DEPARTMENT OF INFORMATICS
UNIVERSITY OF FRIBOURG (SWITZERLAND)

Human-AI Collaborative Approaches for Open-Ended Data Curation

THESIS

Presented to the Faculty of Sciences and Medicine of the University of Fribourg (Switzerland)
in consideration for the award of the academic grade of
Doctor of Philosophy in Computer Science

by

INES AROUS

from

TUNISIA

Thesis No: 5594
UniPrint
2022

Acknowledgements

First and foremost, I would like to thank God. He has given me strength and guidance throughout all the challenging moments of completing this dissertation.

I would like to express my sincere gratitude to my supervisor Prof. Dr. Philippe Cudré-Mauroux, for his mentorship during the Ph.D. His guidance and support helped me achieve so much personal and professional growth. I will always be grateful for the opportunities he has provided me.

I am incredibly grateful to my co-supervisor, Dr. Mourad Khayati, for his invaluable advice, continuous support, and patience. The meetings and conversations were vital in inspiring me to develop research ideas and form comprehensive and objective critiques. I am also profoundly grateful to my co-supervisor, Dr. Jie Yang. His immense knowledge and ample experience have encouraged me in all the time of my academic research and daily life. His dedication to the field of Human-AI will always be an inspiration to me. His insights and knowledge of the subject steered me through my research. I would also like to thank the rest of my thesis committee, Prof. Jiliang Tang, and the jury president Prof. Rolf Ingold for their availability, feedback, and questions, which helped me create the final form of this thesis.

I would like to thank all the members of the eXascale Infolab group who provided stimulating discussions during coffee breaks, lunches, and many outings. Special thanks to Rana and my office mates Michael Luggen, Giuseppe Cuccu, and Alisa Smirnova, who put up with my stresses during my Ph.D.

This work would not have been possible without my family and friends. Their encouragement and unconditional love made me endure all challenges with ease and comfort. To my parents, thank you for guiding me throughout my career. To my sister, thank you for being there whenever I need you. Special thanks to my husband, Yessine, for his unwavering support and belief in me.

Montreal, December 13, 2022

Abstract

Collecting answers to open-ended questions is essential for many applications ranging from business through influencer finding to science through document mining. For instance, collecting the names of social influencers is useful for brand marketing and opinion mining. Another example of an open-ended task is the extraction of justifications from documents for topic classification, which is helpful in literature review and contextual search. Many efforts have been dedicated to automatically curate such a data through machine learning (ML) models. However, these methods have a limited performance since responding to open-ended questions requires intuition, domain knowledge, and reasoning abilities, which are still missing in state-of-the-art ML methods. Moreover, those models require large-scale and high-quality annotations, whose collection is a long and laborious process. Crowdsourcing provides a cost-effective way to answer open-ended questions in a short amount of time. Furthermore, human annotators on crowdsourcing platforms have diverse skill sets, which allow to collect diverse results. Nonetheless, challenges arise since the answers supplied by workers can be prone to errors. Therefore, it is crucial to design solutions to ensure the quality of open-ended crowdsourced data.

This thesis proposes human-AI collaborative approaches to curate —collect, and clean— open-ended data. Overall, our frameworks comprehend human computation and AI model components that interact with each other. In the human computation component, we model workers' performance and optimize their involvement in open-ended data collection to minimize the cost and maximize the data quality. While in the AI model component, we leverage the task's and answers' features in addition to the worker model to learn the quality of their answers. The human computation component and the AI model are updated iteratively, allowing their learning processes to benefit from each other until an agreement on the quality of the answers is reached. Thus, the interaction between the human computation component and the AI model is bidirectional, which is fundamental to ensuring the effectiveness of the human-AI team.

At the technical level, we design human-AI frameworks to collect and clean open-ended data. In the first framework, we integrate answers' properties and workers' reliability to aggregate the collected open-ended answers. We model the answer's quality as dependent on the answer's features and workers' reliability and derive a variational inference algorithm with efficient updating rules to learn our framework's parameters. We compare our framework with existing methods for finding social influencers and show that it substantially improves the state of the art by 8.44% accuracy. In the second framework, we propose an active learning approach to evaluate open-ended answers. We estimate the model's uncertainty about the quality of open-ended answers and route the most uncertain ones to peers for grading. We then combine the model's estimation and peer grading within a Bayesian framework to improve the model's learning. We apply our method in the scholarly domain where the open-ended answers are in this context, the scholarly reviews, and estimate their conformity to conference standards. We show that our framework outperforms existing methods by 10.85% accuracy. Both frameworks achieve high performance, yet, end-users might reject their results for lack of transparency. Consequently, it becomes essential to provide explainable results. We do so by developing a new explainable method that integrates workers' justifications to infer answers' quality. The proposed method incrementally updates the weights of an attention-based model by learning from human justifications while considering the workers' reliability. Extensive validation on real-world datasets for topic classification shows that our framework significantly improves the state of the art in terms of explainability and accuracy.

We consider the methods introduced in this thesis as a step towards better collaboration between machine learning and human computation. Our work establishes principled optimization algorithms that allow the machine learning model's parameters to be updated using worker's modeling and vice-versa. The developed methodologies can be used as a foundation to build more interpretable methods and provide explanations for both their learning process and results. We envision that crowdsourced open-ended data curation can establish a new research direction to solve complex cognitive tasks and allow workers to "learn, not just earn" through these tasks.

Keywords: Human-AI, Data curation, Variational Inference

Résumé

La collecte de réponses à des questions ouvertes est essentielle pour de nombreuses applications allant du secteur commercial (recherche d'influenceurs) au secteur scientifique (analyse de documents). Par exemple, la collecte de noms d'influenceurs sociaux est utile pour le marketing et l'exploration d'opinions. Un autre exemple de tâche ouverte est l'extraction de justifications à partir de documents pour la classification des thématiques, ce qui est utile pour l'analyse de documents et la recherche contextuelle. De nombreux efforts ont été consacrés à l'extraction automatique de ces données à l'aide de modèles d'apprentissage automatique. Cependant, ces méthodes ont des performances limitées car répondre à des questions ouvertes nécessite de l'intuition, une connaissance du domaine et des capacités de raisonnement, qui font encore défaut dans les méthodes d'apprentissage automatique les plus récentes. En outre, ces modèles nécessitent des annotations de grande qualité et à grande échelle, dont la collecte est longue et laborieuse. Le crowdsourcing offre un moyen rentable pour répondre aux questions ouvertes en peu de temps. De plus, les participants sur les plateformes de crowdsourcing ont des compétences diverses, ce qui permet de collecter des résultats variés. Néanmoins, des défis se posent car les réponses fournies par les participants peuvent être sujettes à des erreurs. Par conséquent, il est crucial de concevoir des méthodes pour assurer la qualité des données ouvertes issues du crowdsourcing.

Cette thèse propose des approches collaboratives humain-IA pour la "curation"—collecte et correction— de données ouvertes. Globalement, nos frameworks comprennent des composants de calcul humain et de modèle d'IA qui interagissent les uns avec les autres. Dans le composant de calcul humain, nous modélisons la performance des participants et optimisons leur implication dans la collecte de données ouvertes pour minimiser le coût et maximiser la qualité des données. Dans la composante modèle d'IA, nous exploitons les caractéristiques de la tâche et des réponses en plus du modèle du participant pour apprendre la qualité de leurs réponses. Le composant de calcul humain et le modèle d'IA sont mis à jour de manière itérative, permettant à leurs processus d'apprentissage de bénéficier l'un de l'autre jusqu'à ce qu'un accord sur la qualité des réponses soit atteint. Ainsi, l'interaction entre le composant de calcul humain et le modèle d'IA est bidirectionnelle, ce qui est fondamental pour garantir l'efficacité de l'équipe humain-IA.

Au niveau technique, nous concevons des frameworks humain-IA pour nettoyer et évaluer les données ouvertes. Dans le premier framework, nous intégrons les propriétés des réponses et la fiabilité des participants pour agréger les réponses ouvertes. Nous modélisons la qualité de la réponse comme dépendant des caractéristiques de la réponse et de la fiabilité des participants et nous dérivons un algorithme d'inférence variationnelle avec des règles de mise à jour efficaces pour apprendre les paramètres de notre framework. Nous comparons notre framework aux méthodes existantes sur la recherche d'influenceurs sur les réseaux sociaux et montrons qu'il améliore considérablement l'état de l'art de 11,5 % d'AUC. Dans le second framework, nous proposons une approche d'apprentissage actif pour évaluer les réponses ouvertes. Nous estimons l'incertitude du modèle quant à la qualité des réponses ouvertes et transmettons les réponses les plus incertaines à des experts pour qu'ils les évaluent. Nous combinons ensuite l'estimation du modèle et la notation par les experts dans un framework bayésien pour améliorer l'apprentissage du modèle. Nous appliquons notre méthode dans le domaine scientifique où les réponses ouvertes sont dans ce contexte, les revues scientifiques, et estimons leur conformité aux normes de conférences. Nous montrons que notre framework surpasse les méthodes existantes par une exactitude de 11,6 %. Ces frameworks sont performants, mais les utilisateurs peuvent rejeter leurs résultats par manque de transparence. Par conséquent, il est essentiel de fournir des résultats explicables. Nous le faisons en développant une nouvelle méthode explicable qui intègre les justifications des participants dans l'inférence de la qualité des réponses. La méthode proposée met à jour de façon incrémentale les pondérations d'un modèle basé sur l'attention en apprenant des justifications humaines tout en considérant la fiabilité des participants. Une validation approfondie sur des ensembles de données du monde réel pour la classification de thématiques montre que notre framework améliore significativement l'état de l'art en termes d'explicabilité et d'exactitude.

Nous considérons les méthodes introduites dans cette thèse comme une étape vers une meilleure collaboration entre l'apprentissage automatique et le calcul humain. Notre travail établit des algorithmes d'optimisation basés sur des principes qui permettent de mettre à jour les paramètres du modèle d'apprentissage automatique à l'aide de la modélisation du participant et vice-versa. Les méthodologies développées peuvent être utilisées comme base pour construire des méthodes plus interprétables et fournir des explications à la fois pour leur processus d'apprentissage et leurs résultats. Nous pensons que la curation de données ouvertes par le crowdsourcing peut établir une nouvelle direction de recherche pour résoudre des tâches cognitives complexes et permettre aux participants "d'apprendre, et pas seulement de gagner" grâce à ces tâches.

Mots clefs : humain-IA, Curation de Données, Inférence Variationnelle

Table of Contents

1 Introduction

1.1 Open-ended Data Curation

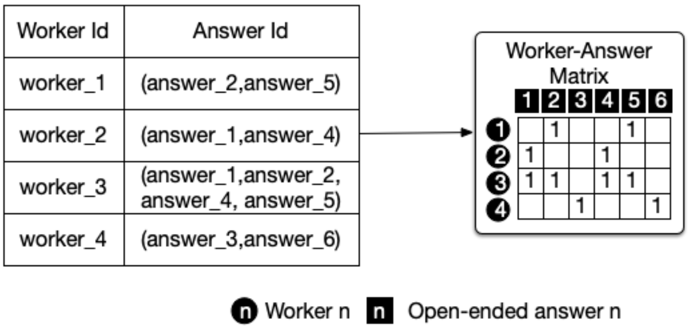
Collecting answers to open-ended questions is important in numerous fields. These questions could be for instance "name a fashion influencer on social media", "write a review about the following article", "justify your rating", which require answers in a free-text form that contain important information cues valuable to many applications. Answers to open-ended questions are used in the business domain where brands collect influencers' user accounts from social media [105, 43, 211], and extract key insights from product reviews [157, 139]. They are also applied in the science domain through language translation [225, 148, 22] and analyzing scientific documents [207, 96] and even in creative work such as fiction writing [89] and design ideation [91, 92]. In this thesis, we use open-ended answers and open-ended data interchangeably to refer to the data collected from open-ended questions.

A lot of effort has been dedicated to curate —collect and clean— open-ended data. Some lines of research focus on ways to generate them automatically. For example, many machine learning models have been proposed to automatically translate textual data from one language to another [74, 229] or generate summaries of scientific documents [55, 151]. While considerable progress has been made in this area, existing methods are still far from perfect in accomplishing these tasks [112, 79]. Moreover, annotation needed to train and evaluate these models are becoming more and more complex as they require domain-specific textual data. Collecting such complex annotations is a long and laborious process even for domain experts. Researchers have been investigating ways to leverage *crowds*, i.e., a large group of participants, for creating open-ended data. Humans excel at complex tasks that require cognitive abilities such as intuition, comprehension, and judgment. In addition, access to a large crowd became easy and possible in a short amount of time, thanks to the emergence of crowdsourcing platforms.

Using crowd workers to collect free-text annotations has led to the emergence of *open-ended crowdsourcing tasks*, which are one of the most popular type of tasks in crowdsourcing platforms [54]. They can be characterized by three main properties: First, their response space can be large and sometimes infinite because workers provide their answers as free-text and thus, they are far less likely to produce identical answers for the same question. Second, a question in an open-ended crowdsourcing task can have multiple correct answers. For instance, there can be multiple correct ways to translate a sentence. Third, they usually require more cognitive effort from workers as they need to reason about the question and generate an answer instead of simply choosing an option among a finite set of possibilities as in traditional crowdsourcing settings.

Collecting open-ended answers from crowd workers is valuable for many reasons. First, open-ended answers are given by a diverse pool of workers, which allows to collect diverse answers and have an accurate representation of the real world. Second, open-ended tasks can be combined with Boolean tasks to improve data quality without significantly increasing the annotation time. For instance, a task where workers are asked to rate a data instance and justify their rating with free text allows to collect more reliable answers than ratings alone. In addition, it offers greater transparency for evaluating both workers and their answers without increasing the task completion time [125]. Third, open-ended answers allow access to more fine-grained information, such as collecting names of domain-specific influencers, which can be helpful in several business applications, including brand marketing and recommendation. Moreover, a lot of studies [223, 130] show the benefit of using fine-grained information as it allows a model to learn both high-level and complex features and increases its ability to generalize.

While open-ended crowdsourcing has a clear advantage over using ML to generate open-ended data automatically, there are challenges to overcome to fully benefit from it. First, crowds can provide incorrect answers, for example, if they are not motivated or qualified for the work. Second, open-ended crowdsourcing tasks have different types and levels of difficulty. Consequently, individual crowds' performance highly depends on the type of task and its difficulty. Moreover, unlike Boolean crowdsourcing, where crowd workers are asked to classify an existing close pool of data instances into *predefined classes*, open-ended crowdsourcing results in open-ended pools of answers – often of large size – that were *all deemed relevant*



Worker Id	Answer Id
worker_1	(answer_2,answer_5)
worker_2	(answer_1,answer_4)
worker_3	(answer_1,answer_2, answer_4, answer_5)
worker_4	(answer_3,answer_6)

Worker-Answer Matrix

	1	2	3	4	5	6
❶		1			1	
❷	1			1		
❸	1	1		1	1	
❹			1			1

Figure 1.1: Example of a worker-answer matrix in open-ended tasks: On the left hand side, the table represents the answers given by each worker (worker_1 gave answer_2 and answer_5, worker_2 gave answer_1 and answer_4, etc.). On the right hand side, we map the collected answers to a worker-answer matrix where rows indicate workers and columns indicate answers.

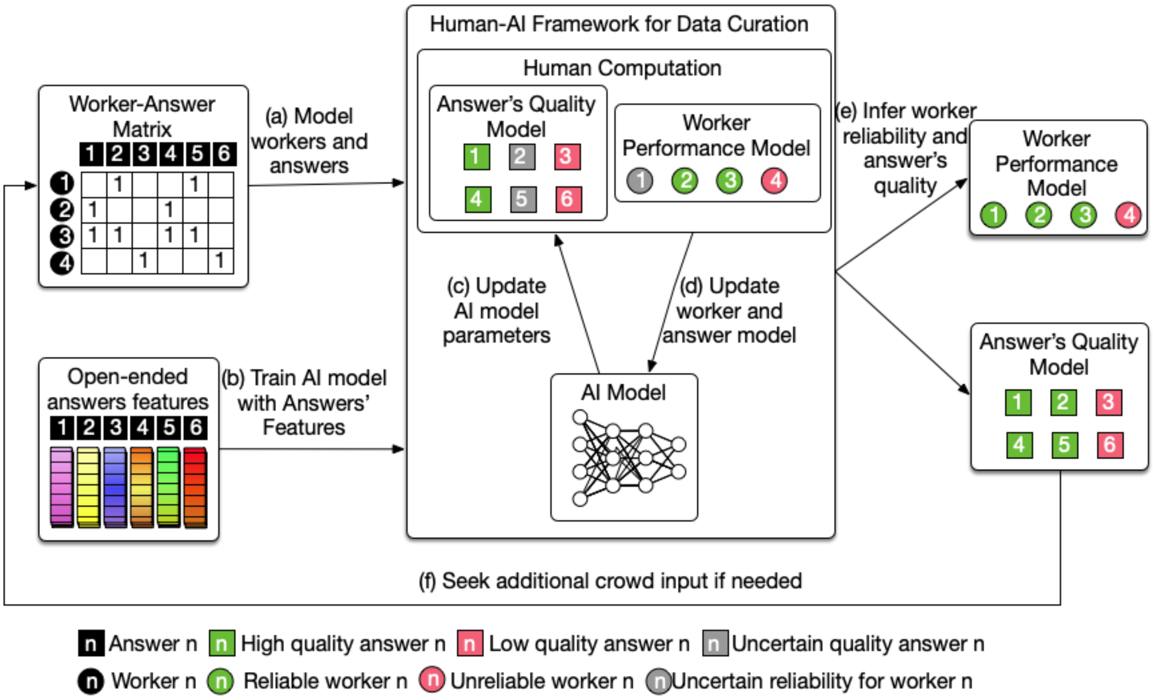
by crowd workers. As a result, the worker-answer matrix in open-ended crowdsourcing is *sparse* and represent *positive-only* data instances as illustrated in Figure 1.1. These properties contrast answers from Boolean crowdsourcing with single-choice or multi-choice questions, where data instances are provided in advance, and workers assign a label indicating a class. Thus, one of the main challenges in open-ended crowdsourcing consists in curating data and filtering incorrect answers.

