiv:2401.15170.

James A Evans and Jacob G Foster. 2011. Metaknowledge. *Science*, 331(6018):721–725.

Yu Fan, Jingwei Ni, Jakob Merane, Etienne Salimbeni, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Florian Geering, Oliver Dreyer, Daniel Brunner, Markus Leippold, Mrinmaya Sachan, Alexander Stremitzer, Christoph Engel, Elliott Ash, and Joel Niklaus. 2025a. Lexam: Benchmarking legal reasoning on 340 law exams. *Preprint*, arXiv:2505.12864.

Yu Fan, Yang Tian, Shauli Ravfogel, Mrinmaya Sachan, Elliott Ash, and Alexander Hoyle. 2025b. The medium is not the message: Deconfounding text embeddings via linear concept erasure. *Preprint*, arXiv:2507.01234.

Eve Fleisig, Rediet Abebe, and Dan Klein. 2023. When the majority is wrong: Modeling annotator disagreement for subjective tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6715–6726, Singapore. Association for Computational Linguistics.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, pages 2591–2597, Online. Association for Computational Linguistics.

Lukas M Fuchs, Yu Fan, and Christian von Scheve. 2021. Value differences between refugees and german citizens: insights from a representative survey. *International Migration*, 59(5):59–81.

Andreas Fügener, Jörn Grahl, Alok Gupta, and Wolfgang Ketter. 2022. Cognitive challenges in human–artificial intelligence collaboration: Investigating the path toward productive delegation. *Information Systems Research*, 33(2):678–696.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Andrew Halterman and Katherine A. Keith. 2025. Codebook llms: Evaluating llms as measurement tools for political science concepts. *Preprint*, arXiv:2407.10747.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024a. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 165–190, Mexico City, Mexico. Association for Computational Linguistics.

Zeyu He, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Shaurya Rohatgi, and Ting-Hao Kenneth Huang. 2024b. If in a crowdsourced data annotation pipeline, a gpt-4. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Brendan Kennedy, Mohammad Atari, Aida Mostafazadeh Davani, Leigh Yeh, Ali Omrani, Yehsong Kim, Kris Coombs, Shreya Havaldar, Gwenyth Portillo-Wightman, Elaine Gonzalez, et al. 2018. The gab hate corpus: A collection of 27k posts annotated for hate speech. *PsyArXiv. July*, 18.

Brendan Kennedy, Mohammad Atari, Aida Mostafazadeh Davani, Yehsong Kim, Kris Coombs, Gwenyth Portillo-Wightman, Shreya Havaldar, Elaine Gonzalez, Joseph Hoover, Aida Azatian, Gabriel Cardenas, Alyzeh Hussain, Austin Lara, Adam Omary, Christina Park, Xin Wang, Clarisa Wijaya, Yong Zhang, Beth Meyerowitz, and Morteza Dehghani. 2021. The Gab Hate Corpus: a collection of 27k posts annotated for hate speech. OSF Preprint. 27,665 posts from Gab annotated by multiple annotators; CC-BY 4.0.

Hannah Kim, Kushan Mitra, Rafael Li Chen, Sajjadur Rahman, and Dan Zhang. 2024. MEGAnno+: A human-LLM collaborative annotation system. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–176, St. Julians, Malta. Association for Computational Linguistics.

Elisabeth Kirsten, Annalina Buckmann, Leona Lassak, Nele Borgert, Abraham Mhaidli, and Steffen Becker. 2025. From assistance to autonomy – a researcher study on the potential of ai support for qualitative data analysis. *Preprint*, arXiv:2501.19275.

Michelle S. Lam, Janice Teoh, James A. Landay, Jeffrey Heer, and Michael S. Bernstein. 2024. Concept induction: Analyzing unstructured text with high-level concepts using lloom. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, page 1–28. ACM.

Josh Levy-Kramer. 2018. k-means-constrained.

Rajiv Movva, Kenny Peng, Nikhil Garg, Jon Kleinberg, and Emma Pierson. 2025. Sparse autoencoders for hypothesis generation. *Preprint*, arXiv:2502.04382.

Jingwei Ni, Yu Fan, Vilém Zouhar, Donya Rooein, Alexander Hoyle, Mrinmaya Sachan, Markus Leippold, Dirk Hovy, and Elliott Ash. 2025a. Can large language models capture human annotator disagreements? *Preprint*, arXiv:2506.19467.

Jingwei Ni, Tobias Schimanski, Meihong Lin, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2025b. DIRAS: Efficient LLM annotation of document relevance for retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5238–5258, Albuquerque, New Mexico. Association for Computational Linguistics.

Jingwei Ni, Minjing Shi, Dominik Stammbach, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2024. AFaCTA: Assisting the annotation of factual claim detection with reliable LLM annotators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1890–1912, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2024. New embedding models and api updates: text-embedding-3-large. `https://openai.com/index/new-embedding-models-and-api-updates/`. Introduced text-embedding-3-large (3072-dim), offers strongest performance on MIRACL (54.9

OpenAI. 2025a. Introducing gpt-4.1, gpt-4.1 mini & nano. https://openai.com/index/gpt-4-1/. Released April 14, 2025; includes full, mini, and nano variants (1M token context window; optimized for coding and instruction-following).

OpenAI. 2025b. Introducing openai o3 and o4-mini. O3 is a reasoning-focused generative model with advanced capabilities in coding, math, and visual perception; system card provides detailed benchmarks and safety evaluations.