The problem of curating data in crowdsourcing have been extensively studied over the years [238]. However, researchers have mainly focused on Boolean tasks where workers answers are either binary or categorical. For these tasks, a common practice consists of having multiple workers answer the same question. For instance, in a sentiment analysis task, multiple workers are presented with the same document and asked to select the sentiment it conveys among a set of options (e.g., positive, negative or neutral). Several algorithmic techniques have been developed to aggregate workers' answers leveraging mainly worker's disagreement. A simple aggregation method is majority voting, where the majority of answers are used as the ground truth. Other aggregation methods [50, 238] build on top of the expectation-maximization framework of Dawid and Skene [49], where they model the worker's performance or the task difficulty to estimate the quality of answers. These solutions rely on worker's disagreement which is not sufficient for open-ended answers aggregation.

In this thesis, we propose to address the problem of data curation in open-ended crowdsourcing using *human-AI collaborative approaches*. Our methods integrate both machine learning and crowdsourcing to learn from the task's context and answers' features while considering worker's performance depending on the system's goal. Overall, these hybrid systems are designed to benefit from the complementary strengths of humans and machines intelligence to overcome the challenges raised by the problem of data quality in open-ended crowdsourcing (e.g., sparse and positive-only answers). Such an integration, when carefully designed, has been shown to achieve better performance than humans or machines working alone and mitigate many of the limitations of state-of-the-art AI models [84, 198]. In the human-AI collaborative frameworks we designed along this thesis, the human computation component and the AI model have different roles in our frameworks. Their respective roles are depicted in Figure 1.2 and can be summarized as follows:

- **Human Computation Component:** In our frameworks, the human computation component models the workers' performance and the quality of their answers. When using workers' input, we deal with cost and quality challenges, where we need to minimize the cost while ensuring high data quality. To address this challenge, the human computation component interacts with the AI model component to optimize workers' involvement and identify when and where human input is needed. We do so by updating the AI model parameters based on the inferred data quality and extracting the most informative data instances to annotate.

- **AI Model Component:** The AI component learns the quality of workers' answers from


Human-AI Framework for Data Curation
Human Computation
Answer's Quality Model
1 2 3 4 5 6
Worker Performance Model
1 2 3 4
Worker-Answer Matrix
(a) Model workers and answers
Open-ended answers features
1 2 3 4 5 6
(b) Train AI model with Answers' Features
(c) Update AI model parameters
(d) Update worker and answer model
AI Model
(e) Infer worker reliability and answer's quality
Worker Performance Model
1 2 3 4
Answer's Quality Model
1 2 3 4 5 6
(f) Seek additional crowd input if needed
n Answer n
n High quality answer n
n Low quality answer n
n Uncertain quality answer n
n Worker n
n Reliable worker n
n Unreliable worker n
n Uncertain reliability for worker n

	1	2	3	4	5	6
1		1			1	
2	1			1		
3	1	1		1	1	
4			1			1

Figure 1.2: Human-AI frameworks: (a) Crowds provide open-ended data that we represent as a worker-answer matrix, and we use it to model workers' performance and the quality of their answers. (b) We extract features from the open-ended answers and use them for training the AI model. (c) We update the AI model parameters based on the estimated answer's quality from the human computation component. (d) AI model evaluates the quality of open-ended data, and the worker's performance model is updated accordingly. (e) We iterate between steps (c) and (d) until an agreement is reached on the worker's performance and the answer's quality. (f) We evaluate the results and seek additional crowd input if the framework is uncertain about the quality of answers.

the answers' features and expert labels (if available) and estimates the quality of unseen data instances. We explore several strategies to improve its performance by updating its parameters based on the human computation component. We then update the human computation component based on the learned AI model, where we update the workers' performance model and the answer's quality model using the estimated answers' quality.

Our work contributes to the field of human-AI collaboration paradigm by proposing systems where the interaction between the human computation and the AI model is bidirectional, which is fundamental to ensure the effectiveness of the human-AI team. Since the tasks tackled within this thesis are intrinsically complex and hard for machines to process automatically, it was important to identify how and when to use the human input and simultaneously ensure that the model's behavior meets human's expectation by updating the model's parameters. In establishing such collaborative systems, we pave the way for better collaboration between machine learning and human computation. We foresee the developed frameworks as a first step towards better human-AI systems where they can for instance notify each other in case of uncertainty and easily interpret and rectify each other's learning. In what follows, we discuss in more detail the research questions and the contributions addressed in this thesis.

1.2 Research Questions

This thesis aims at investigating, designing and evaluating methods that effectively and efficiently curate data in open-ended crowdsourcing. The curation pipeline for open-ended answers comprehends multiple aspects including 1) collecting answers from workers and aggregating them by inferring the correct ones, 2) reducing workers' efforts when evaluating open-ended answers, and 3) leveraging the aggregated answers to enhance a model's explainability. In this thesis, we develop frameworks that contribute to each one of these aspects. Specifically, we aim to answer the following research questions:

- RQ1: How can we aggregate open-ended answers?
- RQ2: How can we learn from external sources for truth inference in open-ended tasks?
- RQ3: How can we jointly learn from a peer grading mechanism and a machine learning model the evaluation of open-ended answers?
- RQ4: How can we minimize workers' efforts in open-ended answers evaluation?
- RQ5: How can we effectively learn from humans' justification and enhance an AI model's explainability?

Overall, this thesis pushes the understanding of the open-ended crowdsourcing field and proposes novel human-AI systems where we seek to deeply integrate human and machine intelligence within unified frameworks.

1.3 Summary of the Contributions

In the following, we give a short overview of our main contributions and list the associated conference papers that were published to address the research questions introduced above.

1.3.1 Open-ended answers aggregation

We first tackle RQ1 and RQ2, where we address the problem of open-ended answers aggregation by leveraging external features to infer the truth. We do so by modeling the answers' quality as dependent on their features and the workers' reliability. Our approach is a Bayesian framework that integrates machine learning with crowdsourcing and simultaneously estimates workers' performance and the quality of answers. We formalize the learning processes of the machine learning and the Bayesian models with a variational inference method. The advantage of using variational inference in OpenCrowd is twofold: First, it allows us to have a principled optimization algorithm for updating the parameters of both our probabilistic model and a machine learning model until an agreement on the answers' quality is reached. Second, it allows us to model worker's reliability as a continuous variable, which gives us a measure of *confidence* in estimating worker's reliability.

Our work, OpenCrowd, focuses on data enumeration tasks for influencer finding where workers name social media accounts of candidate influencers. The developed aggregation mechanism infers the real influencers based on both worker's reliability and the features extracted from social media. This work was published as follows:

Arous, Ines, Jie Yang, Mourad Khayati, and Philippe Cudré-Mauroux. "OpenCrowd: A human-AI collaborative approach for finding social influencers via open-ended answers aggregation." In Proceedings of The Web Conference 2020.

1.3.2 Peer grading for open-ended answers' evaluation

Next, we address RQ3 and RQ4 by investigating how to optimize workers' efforts in evaluating open-ended answers. We propose an active learning approach where we estimate a machine learning model uncertainty about its evaluation of open-ended answers and route the most uncertain data instances to peers for grading. Our approach is a Bayesian framework that integrates a machine learning model with peer grading to collaboratively assess multiple criteria in open-ended answers. We model in our framework answers' quality, grader's reliability and bias with continuous variables to quantify the accuracy of our estimation. Similarly to OpenCrowd, we use a variational inference method to update both model parameters and graders' reliability incrementally.

We apply our method to the scholarly domain, where in this context, the open-ended answers are the scholarly reviews and aim to assess their conformity to conference standards based on a set of criteria. Fairness is critical in such a domain; therefore, our human-in-the-loop system allows us to augment the machine learning model's prediction with experts' input and consider experts' bias in grading. Overall, the proposed approach evaluates the conformity of reviews based on extracted features and grades assigned by experts while considering their bias. Our approach was published as follows:

Arous, Ines, Jie Yang, Mourad Khayati, and Philippe Cudré-Mauroux. "Peer grading the peer reviews: a dual-role approach for lightening the scholarly paper review process." In Proceedings of the Web Conference 2021.

1.3.3 Enhancing Model's explainability with open-ended answers

This work tackles RQ5 and investigates ways to learn from human justification to improve the model's explainability. We focus on a particular type of machine learning model: attention-based models [221]. These models assign a distribution to the input units that supposedly reflect their importance. However, recent studies show that the assigned distribution does not correlate with human judgment [78]. Therefore, we propose a framework named "MARTA" that simultaneously learns from workers' labels and justification and trains an attention mechanism such that human justification is aligned with the model's explanation. MARTA iterates between updating workers' performance and the model's learning process by modifying the attention distribution based on human rationales.

We apply our method to classify Wikipedia articles and product reviews according to their topic. In these tasks, workers annotate a text and highlight relevant pieces justifying their annotation. Our framework injects workers' justification into model learning and improves its performance in terms of classification and explainability. This method was published as follows:

Arous, Ines, Ljiljana Dolamic, Jie Yang, Akansha Bhardwaj, Giuseppe Cuccu, and Philippe Cudré-Mauroux. "Marta: Leveraging human rationales for explainable text classification." In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence. 2021.

1.3.4 Additional contributions

In this thesis, we address aspects related to the problem of curation in textual data collected through open-ended questions. Data inconsistencies do not appear only in textual data, but are also prevalent in other types of data. For instance, real-world time series are collected through sensors and often contain blocks of missing values due to some failure or irregular time intervals. Data curation is then needed to recover the missing blocks so that further analysis can be conducted on the time series. In this context, we contribute to a series of projects to address the problem of data curation in time series.

In a first contribution in the time series field, we design a tool named RecovDB, a relational database system enhanced with advanced matrix decomposition algorithm for missing block recovery in time series. We tightly integrate the recovery algorithm with a relational database management system MonetDB to minimize the data conversion and transfer costs. RecovDB achieves high accuracy by benefiting from different types of correlation between time series (positive, negative, and mixed), which was not possible in existing recovery systems. The tool allows users to interact with the system and test different scenarios by recovering multiple time series at once. It also allows users to vary the size of missing blocks and measure the accuracy of the recovery in real time. Our work was published as a demo paper as follows:

Arous, Ines, Mourad Khayati, Philippe Cudré-Mauroux, Ying Zhang, Martin Kersten, and Svetlin Stalinlov. "Recovdb: Accurate and efficient missing blocks recovery for large time series."

In 2019 IEEE 35th international conference on data engineering (ICDE). IEEE, 2019.

The demo is accessible through this link: http://revival.exascale.info/recovery/recovdb.php

In a follow up work, we design a new online recovery technique to recover multiple time series streams in linear time. Our recovery technique implements a novel incremental version of a matrix decomposition technique, namely the centroid decomposition, and reduces its complexity from quadratic to linear. Our method achieves high performance on real-world time series with different properties in terms of accuracy and efficiency. Our technique was published as follows:

Khayati, Mourad, Ines Arous, Zakhar Tymchenko, and Philippe Cudré-Mauroux. "ORBITS: online recovery of missing values in multiple time series streams." Proceedings of the VLDB Endowment 14, no. 3 (2020).

1.4 Thesis Outline

We organize the rest of this thesis as follows. Chapter 2 reviews relevant work on human-AI collaborative approaches and existing methods tackling the problem of data curation in crowdsourcing. Next, in Chapter 3, we present and evaluate our framework for open-ended data aggregation. We center our study around a data enumeration task for influencer finding, where we design a human-AI system that integrates external features from social media for truth inference. Chapter 4 introduces “Peer Grading Peer Reviews", a model that solicits peers to assess the quality of open-ended answers based on multiple criteria. Our framework is designed for the academic domain where open-ended answers are scholarly reviews, and the goal is to assess their conformity to conference standards. Chapter 5 considers ways to leverage open-ended answers to enhance models’ explainability. We inject the worker’s justification expressed as free text in model’s learning to obtain explainable results. We conclude with Chapter 6, summarizing our main findings, and providing an outlook for future developments in human-AI.

2 Background

2.1 Introduction

Crowdsourcing refers to recruiting human workers to solve complex problems that are hard for machines to perform accurately[68]. Nowadays, crowdsourcing is performed on online platforms such as Amazon Mechanical Turk (Mturk), Toloka, and Appen. There are two main parties in these platforms: requesters and crowd workers. Requesters design tasks and post them on a crowdsourcing platform which is then performed by workers. For instance, a crowdsourcing task could be identifying the color of a car in a photo. Due to the wide deployment of these platforms, crowdsourcing became accessible, affordable, and easy. Currently, these platforms count over 500k workers each, and they are operating in at least 100 countries [194, 8, 191]. They offer a friendly user interface for requesters with pre-made templates for various tasks. We distinguish two main types of tasks in crowdsourcing platforms:

- **Boolean Tasks:** They usually require intuition and commonsense. In Boolean tasks, a requester prepares a set of data instances and aims to classify them into predefined categories. They include item comparison, rating, and classification tasks. For example, a popular Boolean task is sentiment analysis, where workers classify a set of documents as positive, negative, or neutral according to the sentiment they convey.
- **Open-Ended Tasks:** They usually require creativity, reasoning, or domain knowledge. In open-ended tasks, requesters aim to collect complex and unstructured data for jobs such as text comprehension and translation, where workers provide their answers as free text. The collected answers in these tasks are unbounded and hence are called open-ended.

In this chapter, we define open-ended tasks' properties and discuss efforts related to the problem of open-ended data curation. Then, we methodologically review existing work related to the developed frameworks within this thesis. We start by giving some background on algorithms and systems leveraging human computation, namely human-in-the-loop systems. Then, we focus on a particular application area of human-in-the-loop systems closely related to our work: human AI. Finally, we discuss the state-of-the-art in the "learn from crowds" line of research.

2.2 Open-Ended Tasks

Open-ended tasks in crowdsourcing require complex annotations that usually cannot be mapped to categorical variables or single-dimensional ordinal variables. In this type of tasks, workers answer in a free-form text and are far less likely to produce identical labels for the same data instance.

We distinguish two categories of open-ended tasks depending on their correct answer space: finite or infinite. Open-ended tasks with finite number of solutions include image description [12, 73], language translation [225, 148, 22] and phrase extraction [162]. The number of possible solutions for these tasks can be large but usually finite. These tasks are often decomposed into a chain of bite-sized subtasks, and requesters can control the quality of answers by predefining correct answers for some subtasks. For instance, a text translation task can be decomposed into sentence-by-sentence translation tasks, and requesters can define a correct translation for some sentences. In this category, we also find tasks with a unique solution, such as audio transcription [48], arithmetic problems [113, 210] or counting, where requesters expect a unique solution as a correct answer. These tasks can be mapped to categorical tasks and evaluated using Boolean truth inference methods [238]. Open-ended tasks with an infinite number of solutions include writing [189, 89], drawing [138, 114] and enumerating domain-specific objects [9, 193]. For instance, writing a story has an infinity of possible correct answers. The answers collected from these tasks reflect the broad knowledge of crowd workers, which can be insightful in domain-specific applications. In our work, we focus on the category of open-ended tasks with a very large to an infinite number of solutions.

Curating open-ended answers with an infinite solution space has been mainly studied in the education field [65, 30, 135] where students solve open-ended problems in the form of a short answer or an essay. The solutions developed in this area rely mainly on a series of pattern matching operations such as regular expressions [82] and semantic word matching [47] or variations of latent semantic analysis [150, 90] to evaluate the answers. Another approach for evaluating open-ended answers in education consists of using peer assessment [208, 126], where students grade each others' answers. In these efforts, there is an assumption that all students share the same knowledge, which is not the case for workers in crowdsourcing platforms. Unlike them, the frameworks developed in this thesis model workers' performance in evaluating open-ended answers.

A separate line of research in human computation and crowdsourcing has investigated the task design for motivating workers to improve the quality of collected open-ended answers. For example, Teevan et al. propose to decompose the task of collaborative writing into sub-tasks and concatenate the individually written paragraphs into a report. A similar system is developed by Kim et al., where they propose to write short fiction stories by having workers iterate between reflecting on general ideas and decomposing them into low-level tasks. Other systems [138, 114] assist designers in drawing and sketching ideas by completing their drawings or generating new sketches. These systems encourage workers to be creative, but the quality of the collected answers is hard to curate [146]. In our work, we address different aspects of curating open-ended answers and aim to propose frameworks to collect and clean them for different downstream tasks.

2.3 Human-in-the-Loop Systems

Human-in-the-loop Systems require human interaction where systems are designed such that they optimize the integration of human contribution to obtain better and more accurate results. Many of them are shaped as Games With A Purpose (GWAP), where game designers leverage the fun incentive to engage crowds into playing a game and, as a side-effect, collect and annotate data [200]. Early examples include the ESP game [201] and Peekaboom [202], where players annotate objects in images. Both games help collect data for training computer vision algorithms. Another popular example is ReCaptcha[203], a security measure used by websites to prevent automated programs from hacking their system by asking humans to perform complex tasks for computers, such as transcribing distorted text. The collected human annotations are then used for digitizing old-printed books. These games helped collect large data quickly, and therefore, several studies have been investigating ways to use gamification to solve large-scale problems. For instance, FindItOut [15] is used to extract relations between concepts and train downstream AI tasks such as commonsense question answering.

Human-in-the-loop systems have been proposed to solve data-related problems in a variety of domains including the database domain. Among the first crowd-powered database systems is CrowdDB [62] proposed in 2011. It is a relational query processing system that uses microtask-based crowdsourcing to answer queries that can not be processed by database systems nor search engines alone. The system was used for subjective comparisons where for instance, workers select the best image for a motivational slide among a set of images, a task relatively simple for humans yet complex for machines. Since then, many specific database problems have been addressed using human-in-the-loop systems, where some have investigated how to implement crowd-based operators such as sorts [124], joins [124, 205] and filters [144], while others were application-oriented such as systems for outlier detection [36], entity resolution [213] and data integration [108]. Another line of work deals with task assignment where several strategies have been developed to optimally assign tasks to workers. For example, QASCA [235] incorporates evaluation metrics into assignment strategies. Other assignment systems [59, 236] consider workers' previous answers and the task domain to match workers with the available tasks.

Another area where human-in-the-loop systems have been extensively used is information retrieval. Systems like CrowdSearcher [29] have been used to obtain answers on domain-specific topics by using crowds on social networks. Other systems were developed to support complex search queries such as DataSift [145] for images and videos queries where workers contribute to the system by reformulating non-textual queries to textual ones and ensuring the retrieved answers are correct, ShapeSearch [177] for searching patterns in large datasets described by workers in natural language queries and DynSP [77] for decomposing complex questions into simple inter-related ones.

A particular application area of human-in-the-loop systems is artificial intelligence where a new field called "Human-AI collaboration" has emerged [198]. Since the frameworks developed within this thesis are closely related to the human-AI field, we discuss in detail the advances established in this area in the next section.

2.4 Human-AI Collaborative Approaches

"Human-AI collaboration" [80, 31, 100], also named "human-AI teaming" [239, 231, 16], or "human-AI hybrid systems" [84, 67] aims to have humans and AI closely collaborating and leverage their complementary strengths to solve tasks that are complex for AI models or humans alone [70, 198]. The existing human-AI systems can be broadly classified into two categories based on the position of human computation in the collaboration pipeline:

2.4.1 Human computation *before* model training

Before model training, human computation is commonly used for data annotation. Examples include crowd annotation datasets for natural language processing tasks such as text classification [226, 125], question-answering [77, 15] and sentiment analysis [181]. It is also used for annotating images and videos such as for the ImageNet [52] and the ESP [201] datasets.

Human computation can also be used for feature selection, where human crowd workers select the most relevant features for the task at hand. These features are often hard to extract automatically from the raw data and can be very useful to build effective predictive or classification models. For instance, Correia and Lecue [46] propose a human-AI framework where domain experts select relevant features and a reinforcement learning model leverages the selected features to minimize the model's loss function. Another approach [240] was proposed for modeling feature discovery via crowdsourcing by detecting common features in multiple data instances. Other methods [42, 165] optimize the involvement of humans in the feature selection process by providing an aggregated view on features to users.

2.4.2 Human computation *after* model training

After model training, human computation can be leveraged in three main procedures: model selection, evaluation and debugging. Model selection indicates the procedure in which a specific AI model can be chosen among a set of candidates according to different evaluation metrics. In the literature, we can find several methods where humans collaborate to select the AI model based on pre-defined criteria. For instance, in [160], crowd workers selected a classification algorithm among a set of models based on its output and an explanation of its results on several data instances. Similar ideas have been explored where end-users select a set of classification models from an ensemble [167] or select the parameters of a model [141] based on the visualization of their classification results.

Metrics such as accuracy and F1 have been extensively used in order to evaluate AI models. Recent studies show that they do not necessarily correlate with human perception of model's performance and therefore, several strategies have been used to simulate human perception in order to evaluate AI models. Some of these strategies rely on explicit user's feedback to evaluate the model's performance by reporting their satisfaction [93, 33] or by setting a threshold on F1 to accept model's results [127]. Another line of research develop new metrics to reflect human judgment [168, 41]. For instance, Schuff et al. propose new metrics to reflect human perception of explainablity in question answering models. Another approach consists in combining different metrics to reflect human evaluation of model's results [41].

Another line of human computation and crowdsourcing research focused on investigating the task design for model's debugging [14, 172, 98]. For instance, the system proposed in [98] explains the reasons for the model's predictions to its end-user, who in turn correct its results by adding relevant features or removing noisy terms. Such a system allowed users to understand the model's functioning better and helped them correct its mistakes. Similarly, other studies investigated how users can interact with machine learning models to understand how input features affect the model's output [95, 45, 14], and distinguish between what the model has learned correctly and what it has missed [172].

The aforementioned human-AI collaborative approaches consider human computation and artificial intelligence as disentangled processes. The methods proposed in this thesis provide ways to deeply integrate human and machine intelligence, where human characteristics (e.g., reliability and bias) and model parameters are iteratively inferred in a mutually boosting manner until the desired result is achieved.

2.5 Learn from Crowds

Our work can also be seen as a development of the “Learn from Crowds" line of research [158, 218, 190], which investigates how to enable machine learning models to learn from noisy labels. The most challenging problems here are two folds: 1) truth inference from noisy labels, and 2) model training given noisy labels and the inferred truth.

To address the first problem, a common strategy in crowdsourcing consists in assigning each data point to multiple workers and then use an aggregation algorithm to infer the truth taking into account worker's annotation quality. A straightforward approach to address the problem of truth inference is Majority Voting (MV), which takes the answer given by the majority of workers as the truth. However, MV has as a main limitation that all workers are regarded as equal, independently from their performance. In reality, an experienced worker carefully reads the instructions and responds to the task, while other workers may randomly answer the task. Thus, it is important to model worker's annotation quality. One of the earliest work in this area can be traced back to Dawid & Skene (D&S) in (1979), where they build an Expectation-Maximization (EM) framework where they model worker's performance as a confusion matrix. Since then, several methods were built on top of the D&S model. For instance, ZenCrowd [50] adopts a similar EM framework while modeling worker's performance with a scalar parameter, which is less expressive than the confusion matrix used in the D&S model but is more robust for sparse worker-answer matrices. Similarly, Whitehill et al. models worker reliability as a single parameter and additionally model task difficulty. These methods aim to only infer the truth from worker's answers and are used as a pre-processing step before model training.

Among the first methods developed to combine truth inference from noisy labels with training AI models was *Learning from crowds* (LFC) [158]. It is an EM framework that estimates the true label by considering both crowd annotations and the output of a logistic regression classifier. However, LFC does not consider the dependency between worker's performance and task properties. To address that problem, Yan et al. proposed to model worker's reliability as dependent on data points properties. A similar approach was proposed by Ma et al. [120] who developed a joint model that captures the task domain from textual descriptions and worker reliability from the worker-answer matrix using Gibbs-EM [204]. Li et al. proposed RAM which models worker's expertise as their ability to distinguish the label relevance with an expectation maximization algorithm.

With the rise of deep learning, a lot of effort has been dedicated to enable deep learning models to learn from crowds [219, 162, 11]. Yang et al. propose a Bayesian framework to learn annotator's reliability and train a deep learning model simultaneously. Rodrigues and Pereira propose a crowd layer that can train deep neural networks end-to-end directly from the noisy labels of multiple workers using backpropagation. Atarashi et al. propose a generative model for semi-supervised learning from crowds where they use a deep neural network for representing the data distribution. Shi et al. address the problem of multi-label semi-supervised learning from crowds and propose a probabilistic method based on a deep sequential generative model. Li'ang Yin et al. use the analogy with autoencoders where they use neural networks as a classifier and a reconstructor, and model inferred labels from crowdsourced data as latent features. Sabetpour et al. propose *AggSLC*, which jointly models the workers' labels and reliability, the machine learning predictions, and the characteristics of a task for sequential label aggregation. To mitigate the effects of annotator group bias on model training, Liu et al. proposed GroupAnno an extended Expectation Maximization algorithm that estimates annotator group bias and update a recurrent neural network simultaneously.

Most of the aforementioned methods rely on computing the exact Bayesian posterior with Expectation Maximization, which requires to integrate over all latent variables. Such an approach, while being deterministic, can be computationally infeasible in complex models and large-scale applications [227]. To that end, one can use Variational Inference (VI), a method that makes Bayesian inference computationally efficient and scalable to large data sets [195, 227]. Since its introduction, VI has been used in several domains including crowdsourcing. The very first work that used variational inference in a crowdsourcing setting was proposed by Liu et al., where they developed a belief propagation and a mean field algorithms for truth inference. Their results show the potential of variational inference as an efficient and effective method in crowdsourcing if the parameters' priors are carefully chosen. Moreover, they prove that variational inference-based methods are a generalization of many existing truth inference methods such as MV and D&S. However, their proposed method cannot leverage machine learning to generalize over other data instances. A similar approach [166] was used to estimate annotators confusion matrices and the probability of each class in a categorical task where they leverage variational inference for fast convergence. Other methods have used variational inference for label aggregation in online scenarios [134]. More recently, Kim et al. proposed two frameworks deepMF and deepBP, alternating variational inference and deep learning to utilize both the prior of worker behavior and the tasks features.

In the frameworks we have developed within this thesis, we develop variational inference algorithms that integrate human computation with artificial intelligence models such that their learning processes benefit from each other. Using variational inference allowed us to scale to larger datasets than the ones traditionally used in crowdsourcing.

2.6 Conclusion

Human-AI collaborative approaches have been used in several domains. A lot of effort has been dedicated to exploring how to best leverage human computation for training AI models. Most of the existing methods suffer from at least one of these two limitations: 1) the human computation and the AI model training are separated, which might lead to biased results since the model is unaware of the labeling sources' accuracy, or 2) they can not deal with large-scale and complex applications as it is the case with open-ended crowdsourcing. In this thesis, we address these limitations and propose methods for improving data quality in open-ended crowdsourcing. In particular, we tackle several aspects of the data curation problem of open-ended crowdsourcing, including cleaning and evaluating open-ended answers and injecting them into a model's learning to improve its explainability.

In the next chapter, we start by addressing the problem of collecting and aggregating open-ended answers. We focus on finding influencers as an application, where we ask workers to provide the names of user accounts they consider influencers.

3 OpenCrowd: A Human-AI Collaborative Approach for Finding Social Influencers via Open-Ended Answers Aggregation

Collecting answers to open-ended crowdsourcing tasks is valuable as it allows to collect large-scale information from a diverse skill set of workers. With thousands of workers participating in crowdsourcing platforms, the access to a large crowd became easy and cheap. Nonetheless, workers' answers are prone to errors due to lack of expertise, knowledge or motivation. To benefit from the collected answers, we need to aggregate open-ended answers and identify the correct ones.

To tackle open-ended answers aggregation, we present OpenCrowd, a unified Bayesian framework that seamlessly incorporates machine learning and crowdsourcing. Our framework bootstraps the learning process using a small number of expert labels and then jointly learns a feature-based answer quality model and the reliability of the workers. Model parameters and worker reliability are updated iteratively, allowing their learning processes to benefit from each other until an agreement on the quality of the answers is reached. We derive a *principled* optimization algorithm based on variational inference with efficient updating rules for learning OpenCrowd parameters. We apply our method to find social influencers where workers name user account of candidate influencers in an open-ended crowdsourcing task. Experimental results on finding social influencers in different domains show that our approach substantially improves the state of the art by 11.5% AUC. Moreover, we empirically show that our approach is particularly useful in finding micro-influencers, who are very directly engaged with smaller audiences.

3.1 Introduction

Social influence is an important mechanism impacting the dynamics of social networks. Social influencers are users who regularly produce authoritative or novel content on specific topics and who can reach and engage a potentially large group of followers. Finding social influencers has become a fundamental task in many online applications, ranging from brand marketing [161, 196] to opinion mining [143, 107], expert finding for question answering [159], minimizing misinformation propagation [182], or analyzing presidential elections [27].

The task of finding social influencers is challenging due to the complexity in quantifying user engagement, the subjectivity in perceiving social influence, and the need for expert knowledge in determining the authenticity of user-generated content. Existing techniques mainly tackle this problem using supervised machine learning approaches that rely on a training set hand-labeled by domain experts [43, 105, 211]. While models trained in this fashion are effective at finding social influencers who are similar to those in the training data, they are intrinsically limited by the availability of expert labels. These labels are typically very hard to gather. As an example, our collaboration with the largest European fashion retailer[1] reveals that an expert can only recognize no more than 200 fashion influencers on Twitter over a 3-week period of time. Finding social influencers is, therefore, a long and usually laborious process even for domain experts [76].

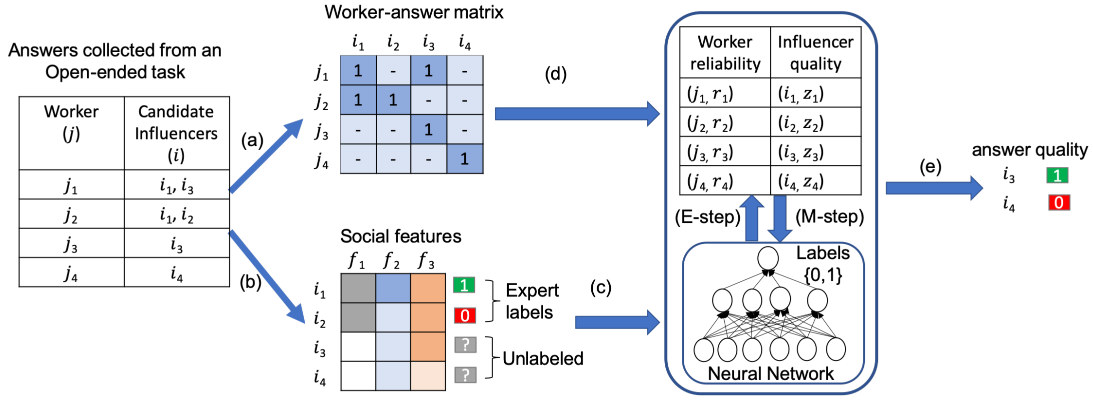


Figure 3.1: The OpenCrowd framework (a) creates a worker-answer matrix from workers answers; (b) extracts social features for the Candidate Influencers $(i_1, \ldots, i_4)$; (c) uses the social features and the expert labels to train an answer quality model and estimate unknown labels; (d) (E-step) uses both the worker-answer matrix and the labels generated by the answer quality model to estimate the worker's reliability and the candidate influencer's labels; (M-step) uses the new candidate influencer's labels (influencer quality) to retrain the answer quality model; and (e) generates labels for the unlabeled candidate influencers.

Compared to an individual expert, online crowds possess *as a whole* a broader knowledge of social influencers in several domains, e.g., fashion, fitness, or information technology. As an example, while it is generally difficult for an expert to come up with a long list of fashion influencers in a short period of time, it is much easier to obtain such a list by asking online workers. Therefore, we advocate a human computation approach that crowdsources the task of finding social influencers in the form of open-ended question-answering, a popular and crucially important, yet severely understudied class of crowdsourcing [146]. Specifically, we consider a task where the crowd is asked to name as many social influencers as possible in a predefined domain. By aggregating the answers from a large number of crowd workers, we can identify the identities (e.g., usernames on Twitter) of a large number of social influencers in an efficient and cost-effective manner.