Nicholas Pangakis, Samuel Wolken, and Neil Fasching. 2023. Automated annotation with generative ai requires validation. *ArXiv*, abs/2306.00176.

Chau Minh Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. 2024. TopicGPT: A prompt-based topic modeling framework. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2956–2984, Mexico City, Mexico. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023a. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023b. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *Preprint*, arXiv:2305.14975.

Petter Törnberg. 2024. Best practices for text annotation with large language models. *ArXiv*, abs/2402.05129.

Petter Törnberg. 2023. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *Preprint*, arXiv:2304.06588.

Vijay Viswanathan, Kiril Gashteovski, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2024. Large language models enable few-shot clustering. *Transactions of the Association for Computational Linguistics*, 12:321–333.

Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. 2024. Human-llm collaborative annotation through effective verification of llm labels. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-driven explainable clustering via language descriptions. *Preprint*, arXiv:2305.13749.

John Wilkerson and Andreu Casas. 2017. Large-scale computerized text analysis in political science: Opportunities and challenges. *Annual Review of Political Science*, 20(1):529–544.

Ziang Xiao, Xingdi Yuan, Q Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. 2023. Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for deductive coding. In *Companion proceedings of the 28th international conference on intelligent user interfaces*, pages 75–78.

Tian Xie and Xueru Zhang. 2024. Automating data annotation under strategic human agents: Risks and potential solutions. *Preprint*, arXiv:2405.08027.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. ClusterLLM: Large language models as a guide for text clustering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13903–13920, Singapore. Association for Computational Linguistics.

Chengshuai Zhao, Zhen Tan, Chau-Wai Wong, Xinyan Zhao, Tianlong Chen, and Huan Liu. 2025. SCALE: Towards collaborative content analysis in social science with large language model agents and human intervention. *Preprint*, arXiv:2502.10937.

Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan Klein, and Jacob Steinhardt. 2023. Goal driven discovery of distributional differences via language descriptions. In *Advances in Neural Information Processing Systems*, volume 36, pages 40204–40237. Curran Associates, Inc.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can large language models transform computational social science? *Computational Linguistics*, 50(1):237–291.

A Pre-Interaction Survey Questions

In the pre-interaction survey, we included following questions:

(1) General User Background

- Please briefly describe your professional/academic background and current research areas.
- What is your previous experience with data annotation and qualitative coding? *(None, Beginner, Intermediate, Advanced, Expert)*
- Have you used annotation support or guideline-generation tools previously? *(Yes, No)*

(2) Expectation

- What is your normal workflow of identifying edge cases in text annotation?
- In the dataset that you plan to analyze with AutoDETECT, did you already know any edge cases? *(Yes, No)*

B Post-Interaction Survey Questions

In the post-interaction survey, we included following questions:

(1) Task Completeness, Clarity, and Ease of Use

- Were you clearly able to understand the steps for configuring an annotation task using the interface? *(Five-level Likert item, strongly agree to strongly disagree)*
- Did you encounter any difficulty navigating through different interface components (homepage to analysis dashboard)? *(Yes, No)*
- If yes, please briefly explain your difficulty.

(2) Annotation Results Visualization and Interpretation

- Were you able to clearly interpret and interact with the annotation results displayed in the scatter plots ("All Examples" and "Suggested Edge Cases")? *(Five-level Likert item, strongly agree to strongly disagree)*
- Did interactive features (e.g., clicking points to highlight examples across lists and plots) support your understanding of annotation results effectively? *(Five-level Likert item, supports fully to doesn't support at all)*
- Would you prefer alternative ways of visualizing or interacting with annotation results visually? *(Yes, No)*
- If yes, please describe briefly.

(3) Edge Case Identification and Handling

- Was the component provided by the system clearly identifying relevant and helpful edge cases (cases that require additional annotation guidance) in your corpus? *(Five-level Likert item, strongly agree to strongly disagree)*
- Do the edge cases make sense? *(Five-level Likert item, makes total sense to makes absolutely no sense)*
- Are the proposed rules easy to follow? *(Five-level Likert item, very easy to very difficult)*
- Please describe briefly your reasoning for accepting or rejecting suggested edge-case rules. What information or criteria were the most important for your decisions?

(4) Iterative Optimization Support

- Did you find the iterative approach ("Iterate" button functionality) helpful for progressively refining your annotation guidelines and labels? *(Five-level Likert item, very helpful to not helpful at all)*
- How many iterations (approximately) did you perform? Did subsequent iterations help significantly in clarifying your annotation guidelines? Please briefly explain.

(5) General User Experience and Satisfaction

- Did Co-DETECT help you to find some new edge cases that you didn't notice before? *(Yes, No, Maybe)*
- How satisfied are you overall with the functionality that this tool offers you in developing codebooks and annotation guidelines? *(Five-level Likert item, very satisfied to not satisfied at all)*
- Do you still have concern that e.g. there are missing edge cases not identified by the system? *(Yes, No, Maybe)*
- What features, if any, do you think are missing or need improvement in this tool?

(6) Open-ended Feedback and Improvement Suggestions

- What did you like the most about the user interface and its functionality?
- What improvements or additions would you propose to enhance the usability or functionality of the current interface?
- (Optional) Any additional comments or suggestions about the tool or your experience using it?