[1] Zalando SE: https://research.zalando.com/

Despite its obvious benefits, aggregating answers from open-ended crowdsourcing campaigns is challenging: individual crowd workers may only possess fragmented knowledge that is of low-quality. Unlike Boolean crowdsourcing, where crowd workers are asked to classify an existing close pool of data instances into *predefined classes*, open-ended crowdsourcing results in open-ended pools of answers – often of large size – that were *all deemed relevant* by crowd workers. The input data for open-ended answers aggregation is, therefore, a positive-only worker-answer matrix, where each entry indicates the "given by" relationship between an answer and a certain worker, as illustrated in Figure 3.1. This comes in contrast to the input data for aggregating answers from Boolean crowdsourcing, where each entry indicates a class (e.g., 0 or 1 for the binary case) assigned by a worker to a data instance. As an implication, existing answers aggregation methods [49, 214, 212, 237], which are designed to leverage the disagreement between workers' answers, do not yield good performance for open-ended answers aggregation (cf. Section 3.5).

To address the problem of open-ended answers aggregation, we introduce a human-AI collaborative approach that integrates both machine learning and crowdsourcing for aggregating open-ended answers. We present OpenCrowd, a Bayesian framework that models the true label of a candidate influencer as dependent on both the features of the candidate and the reliability of the workers who named the candidate. To infer the truth, OpenCrowd leverages a small number of expert labels to bootstrap the inference process. It then jointly learns a feature-based model for the quality of the answers and the reliability of the crowd workers. The model parameters and worker reliability are updated in an iterative manner, allowing their learning processes to benefit from each other until an agreement on answer quality is reached. The overall learning process is illustrated in Figure 3.1. We formalize such a learning process with a principled optimization algorithm based on variational expectation-maximization. In particular, we derive updating rules that allow both model parameters and worker reliability to be updated incrementally at each new iteration. By doing so, OpenCrowd parameters can be efficiently learned with little extra computational cost compared to the computational cost for training a feature-based answer quality model.

To the best of our knowledge, we are the first to adopt a human-AI collaborative approach for finding social influencers. Our proposed framework is a generic one that can incorporate any machine learning models with crowdsourcing. Moreover, as it solicits contribution directly from crowd workers, the framework is effective in finding a particular type of influencers known as "micro-influencers" [216, 18]. These social influencers are deeply connected to specific niche audiences, thus are able to effectively deliver messages to a highly relevant audience. Unlike macro-influencers who have a huge number of followers (e.g., millions), micro-influencers often have relatively fewer followers, yet they enjoy a more trustworthy reputation (e.g., higher conversion rate in product promotion) and direct relationship with them.

In summary, we make the following key contributions:

- We propose OpenCrowd, a Bayesian framework for finding social influencers through open-ended answers aggregation;
- We derive an efficient learning algorithm based on variational inference with incremental updating rules for OpenCrowd parameter estimation;
- We conduct an extensive evaluation on two domains – fashion and information technology – and show that OpenCrowd substantially improves the state of the art by 11.5% AUC.

3.2 Related Work

In this section, we first review existing answers aggregation methods applicable for finding social influencers. Then, we discuss metric and feature-based techniques proposed to solve this problem.

3.2.1 Answers Aggregation

Influence is defined as “the power of producing an effect on the character or behavior of someone” (Oxford Dictionary). This concept is intrinsically difficult to quantify especially in a large scale context. Answers aggregation provides an efficient and cost-effective manner to identify a large number of social influencers. Methods have been mainly developed in Boolean crowdsourcing. Typical methods include majority voting [174] and those based on expectation-maximization (EM), which simultaneously estimate the true labels and parameters related to the annotation process such as worker reliability and task difficulty [237]. Dawid and Skene [49] make a seminal contribution by proposing to model the worker’s reliability with a confusion matrix for answers aggregation. Demartini et al. [50] address a similar problem while modeling worker reliability as a scalar parameter, which can be less expressive but more robust for highly sparse worker-answer matrices – as we discuss in our experiments. Whitehill et al. [214] introduce a similar method yet further propose to model task difficulty in addition to worker reliability. Closer to our method is LFC proposed by Raykar et al. [158], which models worker reliability as a latent variable with a prior distribution, thus capable of quantifying the uncertainty of the inference. Unlike these techniques, our proposed framework further incorporates existing labels and social features, thus extending the applicability of answers aggregation to open-ended tasks.

While little work has focused on open-ended answers aggregation [146], some techniques consider features of answers or tasks for answers aggregation. A seminal work by Welinder et al. [212] considers the implicit dimensions of worker expertise and task domains and propose a probabilistic model where such dimensions are modeled as feature vectors of tasks to be learned from the worker-answer matrix. A similar line of work takes advantage of explicit task features to learn such dimensions [59, 120, 236]. Ma et al. [120] propose a joint model that captures the task domain from textual descriptions and worker reliability from the worker-answer matrix. Zheng et al. [236] further consider external knowledge bases to better capture task domains. Fan et al. introduce iCrowd [59], which measures the topical similarity of tasks by employing topic modeling techniques (e.g., LDA [25]), and leverages such similarity for a better estimate of worker reliability. A similar idea is investigated by Lakkaraju et al. [101], which also considers similarity among workers based on their features. All these works, however, rely on unsupervised topic models or ad-hoc modeling of specific features to help estimate worker reliability. Our proposed framework is different in that it incorporates crowdsourcing and supervised models that can consume any features.

Human-AI Collaboration.

Our work is related to the emerging field of human-AI collaboration paradigm arising from the intersection between human computation and machine learning [198]. Human computation has been used to enhance machine learning systems by generating the data [237, 219] *before* model training, or providing interpretations for model decisions [160] and debugging the system or the data [137, 220] *after* model training. Typically, human computation and machine learning are treated as disentangled processes. Our approach provides a way to deeply integrate human and machine intelligence in a Bayesian framework, where human characteristics (i.e., reliability) and model parameters are iteratively inferred in a mutually boosting manner until the decisions from aggregated human answers and those from the model agree with each other. Our work can be seen as a development of the “learning-from-crowds” line of research [158, 190, 219], which considers the machine learning problem in the context of noisy labels contributed by the crowd. Unlike existing work, our framework is generic in that it does not assume any type of machine learning models, thus is applicable to a wider range of problems and application domains.

3.2.2 Social Influencer Finding

Existing methods for social influencer finding can be categorized into two classes: metric and feature-based. Common metrics for identifying social influencers include the number of followers, the number of mentions, and the ratio of the number of comments/likes to the number of followers [87, 71]. These metrics are often insufficient to fully capture the degree of influence because of the difficulty to measure content authenticity and engagement with audience. The latter dimension, i.e., engagement, has become a key consideration along with the shift of focus in industry from finding macro-influencers (including celebrities) to micro-influencers [216, 18].

An alternative approach to finding social influencer is machine learning, which can detect influencers by weighting a large number of social features. Existing work has considered a variety of social features including metadata features such as the number of followers and followees [105, 43], the number of retweets and mentions [35], semantic features such as the topics of a candidate influencer’s microposts [159, 211], or features derived from user behavioral data such as the activeness of a candidate influencer in online activities [2, 105]. In addition to those, several pieces of work consider a specific type of feature, namely the structure of the social network among the influencers and other online users, to improve the accuracy in finding social influencers [111, 187, 188, 60, 155, 142]. For instance, Tang et al. [187, 188] propose to find the influencers as the nodes from which the spread of information is maximized. Qiu et al. [155] adopt a deep learning framework where the network embedding and some user-specifc features are fed into a deep neural network for predicting social influencers. Bi et al. [23] introduce a model to incorporate the content of tweets and the followee distribution of microblogs. Similarly, Pal et al. [142] proposed a probabilistic clustering method to produce a ranked list of influencers using node degree, information diffusion, and metrics related to tweets' content.

A less discussed aspect in the machine learning approach is training example creation, which is generally performed by experts through manual screening. The process involves careful examination of content quality, feed consistency, and estimation of the rate of high-quality interactions with the audience. Such a process does not scale for a large number of influencers. Unlike existing methods, our proposed framework only requires a small number of expert labels and shifts the burden of label creation to online workers through crowdsourcing, which is fast, scalable, and cost-effective.

3.3 Problem Formulation

In this section, we first introduce the notations used in the chapter and then formally define our problem.

Notations.

We use boldface lowercase letters to denote vectors and boldface uppercase letters to denote matrices. For an arbitrary matrix $\mathbf{M}$, we use $\mathbf{M}_{i,j}$ to denote the entry at the i-th row and j-th column. We use capital letters (e.g., $\mathscr{P}$) in calligraphic math font to denote sets. We use $x \propto y$ to denote that the two variables x and y are proportionally related, i.e., $x = ky$, where k is a constant.

Table 3.1 summarizes the notations used throughout this chapter. We denote the set of unique *candidate* social influencers named by the crowd workers as $\mathscr{I}$ and the set of workers as $\mathscr{J}$. We use $\mathbf{A}_{i,j} = 1$ to denote that the candidate influencer i is an answer provided by worker j, and $\mathbf{A}_{i,j} = 0$ otherwise. Due to the fact that an individual worker can only provide a limited number of candidate influencers, $\mathbf{A}_{i,j}$ is a sparse matrix where only a small proportion of the entries are non-zero. For each candidate influencer $i \in \mathscr{I}$, we collect her social features as described in detail in Section 3.5.1, and denote the resulting feature vector by $\mathbf{x}_i$. The subset of $\mathscr{I}$, denoted as $\mathscr{I}_{\mathscr{L}} \subset \mathscr{I}$, represents candidate influencers who are associated with expert labels y_i (for $i \in \mathscr{I}_{\mathscr{L}}$). Note that we only have a relatively small number of candidate influencers

Table 3.1: Notations.

Notation	Description
$\mathscr{I}$	Set of candidate social influencers
$\mathscr{I}_{\mathscr{L}}$	Set of labeled candidate social influencers
$\mathscr{I}_j$	Set of candidate influencers relevant to a worker j
$\mathscr{J}$	Set of workers
$\mathscr{J}_i$	Set of workers relevant to a candidate influencer i
$\mathscr{W}_I$	Set of parameters for the answer quality model
$\mathbf{A}$	Worker-answer matrix
$\mathbf{x}_i$	Social feature vector of a candidate influencer
z_i	Influencer quality distribution
r_j	Worker reliability distribution
θ_i	Parameter of the influencer quality distribution
A, B	Parameters of the prior distribution of worker reliability
α, β	Variational parameters of the worker reliability distribution

with expert labels, namely, $|\mathscr{I}_{\mathscr{L}}| \ll |\mathscr{I}|$ and that we aim at estimating the true labels of the candidate influencers who are in $\mathscr{I} \setminus \mathscr{I}_{\mathscr{L}}$.

Problem Definition.

Let $\mathscr{I}$ be a set of candidate social influencers and $\mathscr{I}_{\mathscr{L}}$ be the subset labeled by experts. Let also $\mathscr{J}$ be the set of workers who collectively nominated $\mathscr{I}$, where each candidate influencer can be named by a different number of workers. We aim at inferring the true labels $z_i \in \{0, 1\}$ for all candidate influencers in $\mathscr{I} \setminus \mathscr{I}_{\mathscr{L}}$.

Note that in an open-ended answers aggregation setting, we do not control the number of answers provided by each worker [146]. Hence, the number of workers relevant to different candidate influencer can vary from one to many, rendering the aggregation task highly challenging. This comes in contrast to the conventional crowdsourcing setting, where the number of workers is usually fixed for every data instance (e.g., five workers per instance), which simplifies answers aggregation that relies on worker disagreement.

3.4 The OpenCrowd Framework

OpenCrowd is a unified Bayesian framework that incorporates both supervised learning and crowdsourcing for identifying true social influencers via open-ended answers aggregation. In this section, we first describe the model and then present our variational inference algorithm for learning OpenCrowd parameters. Next, we present an extension of OpenCrowd for multiclass classification.

3.4.1 OpenCrowd as a Generative Model

We represent the generative process of answers as conditioned on both the true labels of the answers and the reliability of the workers. We model the true label of a candidate influencer $z_i \in \{0,1\}$ with a Bernoulli distribution:

$$z_i \sim Ber(\theta_i), \theta_i = \sigma(f^{\mathscr{W}_I}(\mathbf{x}_i)), \tag{3.1}$$

where θ_i is the parameter of the distribution predicted by the social features of the candidate influencer through a feature-based answer quality model, denoted by $f(\cdot)$; $\mathscr{W}_I$ is the set of the model parameters; $\sigma(\cdot)$ is a sigmoid function. We denote $f(\cdot)$ as a generic function that can be instantiated with any supervised learning model, be it a linear model or a neural network. Note that $\mathscr{W}_I$ is shared across all candidate influencers [69], which allows us to exploit the similarity among candidate influencers.

We represent worker reliability as $r_j \in [0,1]$ $(j \in \mathscr{J})$ where $r_j = 1$ indicates that the worker is fully reliable and $r_j = 0$ otherwise. In practice, we would like to have a measure of *confidence* in estimating the reliability of the workers providing different numbers of answers: we should be more confident in estimating the reliability of workers who provide 50 answers than those who provide 5 answers only. To quantify the confidence in our inference, we adopt a Bayesian treatment of r_j by introducing a prior, thus modeling r_j as a latent variable. Given that r_j is a continuous variable in $[0,1]$, we choose a Beta distribution to model its prior:

$$r_j \sim Beta(A,B), \tag{3.2}$$

where A and B are the parameters of the distribution. The incorporation of confidence makes our framework more robust to overfitting, as we show later in Section 3.5.2.

We now define the likelihood of a worker j naming a candidate influencer i as the probability conditioned on the worker's reliability r_j and the true label of the candidate z_i:

$$p(\mathbf{A}_{i,j}|z_i, r_j) = r_j^{\mathbb{1}[z_i = \mathbf{A}_{i,j}]}(1-r_j)^{\mathbb{1}[z_i \neq \mathbf{A}_{i,j}]}, \tag{3.3}$$

where $\mathbb{1}[\cdot]$ is an indicator function returning 1 if the statement is True and 0 otherwise. Note that Eq. (3.3) considers a worker to be reliable if she does not name a candidate influencer who is indeed not a real influencer. It is, however, likely that a worker does not name a candidate influencer i simply because she did not think of i. That means that we can only partly treat the non-named candidate influencers as those the worker considers as non-influencers. It is, therefore, necessary to introduce negative sampling into the inference algorithm.

Negative Sampling.

Negative sampling consists in taking a random sample of candidate influencers not nominated by a worker as her answers of non-influencers. Such negative samples are useful to improve

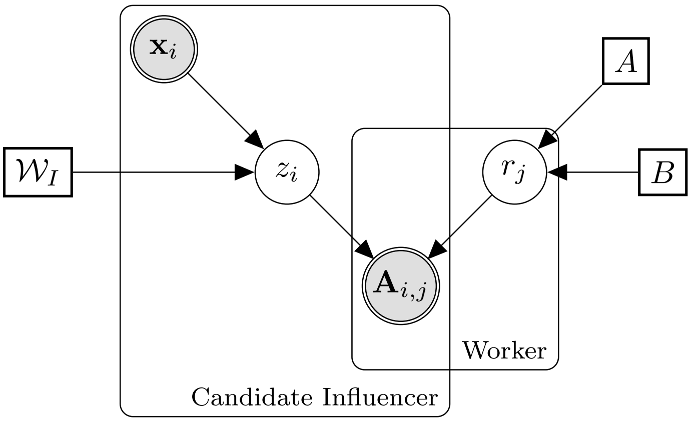


Figure 3.2: Graphical representation of OpenCrowd. Double (greyed) circles represent observed variables, while single circles represent latent variables. Squares represent model parameters. Edges represent conditional relationships in answer generation.

worker reliability inference, as we show in our experiment in Section 3.5.5. For each worker, we consider the candidate influencers named by her and the negatively sampled ones as the candidate influencers relevant to her. Similarly, to estimate the quality of a candidate influencer, we consider not only those workers who have nominated the influencer but also those whose negatively sampled answers contain such an influencer.

The overall OpenCrowd framework is depicted in Figure 3.2. Model learning constitutes of parameter learning for $\mathscr{W}_I$ and posterior inference for the latent variables z_i and r_j.

3.4.2 Variational Inference for OpenCrowd

Learning the parameters of OpenCrowd resorts to maximizing the following likelihood function:

$$p(\mathbf{A}) = \int p(\mathbf{A},\mathbf{z},\mathbf{r}|\mathbf{x}_i;\mathscr{W}_I)\, d\mathbf{z},\mathbf{r}. \tag{3.4}$$

where $\mathbf{z}$ and $\mathbf{r}$ are the latent true labels for all candidate influencers and the reliability of all workers, respectively.

Eq. (3.4) consists of an integral with two latent variables, rendering it computationally unfeasible to optimize [195]. Instead, we consider the log of our likelihood function, i.e.,

$$\log p(\mathbf{A}) = \underbrace{\int q(\mathbf{z},\mathbf{r}) \log(\frac{p(\mathbf{A},\mathbf{z},\mathbf{r}|\mathbf{x}_i;\mathscr{W}_I)}{q(\mathbf{z},\mathbf{r})}) d\mathbf{z},\mathbf{r}}_{\mathscr{L}(\mathscr{W}_I,q)} + \underbrace{\int q(\mathbf{z},\mathbf{r}) \log(\frac{q(\mathbf{z},\mathbf{r})}{p(\mathbf{z},\mathbf{r}|\mathbf{A},\mathbf{x}_i;\mathscr{W}_I)}) d\mathbf{z},\mathbf{r}}_{KL(q||p_{\mathscr{W}_I})} \tag{3.5}$$

where $q(\mathbf{z},\mathbf{r})$ is any probability density function and $KL(\cdot)$ is the KL divergence between two distributions. By doing so, the two parts of the objective function can then be optimized iteratively with a variational expectation-maximization method [195]. Specifically, we iterate between two steps: 1) the E-step where we approximate the distribution of latent variables $p(\mathbf{z},\mathbf{r}|\mathbf{A},\mathbf{x}_i;\mathscr{W}_I)$ with the variational distribution $q(\mathbf{z},\mathbf{r})$, by minimizing the KL-divergence and 2) the M-step where we maximize the first term $\mathscr{L}(\mathscr{W}_I,q)$ of Eq. (3.5) given the newly inferred latent variables.

E-step.

We use the mean-field variational inference approach [26] by assuming that $q(\mathbf{z},\mathbf{r})$ factorizes over the latent variables:

$$q(\mathbf{z},\mathbf{r}) = \prod_i q(z_i) \prod_j q(r_j). \tag{3.6}$$

We further assume the following forms for the factor functions:

$$q(z_i) = Ber(\theta_i), q(r_j) = Beta(\alpha_j,\beta_j). \tag{3.7}$$

where θ_i, α_j and β_j are variational parameters used to perform optimization to minimize the KL-divergence. The latter can then be minimized using coordinate ascent where we update one factor while keeping all others fixed and then iterate until convergence.

In the following, we derive the update rules for the variational distributions $q(z_i)$ and $q(r_j)$. We start by deriving the update rule for $q(z_i)$. Let $p(z_i|x_i;\mathscr{W}_I)$ be the variational distribution of z_i from the last iteration. The KL-divergence in Eq. (3.5) can be easily simplified [26], by keeping only the terms that depend on z_i, to the following:

$$q(z_i) \propto p(z_i|x_i;\mathscr{W}_I) \prod_{j\in\mathscr{J}_i} \exp\{g_{q(r_j)}(p(\mathbf{A}_{i,j}|z_i,r_j))\}. \tag{3.8}$$

where $\mathscr{J}_i$ is the set of workers relevant to a candidate influencer i and $g_x(\cdot)$ is the expectation term $\mathbb{E}_x[\log(\cdot)]$ with x being a variational distribution. Based on this equation, we show in the next lemma how to efficiently update $q(z_i)$ using the feature-based answer quality model and the worker reliability parameters from the previous iteration.

Lemma 1 (Incremental Answer Quality)**.** *The true label distribution $q(z_i)$ of a candidate influencer i can be incrementally updated from the output of the answer quality model θ_i and the worker's reliability parameters α_j and β_j ($j \in \mathscr{J}_i$) in the previous iteration:*

$$q(z_i=1) \propto \begin{cases} \theta_i \prod_{j\in\mathscr{J}_i} \exp\{\Psi(\beta_j)-\Psi(\alpha_j+\beta_j)\}, & if \mathbf{A}_{i,j}=0, \\ \theta_i \prod_{j\in\mathscr{J}_i} \exp\{\Psi(\alpha_j)-\Psi(\alpha_j+\beta_j)\}, & if \mathbf{A}_{i,j}=1. \end{cases} \tag{3.9}$$

where $\Psi(\cdot)$ is the Digamma function. If $q(z_i=0)$ then θ_i is replaced with $(1-\theta_i)$ and, $\Psi(\beta_j)$ and $\Psi(\alpha_j)$ are swapped.

Proof. We show the proof only for $z_i = 1$ since the proof for $z_i = 0$ follows similarly. Using Eq. (3.1), we have:

$$p(z_i = 1|x_i;\mathscr{W}_I) = \theta_i. \tag{3.10}$$

We substitute the probabilities $p(z_i|x_i;\mathscr{W}_I)$ and $p(\mathbf{A}_{i,j}|z_i,r_j)$ in Eq. (3.8) by their respective definitions in Eq. (3.10) and Eq. (3.3) and get:

$$q(z_i=1)\propto \begin{cases} \theta_i \prod_{j\in\mathscr{J}_i} \exp\{g_{q(r_j)}(1-r_j)\}, & \text{if } \mathbf{A}_{i,j}=0,\\ \theta_i \prod_{j\in\mathscr{J}_i} \exp\{g_{q(r_j)}(r_j)\}, & \text{if } \mathbf{A}_{i,j}=1. \end{cases} \tag{3.11}$$

By computing the geometric mean of the beta distribution [121], we can evaluate the expectations $g_x(.)$ as follows:

$$\begin{aligned} g_{q(r_j)}(1-r_j) &= \Psi(\beta_j)-\Psi(\alpha_j+\beta_j),\\ g_{q(r_j)}(r_j) &= \Psi(\alpha_j)-\Psi(\alpha_j+\beta_j). \end{aligned} \tag{3.12}$$

Putting (3.12) into (3.11), the update equation can be simplified as:

$$q(z_i=1)\propto \begin{cases} \theta_i \prod_{j\in\mathscr{J}_i} \exp\{\Psi(\beta_j)-\Psi(\alpha_j+\beta_j)\}, & \text{if } \mathbf{A}_{i,j}=0,\\ \theta_i \prod_{j\in\mathscr{J}_i} \exp\{\Psi(\alpha_j)-\Psi(\alpha_j+\beta_j)\}, & \text{if } \mathbf{A}_{i,j}=1. \end{cases} \tag{3.13}$$

which concludes the proof. □

Next, we show how to efficiently update the variational distribution $q(r_j)$. Let $p(r_j)$ be the variational distribution of r_j from the last iteration, θ' be the true label distribution from the current iteration and $\mathscr{I}_j$ be the set of all candidate influencers relevant to a worker. The KL-divergence in Eq. (3.5) can be simplified, similarly to Eq. (3.8), by keeping only the terms that depend on r_j to get:

$$q(r_j)\propto\ p(r_j)\prod_{i\in\mathscr{I}_j}\exp\{g_{\theta'}(p(\mathbf{A}_{i,j}|z_i,r_j))\}. \tag{3.14}$$

The following lemma shows how to solve Eq. (3.14) using an incremental updating rule.

Lemma 2 (Incremental Worker Reliability)**.** *The reliability distribution $q(r_j)$ of worker j can be incrementally updated using the reliability parameters α_j and β_j from the last iteration and the true label distribution from the current iteration, denoted as θ':*

$$q(r_j)\propto \begin{cases} Beta(\alpha_j+\sum_{i\in\mathscr{I}_j}\theta',\beta_j+\sum_{i\in\mathscr{I}_j}(1-\theta')), & if\mathbf{A}_{i,j}=1,\\ Beta(\alpha_j+\sum_{i\in\mathscr{I}_j}(1-\theta'),\beta_j+\sum_{i\in\mathscr{I}_j}\theta'), & if\mathbf{A}_{i,j}=0. \end{cases} \tag{3.15}$$

Proof. We replace the probablity $p(r_j)$ in Eq. (3.14) by the Beta distribution with parameters α_j and β_j from the previous iteration:

$$q(r_j)\propto Beta(\alpha_j,\beta_j)\prod_{i\in\mathscr{I}_j}\exp\{g_{\theta'}(p(\mathbf{A}_{i,j}|z_i,r_j))\}. \tag{3.16}$$

The expectation term in Eq. (3.16) can be evaluated as follows:

$$\exp\{g_{\theta'}(p(\mathbf{A}_{i,j}|z_i,r_j))\} = \begin{cases} r_j^{\theta'}(1-r_j)^{(1-\theta')}, & \text{if } \mathbf{A}_{i,j}=1, \\ r_j^{(1-\theta')}(1-r_j)^{\theta'}, & \text{if } \mathbf{A}_{i,j}=0. \end{cases} \tag{3.17}$$

In the case when $\mathbf{A}_{i,j}=1$, we use the expressions from Eq. (3.17) to replace the second term in Eq. (3.16) as follows:

$$q(r_j) \propto Beta(\alpha_j,\beta_j)\prod_{i\in\mathscr{I}_j} r_j^{\theta'}(1-r_j)^{(1-\theta')}. \tag{3.18}$$

The probability density function of r_j's distribution is given by:

$$Beta(\alpha_j,\beta_j) \propto r_j^{(\alpha_j-1)}(1-r_j)^{(\beta_j-1)}. \tag{3.19}$$

Putting (3.19) into (3.18), we get:

$$\begin{aligned} q(r_j) &\propto r_j^{(\alpha_j-1)}(1-r_j)^{(\beta_j-1)}\prod_{i\in\mathscr{I}_j} r_j^{\theta'}(1-r_j)^{(1-\theta')} \\ &\propto \prod_{i\in\mathscr{I}_j} r_j^{(\alpha_j-1)}(1-r_j)^{(\beta_j-1)} r_j^{\theta'}(1-r_j)^{(1-\theta')} \\ &\propto \prod_{i\in\mathscr{I}_j} r_j^{(\alpha_j+\theta'-1)}(1-r_j)^{(\beta_j-1+(1-\theta'))} \\ &\propto r_j^{(\alpha_j+\sum_{i\in\mathscr{I}_j}\theta'-1)}(1-r_j)^{(\beta_j+\sum_{i\in\mathscr{I}_j}(1-\theta')-1)}. \end{aligned} \tag{3.20}$$

Thus, if $\mathbf{A}_{i,j}=1$ we have:

$$q(r_j) \propto Beta(\alpha_j+\sum_{i\in\mathscr{I}_j}\theta',\beta_j+\sum_{i\in\mathscr{I}_j}(1-\theta')). \tag{3.21}$$

Following the same steps, we similarly obtain the expression of $q(r_j)$ in case $\mathbf{A}_{i,j}=0$:

$$q(r_j) \propto Beta(\alpha_j+\sum_{i\in\mathscr{I}_j}(1-\theta'),\beta_j+\sum_{i\in\mathscr{I}_j}\theta'). \tag{3.22}$$

□

Algorithm 1: Coordinate Ascent Variational Inference

```
   Input     : $\mathbf{A}, \mathbf{x}_i$ $(\forall i \in \mathscr{I})$, $y_i$ $(\forall i \in \mathscr{I}_{\mathscr{L}})$
   Output    : Variational distributions: $q(z_i)$ and $q(r_j)$
   Initialize: Variational parameters: $\theta_i$, $\alpha_j = A$, $\beta_j = B$; parameter of the influencer
               predictor $(f)$: $\mathscr{W}_I$
1  while Eq. (3.5) has not converged do
2      E-step:
3      for $i \in \mathscr{I}$ do
4          update $q(z_i)$ using Lemma 1;
5      for $j \in \mathscr{J}$ do
6          update $q(r_j)$ using Lemma 2;
7      M-step:
8      for $i \in \mathscr{I}$ do
9          Update $\mathscr{W}_I$ via standard gradient descent;
```

M-step.

Given the true labels of candidate influencers and the worker reliability inferred by the E-step, the M-step maximizes the first term of Eq. (3.5) to learn the parameters α_j and β_j:

$$\begin{aligned}
\mathscr{L}(\mathscr{W}_I, q) &= \int q(z_i, r_j) \log p(\mathbf{A}_{i,j}, z_i, r_j | \mathbf{x}_i; \mathscr{W}_I)\, dz_i, r_j + const. \\
&= \sum_{z_i} \int q(z_i, r_j) \log[p(\mathbf{A}_{i,j} | z_i, r_j) p(z_i | \mathbf{x}_i; \mathscr{W}_I)]\, dr_j + const. \\
&= \underbrace{\sum_{z_i} \int q(z_i, r_j) \log p(\mathbf{A}_{i,j} | z_i, r_j)\, dr_j}_{\mathscr{M}_1} + \underbrace{\sum_{z_i} q(z_i) \log p(z_i | \mathbf{x}_i; \mathscr{W}_I)}_{\mathscr{M}_2} + const. \qquad (3.23)
\end{aligned}$$

where $const. = \mathbb{E}_{q(z_i, r_j)} log(\frac{1}{q(z_i, r_j)})$ is a constant. Only the second part of $\mathscr{L}(\mathscr{W}_I, q)$, i.e., $\mathscr{M}_2$, depends on the model's parameters. $\mathscr{M}_2$ is exactly the inverse of the cross-entropy between $q(z_i)$ and $p(z_i | \mathbf{x}_i; \mathscr{W}_I)$, which is widely used as the loss function for many classifiers. $\mathscr{M}_2$ can, therefore, be optimized using standard methods [136] (e.g., back-propagation in the case of a neural network).

3.4.3 Algorithm

The overall optimization algorithm is given in Algorithm 1. It iterates over the E-step (rows 2-5) and the M-step (rows 6-7) until our objective function converges. In rows 2-3, we iterate through all candidate influencers where for each candidate influencer i, we update $q(z_i)$ using Lemma 1. Similarly in rows 4-5, we iterate through all workers where for each worker j we update $q(r_j)$ using Lemma 2. In rows 6-7, $\mathscr{W}_I$ can be incrementally updated starting from the values in the previous iteration. The convergence is reached when answer quality $q(z_i)$ is no longer modified by worker reliability in the previous iteration (Eq. (3.8)) and it no longer updates the parameters of the answer quality model (Eq. (3.23)).

The iterations through the relevant workers for each candidate influencer (rows 2-3) require a time complexity of $O(|\mathbf{A}|)$, where $|\mathbf{A}|$ denotes the number of non-zero entries of $\mathbf{A}$. Similarly, the time complexity for row 4-5 is $O(|\mathbf{A}|)$. The overall complexity of the algorithm is, therefore, $O(\#iter \times |\mathbf{A}| + \mathscr{T}_{\mathbf{W}})$, where $\#iter$ is the number of iterations in the variational inference algorithm and $\mathscr{T}_{\mathbf{W}}$ represents the complexity of learning $\mathscr{W}_I$ in a supervised learning setting.

3.4.4 Multiclass OpenCrowd

We extend our framework to perform multiclass classification of open-ended answers where we apply a one-vs-rest strategy. We proceed by using a classifier per class such that for each data instance, we have one class as positive and remaining classes as negative. We maintain the same algorithmic procedure where we iterate between an E step and an M step. For the E step, we modify the formulas in Lemmas 1 and 2 to consider the multiclass case. While for the M step, we reuse the reasoning in Section 3.4.2 to update the parameters of a multiclass classifier. In the following, we describe the alteration of the E step for multiclass classification.

We propose to update Lemma 1 such that we maintain the incremental relation between of the true label distribution $q(z_i)$ and the worker's reliability parameters as follows:

$$q(z_i = c) \propto \begin{cases} \boldsymbol{\theta}_{i,c} \prod_{j \in \mathscr{I}_i} \exp\{\Psi(\beta_j) - \Psi(\alpha_j + \beta_j)\}, & \text{if } \mathbf{A}_{i,j} \neq c. \\ \boldsymbol{\theta}_{i,c} \prod_{j \in \mathscr{I}_i} \exp\{\Psi(\alpha_j) - \Psi(\alpha_j + \beta_j)\}, & \text{if } \mathbf{A}_{i,j} = c \end{cases} \tag{3.24}$$

where $\theta_{i,c}$ is the predicted class through the feature-based answer quality model and $\mathscr{C}$ is the set of classes. We use the conditions $\mathbf{A}_{i,j} = c$ to update the class probability c when a worker j selects it as the class for an influencer i.

Similarly, we modify Lemma 2 and propose to incrementally update the worker's reliability using the reliability parameters α_j and β_j from the last iteration and the predicted answer quality from the current iteration, denoted as $\boldsymbol{\theta'}$:

$$q(r_j) \propto \begin{cases} Beta(\alpha_j + \sum_{i \in \mathscr{I}_j} \boldsymbol{\theta}'_{i,\mathbf{A}_{i,j}}, \beta_j + \sum_{i \in \mathscr{I}_j} (1 - \boldsymbol{\theta}'_{i,\mathbf{A}_{i,j}})), & \text{if } \mathbf{A}_{i,j} \in \mathscr{C}, \\ Beta(\alpha_j + \sum_{i \in \mathscr{I}_j} u, \beta_j + \sum_{i \in \mathscr{I}_j} (1 - u)), & \text{if } \mathbf{A}_{i,j} = 0 \end{cases} \tag{3.25}$$

Note that in this case $\boldsymbol{\theta}'_i$ is a vector containing the class probabilities obtained in the previous step through Eq. (3.24). The condition $\mathbf{A}_{i,j} \in \mathscr{C}$ means that the worker named influencer i and selected its corresponding class, while the condition $\mathbf{A}_{i,j} = 0$ means that the worker did not name influencer i. For the latter case, we use a parameter $u \in [0,1]$ that we set empirically.

The overall OpenCrowd for multiclass classification algorithm is presented in Algorithm 2. It iterates over the E-step (rows 2-5) and M-step (rows 6-7) until the evidence lower bound (ELBO) [26] converges.

Algorithm 2: OpenCrowd for Multiclass Classification

```
Input      : $\mathbf{A}, \mathscr{C}, \mathbf{x}_i \ (\forall i \in \mathscr{I}), y_i \ (\forall i \in \mathscr{I}_{\mathscr{L}})$
Output     : Variational distributions: $q(z_i)$ and $q(r_j)$
Initialize : Variational parameters: $\theta_i$, $\alpha_j = A$, $\beta_j = B$; parameter of the influencer
             predictor ($f$): $u, \mathscr{W}_I$
while the ELBO has not converged do
    for $i \in \mathscr{I}$ do
        update $q(z_i)$ using Equation (3.24);
    for $j \in \mathscr{J}$ do
        update $q(r_j)$ using Equation (3.25);
    for $i \in \mathscr{I}$ do
        Update $\mathscr{W}_I$ via standard gradient descent;
```

3.5 Experiments and Results

This section presents experimental results evaluating the performance of OpenCrowd[2] on two different domains, by comparing it against state-of-the-art Boolean and feature-based aggregation methods. In addition, we investigate the properties of our approach such as the impact of negative sampling on the performance. We start by introducing our experimental setup below before presenting the results of our experiments.

3.5.1 Experimental Setup

Crowdsourcing Task.

We consider the problem of finding social influencers in two domains: fashion and information technology. For both domains, we published question-answering tasks on Figure Eight[3], asking workers to name social influencers they know. To set the context and promote workers to reflect on their experience, we asked workers to assess their domain-specific knowledge (five-point scale), estimate how often do they read social media posts from influencers (never, rarely, sometimes, always), and describe how they got to know the influencers. Workers name candidate influencers by providing their Twitter usernames, from which we retrieve social features (see “Social Feature Extraction”).[4] The task took 2 minutes to complete on average. Workers who completed the task received a reward of 30 cents (USD), with an additional bonus: they were paid 10 additional cents (and up to 50 cents) for every social influencer they provided after naming 3 influencers.

[2] The implementation is available here: https://github.com/eXascaleInfolab/OpenCrowd.

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

[4] Twitter usernames are first verified automatically through Twitter API.

Datasets.

We collected two datasets of candidate influencers in two domains: *Fashion* and *Infotech*. The size of the collected datasets are comparable to typical datasets for Boolean answer aggregation [175]. Key statistics of these datasets are reported in Table 3.2. For evaluating the multiclass version of OpenCrowd, we additionally ask workers to classify 497 *Fashion* candidate influencers to one of the three categories emerging influencer (0), established influencer (1) and other (2). Our manual analysis (see "Expert Assessment") revealed that 30.64% and 43.39% of the crowd answers designate true influencers for Fashion and InfoTech, respectively. The relatively large number of crowd answers collected in a short period of time (<10 hours for both Fashion and InfoTech) confirms our assumption that crowdsourced open-ended question-answering can drastically speed up the data collection for finding social influencers. Moreover, the high sparsity of the answer matrices (Table 3.2) and the fact that the majority of the answers are incorrect substantiates the necessity of open-ended answers aggregation that takes into account the workers' reliability.

Expert Assessment.

We conducted a series of interviews with experts from three leading companies [5] that connect brands to social influencers. We distilled four main characteristics of influencer assessments: authenticity, dedication, branding, and communication. Following their guidelines and examples, three of the authors randomly selected 40% of the candidate influencers and labeled them by manually examining their profile and content on Twitter. In more detail, a candidate was considered as a real influencer whenever she: 1) tweets about a specific topic; 2) posts new content regularly; 3) keeps a consistent and unique style in her posts; and 4) communicates with her followers through comments (mostly for micro-influencers). The authors reached an initial agreement of over 80%. In case of disagreement, they discussed it until reaching a decision.

Social Feature Extraction.

The features used in our framework are extracted from the Twitter account of the named candidate influencers. These features include metadata features such as the number of followers, number of followees and number of tweets, and semantic features such as the topics of a candidate influencer's tweets. In order to extract the topics from the tweets, we first represent all tweets as a bag of words. Then, we apply a grid search in {5, 10, 20, 50, 100} to set a threshold on the word's frequency. For our experiments, we keep only the words that appear more than 20 times. We finally compute the TF-IDF scores of the constructed bag of words and use the scores together with the other features to train our answer quality model.

[5] Collabary (collabary.com), Influencer Check (influencer-check.ch), and Reachbird (reachbird.io)

Table 3.2: Description of the datasets.

Dataset	#Cand. Infl.	#Workers	#Answers	Sparsity
Fashion	890	250	1416	99.36%
InfoTech	1057	200	1643	99.22%

Table 3.3: Performance (accuracy and AUC) comparison of aggregation techniques on two datasets with supervision degree s_deg from 50% to 90%. The best performance is highlighted in bold; the second best performance is marked by '*' for accuracy and by '+' for AUC.

Method	**Metric**	**Fashion**					**InfoTech**				
		50%	60%	70%	80%	90%	50%	60%	70%	80%	90%
DS	Accuracy	0.689	0.716*	0.703	0.688	0.711	0.662	0.660	0.626	0.641	0.536
	AUC	0.191	0.169	0.242+	0.244	0.263	0.174	0.203+	0.222+	0.255	0.272
GLAD	Accuracy	0.697	0.716*	0.724	0.700	0.688	0.669*	0.667	0.637	0.672	0.595
	AUC	0.183	0.189	0.229	0.224	0.263	0.150	0.186	0.138	0.219	0.307+
ZenCrowd	Accuracy	0.701	0.686	0.733*	0.702*	0.688	0.651	0.674*	0.664*	0.683*	0.627
	AUC	0.157	0.175	0.203	0.239	0.287+	0.146	0.198	0.212	0.246	0.234
LFC	Accuracy	**0.721**	0.694	0.718	0.691	0.755*	0.653	0.627	0.643	0.616	0.636*
	AUC	0.203+	0.203+	0.225	0.264+	0.277	0.189+	0.192	0.215	0.276+	0.307+
OpenCrowd	Accuracy	0.708*	**0.740**	**0.751**	**0.769**	**0.889**	**0.734**	**0.782**	**0.790**	**0.797**	**0.804**
	AUC	**0.304**	**0.350**	**0.350**	**0.452**	**0.495**	**0.270**	**0.279**	**0.353**	**0.326**	**0.339**

Comparison Methods.

Due to the lack of existing open-ended answers aggregation methods (cf. Section 3.2), we first compare against the following state-of-the-art closed-pool (Boolean) aggregation methods: 1) ZenCrowd [50], an expectation-maximization (EM) method that estimates worker reliability as a model parameter; 2) Dawid-Skene (DS) [49], an EM method that learns worker reliability as a confusion matrix; 3) GLAD [214], an EM method that simultaneously learns worker reliability and task difficulty; and 4) LFC [158], an EM method that incorporates priors in modeling worker reliability. Then, we compare against existing techniques that take into account task's features for answer aggregation: 1) LFC_SoT [190], a statistical model that estimates both worker reliability and task clarity by clustering workers into groups; 2) CUBAM [212], a Bayesian probabilistic model that learns worker reliability and task domains as a feature vector from the worker-answer matrix; and 3) iCrowd [59], a crowdsourcing framework that considers the topical similarity of tasks based on their textual description for worker reliability inference. For the evaluation of multiclass OpenCrowd, we compare with 4) FaitCrowd [120], an unsupervised method that captures the topics of tasks based on their textual description and models topic-specific worker qualities; and 5) a variation of BCCWords [179], which models worker reliability in the form of confusion matrices. The original version of BCCWords can only consume textual features, we adapt it such that it can process any type of features.

We compare all the EM-based methods in a semi-supervised setting by fixing the known labels in the EM algorithm [175] (See [185, 206] for the case of semi-supervised DS). Then, in order to apply these methods to our problem, we use negative sampling to simulate a worker's answers of non-influencers by sampling the candidate influencers she does not name. We empirically determine the optimal sampling rates for each comparison method. Furthermore, for the techniques that model a task based on its textual description, we use the textual social features as input to model a candidate influencer.

To further investigate the benefits of taking into account the worker-answer matrix, we compare OpenCrowd against some feature-based methods: logistic regression (LR) and a multi-layer perceptron (MLP). We define two variants of our framework: 1) OpenCrowd-EM: OpenCrowd that aggregates workers' answers but models worker reliability as a fixed parameter; and 2) OpenCrowd, our framework that models worker reliability as a latent variable.

Parameter Settings.

The parameters of our framework and those for model training are empirically set. We search for the best model architecture for MLP, and the predictor f in OpenCrowd-EM and OpenCrowd, with 0, 1, and 2 hidden layers, and apply a grid search in {64, 128, 256, 512, 1024} for the dimension of the hidden layers. In model training, we select learning rates from {0.0001, 0.001, 0.01, 0.1, 1} for the learning of $\mathscr{W}_I$ in all variants of our framework, as well as for the learning of r_j in OpenCrowd-EM. To investigate the impact of negative sampling, we experiment with sampling rates (s_rate) in {0, 0.1, 1, 10, 100} where $s\_rate = 10$ indicates for example that for each worker, the negative samples are ten times the size of the candidate influencers named (i.e., deemed as positive) by each worker. For OpenCrowd, we set the priors A and B by sampling from a uniform distribution $\sim [0, 10]$ and update them in the E-step according to Lemma 2.

Evaluation Protocols.

We split the labeled subset of candidate influencers into training, validation, and test sets. OpenCrowd is trained on the answers in the training set, tuned on the validation set and evaluated on the test set. To investigate the impact of the degree of supervision (s_deg) on OpenCrowd performance, we split the labeled subset by $s\_deg \in \{50\%, 60\%, 70\%, 80\%, 90\%\}$, where $s\_deg = 60\%$ means that 60% of the labeled subset is used for training, and the rest for validation and test with equal split. We use accuracy and area under the precision-recall curve (AUC) to measure the performance. Higher values of accuracy and AUC indicate better performance. Note that given the imbalanced classes in our datasets, accuracy is dominated by the results on the non-influencers; similarly, the metric area under the ROC curve would also be biased to the non-influencers. In contrast, the AUC we use – area under the precision-recall curve – is *more indicative* of the performance in our context, as we are more interested in detecting real influencers from the workers' answers [28].

3.5.2 Comparison to Boolean Aggregation

Table 3.3 summarizes the performance of boolean answers aggregation methods on our two datasets with different supervision degrees. We make several observations.

First, we observe that ZenCrowd outperforms DS and GLAD in terms of accuracy and has a comparable performance in terms of AUC. Recall that ZenCrowd is less expressive compared to DS and GLAD, as it only models worker reliability as a parameter. In comparison, DS models worker reliability as a confusion matrix, and GLAD further models the task difficulty (in our context the ambiguity of a candidate influencer being the true influencer). The comparison result indicates that in our context, more expressive models do not necessarily lead to higher performance. This is likely due to the high sparsity of the worker-answer matrices that can easily lead to overfitting. Second, we observe that methods that model worker reliability as a latent variable with a prior distribution, namely LFC and our framework OpenCrowd, outperform the other methods. Such a result confirms the necessity of modeling worker reliability as a latent variable, as it helps to account for the confidence in estimating model parameters. This is particularly important to improve model robustness for sparse datasets similar to our case. We provide more results about this point in Section 3.5.5.

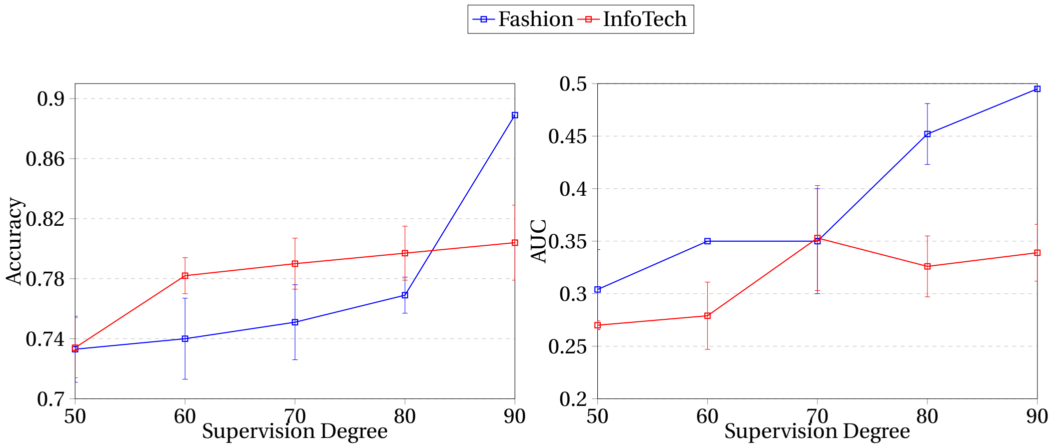
Most importantly, OpenCrowd achieves the best performance among all answers aggregation methods under comparison: it improves the state of the art by 6.94% accuracy and 62.06% AUC on Fashion, and by 17.56% accuracy and 33.54% AUC on InfoTech. This significant improvement clearly demonstrates the effectiveness of our framework in open-ended answers aggregation.

Impact of Supervision Degree.

The supervision degree s_deg controls the number of observed labels in model training. We observe that the performance of our framework increases along with the increase of s_deg, as measured by both accuracy and AUC. This is natural as using more labeled data provides more information in discriminating influencers from non influencers. Such a pattern, however, is not observed for the other methods that we compare to. This is likely due to the fact that the other methods do not take advantage of the social features, which are useful as they serve as a means to propagate the labels to non-labeled candidate influencers. These results show that OpenCrowd is better at utilizing existing labels for answers aggregation.

Robustness.

The learning of OpenCrowd involves two types of random processes, i.e., the random initialization of the parameters (e.g., $\mathscr{W}_I$) and negative sampling. To investigate their impacts on OpenCrowd performance, we measure the standard deviation of OpenCrowd performance over 10 runs as depicted in Figure 3.3. Results show that the standard deviation in terms of accuracy is 0.017 and 0.018 on Fashion and InfoTech, respectively; and in terms of AUC, the



(a) Accuracy

(b) AUC

Figure 3.3: Performance of OpenCrowd across 10 runs.

Table 3.4: Performance (accuracy and AUC) comparison of feature based aggregation techniques on the two datasets.

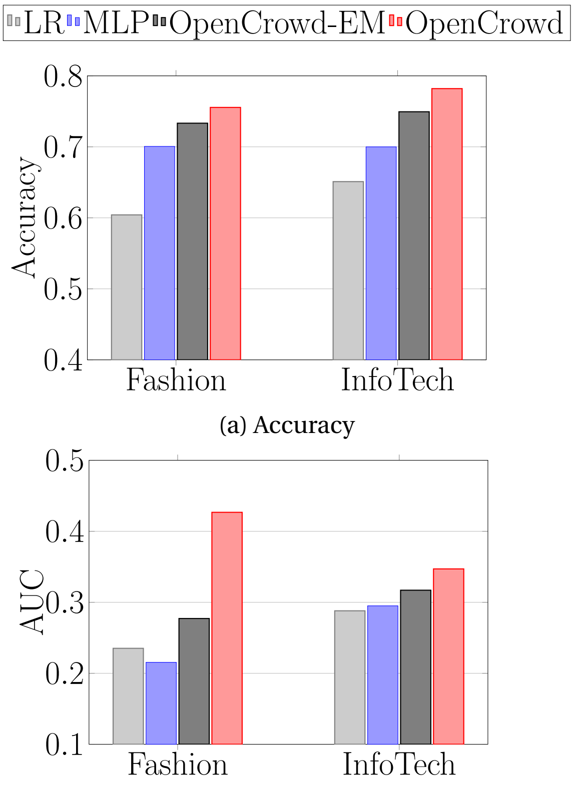
Dataset	Method	Acuracy	AUC
Fashion	CUBAM	0.718	0.290
	iCrowd	0.712	0.301+
	LFC_SoT	0.724*	0.253
	OpenCrowd	**0.751**	**0.350**
InfoTech	CUBAM	0.661	0.326
	iCrowd	0.630	0.372+
	LFC_SoT	0.698*	0.252
	OpenCrowd	**0.790**	**0.397**

standard deviation is 0.023 and 0.028 on Fashion and InfoTech, respectively. The standard deviations are small compared to the absolute accuracy and AUC. Such a result is consistent across different supervision degrees. These results signify the robustness of OpenCrowd across different runs.

3.5.3 Comparison to Feature-Based Aggregation

We now compare the performance of our method against feature-based aggregation techniques. Table 3.4 shows the results of our comparison against these methods in terms of accuracy and AUC on both the Fashion and InfoTech datasets (with a supervision degree of 60%). From these results, we make the following observations.

Among the baselines, LFC_SoT achieves the best accuracy yet the lowest AUC. In fact, LFC_SoT cannot handle the case where some workers do not give an answer to some tasks and hence cannot properly support negative sampling. Since the worker-answer matrix is very sparse in our setting, LFC_SoT labels most candidate influencers as negative (more than 75%). Therefore, LFC_SoT infers most true influencers to be non influencers and hence the results. In contrast,



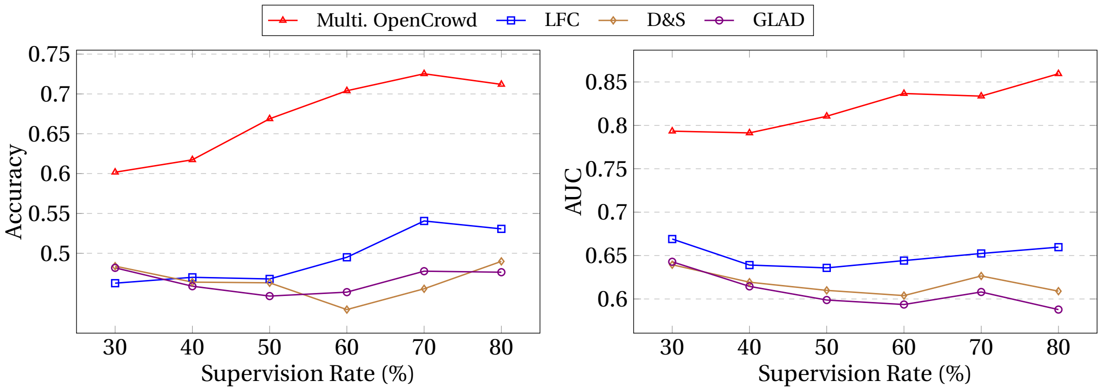
(a) Accuracy

(b) AUC

Figure 3.4: Comparison between feature-based methods and OpenCrowd variants measured by (a) Accuracy and (b) AUC.

iCrowd achieves better performance in terms of AUC than CUBAM and LFC_SoT. Recall that iCrowd takes into account the social features of candidate influencers and combines them with worker reliability to infer the truth. In comparison, CUBAM models the task difficulty as a vector but relies solely on the worker-answer matrix. This result confirms the necessity of taking into account the social features to identify real influencers in the set of candidates.

Overall, OpenCrowd achieves the best performance among all feature-based aggregation methods: it outperforms the second best method by 3.7% accuracy and 16.27% AUC on Fashion and by 13.18% accuracy and 6.7% AUC on InfoTech (on average: 8.44% accuracy and 11.5% AUC). Unlike the baseline methods that do not use social features (e.g., CUBAM) or rely only on textual features (e.g., iCrowd), OpenCrowd is able to leverage any type of social features, including non-textual ones. More importantly, unlike the unsupervised topic modeling used by iCrowd, the supervised answer quality model in OpenCrowd learns from the labeled data the "weights" of social features, thereby making it better at influencer identification.



(a) Accuracy - Influencer

(b) AUC - Influencer

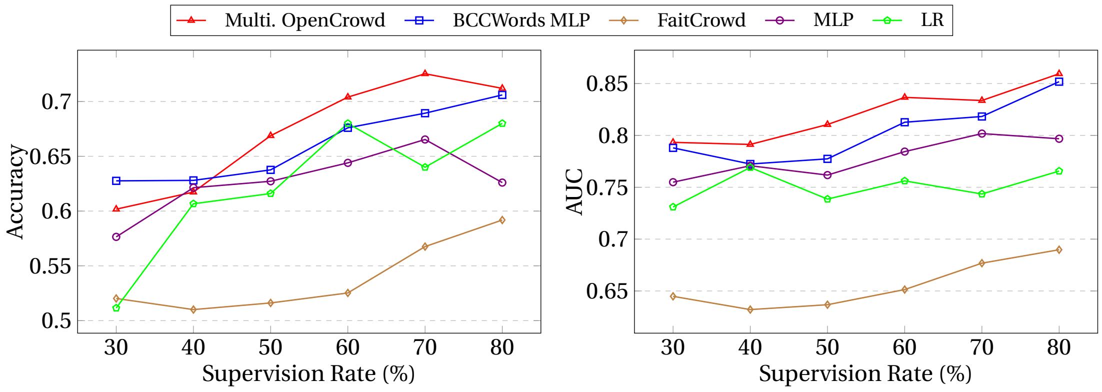
Figure 3.5: Performance of categorical aggregation methods with varying supervision rates.

3.5.4 Evaluation of Multiclass Classification OpenCrowd

We evaluate the multiclass version of OpenCrowd described in Section 3.4.4 with categorical and feature-based methods on the Fashion influencer dataset.

Our comparison with categorical methods is shown in Figure 3.5. We compare Multi. OpenCrowd with the Boolean methods described in Section 3.5.1, except for ZenCrowd [50] as it can not be applied for multiclass classification tasks. In our experiment, we vary the supervision degree between 30% and 80% and measure the performance in accuracy and AUC. First, we observe that increasing the supervision degree improves the performance of LFC in terms of accuracy but does not significantly impact its performance in terms of AUC. Similarly, D&S and GLAD do not improve when increasing the supervision degree. This result is consistent with our comparison with Boolean methods in the binary setting in Section 3.5.2, where we found that they do not benefit from the social features and thus can not benefit from the increase of the labeled candidate influencers. Second, our feature-based method (Multiclass OpenCrowd) generally outperforms baseline methods in accuracy and AUC. Overall, it improves the second best method (LFC) by 20% accuracy and 30% AUC. Finally, we observe that our method improves with increasing the supervision degree as it leverages the labeled data to better discriminate between workers' answers.

Figure 3.6 shows the comparison between Multi. OpenCrowd and feature-based methods. Similarly to our comparison with categorical methods, we vary the supervision degree between 30% and 80% and measure the performance in accuracy and AUC. First, we observe that FaitCrowd performs poorly compared to the baseline methods. Recall that FaitCrowd is an unsupervised approach and does not benefit from the labels to learn the weight of answers' features. The improvement we observe for FaitCrowd when increasing the supervision rate is simply due to the smaller size of the testing set. Second, we observe that the performance of supervised methods (BCCWords MLP, MLP, and LR) increases when increasing the supervision rate. This result is expected as they can better discriminate classes when more labeled data



(a) Accuracy - Influencer

(b) AUC - Influencer

Figure 3.6: Performance of feature-based aggregation methods with varying supervision rates.

becomes available. Finally, we observe that BCCWords MLP outperforms FaitCrowd, MLP, and LR in accuracy and AUC, demonstrating the importance of incorporating workers' answers in their learning.

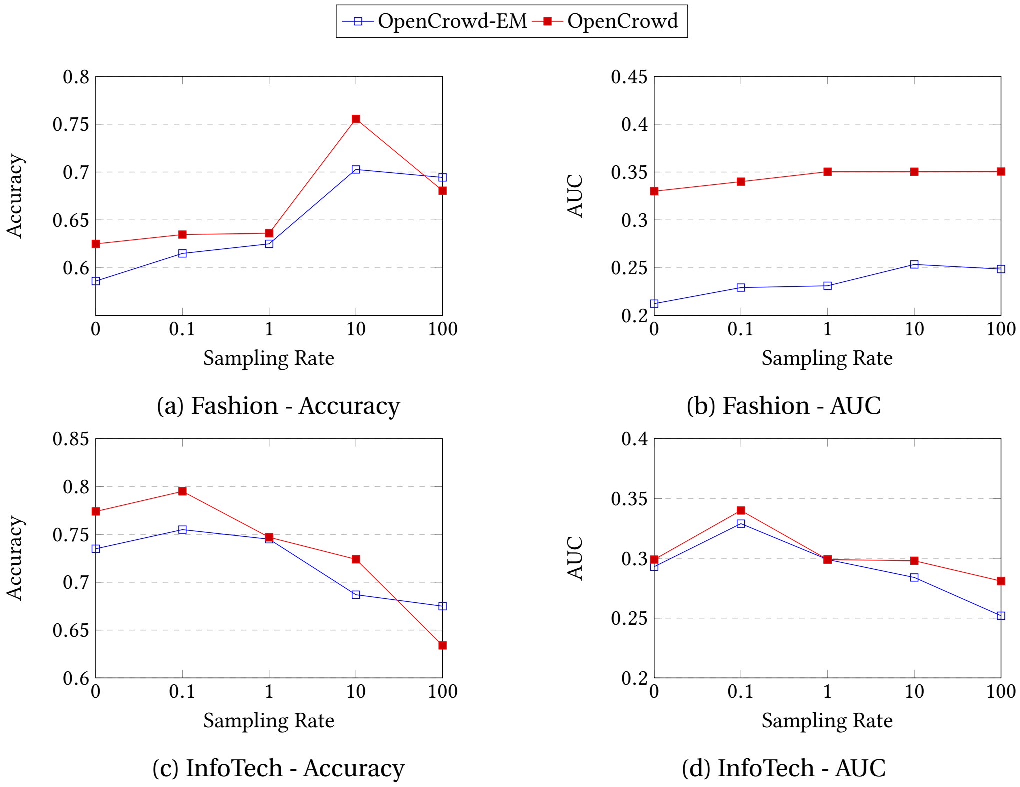
Most importantly, we observe that Multi. OpenCrowd overall achieves the best performance and outperforms the second best method (BCCWords MLP) on average on all supervision rates by 1.55% accuracy and 2.18% AUC. The main difference between Multi. OpenCrowd and BCCWords MLP is a worker's model where the latter uses a confusion matrix while we use a single parameter. This improvement is likely due to the sparsity of our worker-answer matrix, which can easily lead to over-fitting for more expressive models.

3.5.5 Properties of OpenCrowd

The comparison between OpenCrowd variants against feature-based methods (see "Comparison Methods") is shown in Figures 3.4(a,b). OpenCrowd-EM outperforms both LR and MLP by 18.25% and 5.82% accuracy and by 13.69% and 18.05% AUC, respectively. These results show the importance of considering worker reliability in aggregating workers' answers. Among the two variants, OpenCrowd outperforms OpenCrowd-EM by 7.37% accuracy and 31.62% AUC. This result indicates that modeling the worker reliability as a latent variable with a prior distribution not only makes the model more robust, but also improves the aggregation performance.

Impacts of Sampling Rate.

The sampling rate s_rate controls the size of randomly sampled candidate influencers in estimating the workers' reliability. The results are shown in Figure 3.7. We observe that, as the sampling rate increases from 0 to 100, the performance first increases then decreases. Such a result is consistent on both datasets, measured by both accuracy and AUC. The optimal per-



(a) Fashion - Accuracy

(b) Fashion - AUC

(c) InfoTech - Accuracy

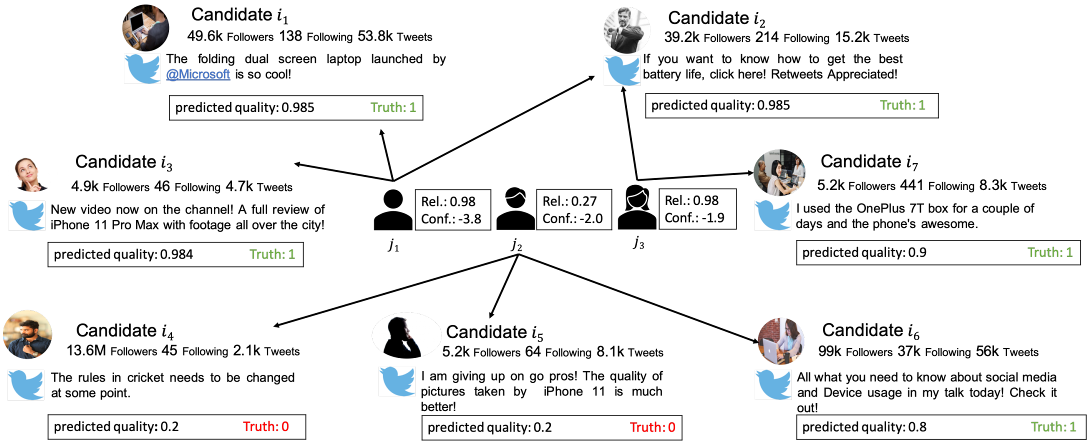
(d) InfoTech - AUC

Figure 3.7: Performance of OpenCrowd with varying s_rate.

formance is reached for $s\_rate = 10$ for Fashion and $s\_rate = 0.1$ for InfoTech, indicating that workers' evaluation on candidate influencers they do not name is more negative on Fashion. Overall, the variation of the performance with different s_rate indicates the importance of selecting the optimal sampling rate. The similarity in performance variation across the two datasets again demonstrates the robustness of OpenCrowd.

Interpretation of Learning Results.

Results of OpenCrowd can be explained in terms of the social features of candidate influencers and of the correlation between worker answers. We show in Figure 3.8 the learning results of real-world examples for three workers and seven candidate influencers from the InfoTech dataset. We also show the mean and confidence (differential entropy [128]) of worker reliability distribution (r_j), the predicted quality (θ_i) and the ground-truth labels of candidate influencers. We observe that workers who name real influencers have a high reliability as inferred by OpenCrowd, and otherwise have a low reliability. For example, the three influencers named by worker j_1, who has the highest reliability, are all real influencers. Among them, candidates i_1 and i_2 clearly exhibit influencer characteristics, e.g., they have a large number of followers and tweets dedicated to InfoTech. These results indicate that our approach is able to correctly infer the reliability of workers by leveraging the social features of the candidate influencers



The folding dual screen laptop launched by @Microsoft is so cool!

If you want to know how to get the best battery life, click here! Retweets Appreciated!

New video now on the channel! A full review of iPhone 11 Pro Max with footage all over the city!

I used the OnePlus 7T box for a couple of days and the phone's awesome.

The rules in cricket needs to be changed at some point.

I am giving up on go pros! The quality of pictures taken by iPhone 11 is much better!

All what you need to know about social media and Device usage in my talk today! Check it out!

Figure 3.8: Examples of workers and their nominated candidate influencers in the InfoTech domain. We show for workers the inferred reliability (Rel.) and inference confidence (Conf.), and for candidate influencers the predicted quality as well as the ground-truth labels. Profile pictures are from public data sources and are randomly assigned to anonymize user identities.

they named. Thanks to the worker's reliability, micro-influencers with a smaller number of followers, such as candidate i_3, can also be successfully detected by our approach. We also observe that worker j_3 has the same high reliability as j_1, despite the fact that only one of the candidates (i_2) she named exhibits influencer characteristics. This is because OpenCrowd leverages the correlation between worker answers in reliability inference: i_2 is named by both workers j_1 and j_3. The difference between the number of answers provided by j_1 and j_3 is captured through the confidence measure: i_1 has a higher confidence than i_3. Most importantly, we observe that the high reliability inferred for j_3 helps to detect an additional micro-influencer that she named, i.e., i_7. These results demonstrate that OpenCrowd can find micro-influencers through reliable workers, whose reliability can be inferred either through further named candidate influencers or through similar workers.

3.6 Conclusion

In this chapter, we have presented OpenCrowd, a unified Bayesian framework that seamlessly incorporates machine learning and crowdsourcing for social influencer identification. Our framework aggregates open-ended answers while modeling both the quality of the workers' answers and their reliability. We derived a principled optimization algorithm based on variational inference with efficient incremental update rules for learning OpenCrowd parameters. Extensive validation on two real-world datasets shows that OpenCrowd is an effective and robust framework that substantially outperforms state-of-the-art answers aggregation methods. Results further show that our framework is particularly useful in finding micro-influencers by exploiting the social features and the correlation between worker answers.

The crowdsourcing framework used in OpenCrowd can only evaluate open-ended answers on a single criterion. Consequently, our method can only be executed on tasks where a single dimension is assessed. Moreover, workers' contribution to OpenCrowd consists of providing open-ended answers but does not interfere with their evaluation. In the next chapter, we will investigate a technique that will allow human contributors to assess open-ended answers via peer grading and evaluate them on many dimensions.

4 Peer Grading the Peer Reviews: A Dual-Role Approach for Lightening the Scholarly Paper Review Process

Open-ended answers are omnipresent in almost all domains. In the scientific domain, for instance, researchers are often solicited to assess each other's scholarly papers and provide feedback in a written review. In this context, the scholarly reviews can be considered open-ended answers as they are provided in a free-text form and answer the question: What are the paper's strengths and weaknesses?

Due to the rapid increase of scholarly paper submissions, conferences recruit many reviewers with different levels of expertise and background to guarantee a minimum number of reviews per paper. Unfortunately, with such a scale, the submitted reviews often do not meet the conformity standards of the conferences. Such a situation poses an ever-bigger burden on the meta-reviewers when trying to reach a final decision.

In this chapter, we propose a human-AI approach that estimates the conformity of reviews to the conference standards. Specifically, we ask peers to grade each other's reviews anonymously concerning important criteria of review conformity such as sufficient justification and objectivity. We introduce a Bayesian framework that learns the conformity of reviews from the peer grading process, historical reviews, and conference decisions while considering grading reliability. Our approach helps meta-reviewers easily identify reviews that require clarification and detect submissions requiring discussions while not inducing additional overhead from reviewers. Through a large-scale crowdsourced study where crowd workers are recruited as graders, we show that the proposed approach outperforms machine learning or review grades alone and can be easily integrated into existing peer review systems.

4.1 Introduction

Peer review is the standard process of evaluating the scientific work of researchers submitted to academic journals or conferences. An essential task in this process comes at the end when the meta-reviewers have to make a decision as to accept a paper or not. Recently, peer review has been challenged by the rapid increase of paper submissions. Consider the example of computer science conferences: The Conference on Neural Information Processing Systems

(NeurIPS) and the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) received 9467 and 6656 submissions in 2020, respectively; the numbers are five times the number of submissions they received in 2010.

To guarantee a minimum number of reviews per paper, those conferences recruit a large number of reviewers of different expertise levels and background. For example, due to the very high number of submissions, some conferences decided to lift the restriction of having published papers in former editions of the same venue to be part of the reviewing board [37]. The submitted reviews do not always meet the conformity standards of the conferences such as the presence of sufficient justification for the claims, the validity of argumentation (e.g., not self-contradictory), and the objectivity of comments. Such a situation poses an ever-bigger burden on the meta-reviewers, who not only have to handle more papers and reviews, but also have to carefully validate the reviews in terms of the conformity to the review standards. For instance, in the NeurIPS example we cite above, each meta-reviewer had to handle up to 19 submissions with around 76 reviews total.

The load could be reduced if we were able to develop methods to automatically detect low-conformity reviews. The need has been explicitly discussed recently by program chairs of the ACM SIGMOD conference [3]: "The chairs discovered low-confidence reviews manually; such reviews, however, should be flagged automatically to allow for immediate action", "automated analysis of the reviews as they come in to spot problematic text ... could dramatically alleviate the overhead that chairs and meta-reviewers endure while trying to detect the problem cases manually". We note that computational methods have provided strong support to streamline several parts of the peer review process, such as those for paper assignment to reviewers [57, 85, 186, 119, 94], finding expert reviewers [51, 61, 129], and reviewer score calibration [64, 131, 5]; however, relatively little work can be found on developing computational methods for detecting low-conformity reviews.

Automatic detection of low-conformity reviews is nontrivial for two main reasons. First, the task is highly complex and requires to assess reviews from a multitude of dimensions [72, 1, 183, 152, 170] including justification, argumentation, objectivity, etc. Assessment on those dimensions is cognitively demanding as it requires to comprehend the review text to understand the various relations among its statements. Second, submission and review information of most conferences are not openly accessible for privacy and confidentiality concerns. This lack of training data limits the performance of existing natural language processing techniques.

To tackle these challenges, we advocate a human-AI collaborative approach for the semi-automatic detection of low-conformity reviews. We involve peer reviewers to grade each other's reviews anonymously with respect to important criteria of review conformity. Simultaneously, a machine learning model joins the assessment for less ambiguous reviews while learning from new peer grading to make connections between the review features and their conformity level. The main advantage of involving machine learning is that the model encapsulates and accumulates human knowledge of review conformity over time: what it learned in the previous editions of a conference can be used for a new edition by simply applying the model to new reviews. Over time, the model improves and the human-AI approach requires less amount of grading from humans to detect low-conformity reviews. The "peer grading peer reviews" mechanism does not disrupt current peer review process: reviewers of the same paper are supposed to read each other's reviews and make adjustments to their own reviews whenever necessary. Making such an explicit step by asking them to grade each other's reviews can potentially stimulate reviewers to be more engaged and promote the quality of the discussion thereafter. Our proposed mechanism is, therefore, a lightweight add-on to the current peer review systems without inducing much extra effort from the reviewers.

At the technical level, we introduce a Bayesian framework that seamlessly integrates machine learning with peer grading for assessing review conformity while allowing the model to learn from peer grading. An important consideration of our framework design is that it models the reliability of the graders, thus taking into account the effect of their various background and expertise levels. To learn the reliability and the parameters of the machine learning model, we derive a principled optimization algorithm based on variational inference. In particular, we derive efficient updating rules that allow both model parameters and grader reliability to be updated incrementally at each iteration. By doing so, both types of parameters can be efficiently learned with little extra computational cost compared to the computational cost for training a machine learning model alone.

To evaluate our proposed approach, we first conduct a small-scale online experiment with real expert reviewers, where we simulate the real peer review process with peer grading. We evaluate the effectiveness of peer grading by taking into account the grading as a weight of the reviewers' recommendation scores in the aggregation and we show that the aggregated score is a better approximation of the meta-decisions as compared to existing aggregation methods, e.g., average or weighted average by self-reported confidence. The number of expert grading is, however, not sufficient for evaluating proposed Bayesian framework. Inspired by the positive results of worker performance in judging the relevance of both scientific papers and search results to specific topics [24, 97], we conduct a larger-scale crowdsourcing study where we collect worker grading to approximate expert grading. We then use worker grading to evaluate our framework on the dataset we collected from the ICLR conference over a three-year time period, which allows us to observe the gradual model improvement over time.

In summary, we make the following key contributions:

- We propose a new dual-role mechanism called "peer grading peer reviews" to lighten the review process. Our approach can be easily integrated into current scholarly peer review systems;
- We introduce a Bayesian framework that integrates a machine learning model with peer grading to collaboratively assess the conformity of scholarly reviews while allowing the model to improve over time;

- We conduct a longitudinal evaluation of our framework across multiple years of a conference, showing that our method substantially improves the state of the art by 10.85% accuracy and that the model improves by 6.67% accuracy over three years.

4.2 Related Work

In this section, we first discuss the state of the art in peer reviewing, then review existing work methodologically related to our framework in review assessment and peer grading.

4.2.1 Scientific Peer Review

In the following, we discuss two relevant topics: computational support for scientific peer review and biases in reviews. State-of-the-art tools from artificial intelligence are making inroads to automate parts of the peer-review process [153]. A typical example is automatic paper assignment to appropriate reviewers. The problem has been formulated as an information retrieval problem [57, 75, 129, 86], where a paper to be assigned is a "query" and each review is represented as a document (e.g., an expertise statement or publications of the reviewer). This problem has also been formulated as matching problem, where the goal is to match a set of papers with reviewers under a given set of constraints, like workload, interest, and conflicts-of-interest [85, 186, 119, 94, 81]. Another important topic is finding expert reviewers. The task generally relies on automatic content analysis of textual documents (e.g., academic publications) and scientometrics (e.g., number of grants and patents), as well as link analysis based on cross-references between documents [51, 61, 129]. Apart from those, work has also been devoted to developing methods for identifying sentiments in reviews [207] and for predicting rebuttal results [66]. Recently, a pre-trained language model SciBERT has been introduced for modeling text in scientific publications [20].

Compared to the large body of work on those problems, relatively little effort can be found on developing automatic tools for review conformity assessment. Recent discussions have pointed to problems in low-conformity reviews, where reviewers can exhibit bias or only support expected, simple results, or ask for unnecessary experiments [3, 21, 153, 4, 58, 19].

Among those problems, biases in reviews is the most extensively studied topic. An important source of review biases comes from the setup of the review process being single- or double-blind. Snodgrass [180] reviews over 600 pieces of literature on reviewing, summarizing the implications of single- and double-blind reviews on fairness, review quality, and efficacy of blinding. In particular, the author points out the significant amount of evidence showing review biases in a single-blind setup, favoring high-prestigious institutions and famous authors. A more recent study by Tomikins et al. [192] through a controlled experiment on the ACM WSDM conference confirms such a finding. Another important source of bias is varying standards of reviewers in providing recommendations. A recent analysis by Shah et al. [171] over the reviews of the Neurips conference finds that the fraction of papers receiving scores over a threshold is not aligned with the meaning of the threshold defined by the conference. For example, nearly 60% of scores were above 3 despite the fact that the reviewers were asked to give a score of 3+ only if the paper lies in the top 30% submissions. This leads to the frustration of many authors whose papers get rejected despite receiving good scores.

Compared to those studies on review biases, other aspects of low-conformity reviews are much less discussed such as the lack of justification for decisions and of arguments. We show in Section 4.4 through an online survey that the lack of justification for arguments and decisions is most often due to low-conformity reviews, which increases the complexity of the meta decisions and, if not handled well, lower the authors' trust in the venue. We envision that automatic methods for low-conformity reviews detection can significantly reduce this issue, similar to what automatic methods for paper-reviewer assignment achieved in the past decades. Our work makes a first attempt along this direction, providing a first-of-it-kind human-in-the-loop AI method that leverages both human and machine intelligence in determining review conformity.

4.2.2 Review Assessment and Peer Grading

In the design of our approach, we draw inspiration from existing methods for review assessment and peer grading, developed in different domains. Methods for review assessment have been mainly developed for e-commerce and online rating platforms. Olatunji et al. [139] propose a convolutional neural network with a context-aware encoding mechanism to predict the product reviews' helpfulness based on the review text. Zhang et al. [232] study the problem of predicting the helpfulness of answers to users' questions on specific product features. Their model is based on a dual attention mechanism to attend the important aspects in QA pairs and common opinions reflected in the reviews. These methods rely in their core on pre-trained language models such as Glove [149] or ALBERT [104]. These language models are trained on massive and heterogeneous corpora to capture text semantics, which provide useful information for review classification. Prediction for scholarly reviews is more challenging than for other types of reviews due to both the cognitive complexity of the task, the highly specialized topic, and the lack of available datasets for model training. Unlike those fully automatic methods, we consider the role of humans (i.e., peers) in our approach as indispensable, as we show in our experiments.

Methods for peer grading have been mainly developed for (online) education and crowdsourcing platforms. In the educational context, Wang et al. [208] study the phenomenon of students dividing up their time between their own homework and grading others from a game theory perspective. Crowd workers have been used to simulate the role of students and to assess homework quality. Mi et al. [126] propose a probabilistic graphical model to aggregate peer grading. Their method considers an online course setup and models both the student and the grader's reliability, imposing a probabilistic relationship between the reliability of a student and the true grade. Carbonara et al. [34] model the peer grading process in MOOCs as an audit game where students play the role of attackers and the course staff play defenders. In the context of crowdsourcing, Labutov et al. [99] propose a framework that fuses both task execution and grading. They adopt an Expectation Maximization algorithm to aggregate the grading by inferring both worker's reliability and task difficulty. From a methodological perspective, our framework is different from those aforementioned methods in that we take a human-AI approach that integrates peer grading and a supervised machine learning model, which is important for both improving the accuracy of review conformity and for reducing manual efforts.

4.3 The PGPR Framework

In this section, we introduce our proposed Bayesian PGPR framework that learns to predict the conformity of reviews from a few peer-graded reviews as well as from historical data (reviews and decisions) of a given venue. We first formally define our problem and then describe our overall framework, followed by our variational inference algorithm for learning PGPR parameters.

4.3.1 Notations and Problem Formulation

Notations

Throughout this chapter, we use boldface lowercase letters to denote vectors and boldface uppercase letters to denote matrices. For an arbitrary matrix $\boldsymbol{M}$, we use $\boldsymbol{M}_{i,j}$ to denote the entry at the i-th row and j-th column. We use capital letters (e.g., $\mathcal{P}$) in calligraphic math font to denote sets and $|\mathcal{P}|$ to denote the cardinality of a set $\mathcal{P}$.

Table 4.1 summarizes the notations used throughout this chapter. We denote the set of reviews with $\mathcal{I}$ and the set of graders as $\mathcal{G}$. We restrict $\mathcal{I}$ to include only the graded reviews without ground truth of conformity – our framework can be initialized with any number of reviews with ground truth, thereby utilizing historical data (see Section 4.3.4). For each review $i \in \mathcal{I}$, we extract a set of features as described in detail in Section 4.5.1 and denote the resulting vector by $\boldsymbol{x}_i$. We use $\boldsymbol{A}_{i,g}$ to denote the grade given by grader $g \in \mathcal{G}$ when reviewing $i \in \mathcal{I}$. Due to the fact that an individual grader can only grade a limited number of reviews, $\boldsymbol{A}$ is a sparse matrix where only a small proportion of the entries are known.

Problem Definition

Let $\mathcal{I}$ be the set of reviews, where each review $i \in \mathcal{I}$ is represented by a feature vector $\boldsymbol{x}_i$. Let $\boldsymbol{A}$ be the grader-review matrix where each element $\boldsymbol{A}_{i,g}$ is a grade given by a grader $g \in \mathcal{G}$ to a review i. Our goal is to infer the conformity score z_i for all reviews $i \in \mathcal{I}$ using $\boldsymbol{x}_i$ and $\boldsymbol{A}$.

Table 4.1: Notations.

Notation	Description
$\mathscr{I}$	Set of reviews
$\mathscr{G}$	Set of graders
$\boldsymbol{A}$	Grader-Review matrix
$\boldsymbol{x}_i$	Feature vector of a review
z_i	Review conformity distribution
r_g	Grader reliability distribution
b_g	Grader bias distribution
μ_i, σ_i	Parameters of the review conformity distribution
A_g, B_g	Parameters of the distribution of grader reliability
m_g, α_g	Parameters of the distribution of grader bias

4.3.2 PGPR as a Bayesian Model

PGPR is a unified Bayesian framework that integrates a machine learning model –modeling review conformity from features– with peer grading for predicting review conformity. Once trained, the machine learning part of PGPR can be used alone to predict conformity of reviews without peer grading.

The overall framework is depicted as a graphical model in Figure 4.1. It models review conformity from both the features (through the machine learning model) and peer grading, which is modeled as a process conditioned on the review conformity and grader properties (i.e., reliability and bias). In the following, we first describe how a machine learning model is embedded into PGPR and then describe the grading process and its integration into our framework.

Learning Conformity

We model review conformity z_i with a Gaussian distribution:

$$z_i \sim \mathscr{N}(\mu_i, \sigma_i), \tag{4.1}$$

where μ_i and σ_i are the mean and the variance of the distribution, respectively. μ_i is predicted from the review features $\boldsymbol{x}_i$ through a neural network of arbitrary architecture.

$$\mu_i = \text{softmax}(f^{\mathscr{W}}(\boldsymbol{x}_i)), \tag{4.2}$$

where the function $f^{W}(\boldsymbol{x}_i)$ models the output of the network layers preceding the softmax layer, parameterized by $\mathscr{W}$ shared across all reviews. The variance σ_i of the Gaussian distribution is automatically learned through our inference algorithm (described in Section 4.3.3). Unlike normal supervised settings, we do not have the ground truth of review conformity μ_i; instead, we are given a set of review grades, which we model next.

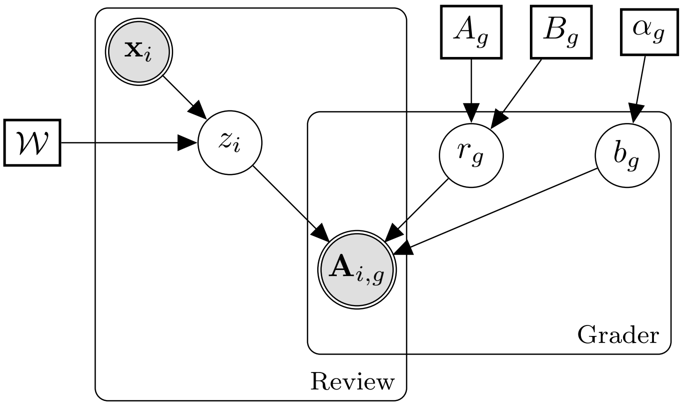


Figure 4.1: Graphical representation of PGPR. Double circles represent observed variables, while single circles represent latent variables. Squares represent model parameters. Edges represent conditional relationships in text classification. On the left-hand side, a machine learning model parameterized by $\mathscr{W}$ predicts the conformity z_i of a review. Each review is represented with a feature vector $\boldsymbol{x}_i$. On the right-hand side, a grader is represented with her reliability distribution r_g with parameters A_g and B_g and her bias b_g with α_g as a prior. The grader assigns a review with grade $\boldsymbol{A}_{i,g}$.

Modeling Review Grades

We model the grading process by considering two important properties of graders, namely reliability and bias. In practice, we would like to have a measure of *confidence* in estimating the reliability and bias of the graders grading different numbers of reviews: we should be more confident in estimating the reliability and bias of graders who grade 50 reviews than those who grade 5 reviews only. To quantify the confidence in our inference, we adopt a Bayesian treatment when modeling both grader properties by introducing prior distributions.

Specifically, we denote the grader reliability by r_g ($g \in \mathscr{G}$) and model it with a Gamma distribution: a higher value indicates a better ability to provide accurate grades.

$$r_g \sim \Gamma(A, B), \tag{4.3}$$

We consider grader bias as the tendency of a grader to give high or low conformity scores to reviews. We denote the grader bias by b_g ($g \in \mathscr{G}$) and model it using a Gaussian distribution.

$$b_g \sim \mathcal{N}(m, \frac{1}{\alpha}). \tag{4.4}$$

Integrating Machine Learning with Peer Grading

We define the likelihood of a grader g giving a score $\boldsymbol{A}_{i,g}$ to review i as a probability conditioned on the grader's reliability r_g, the bias b_g, and the conformity of the review z_i.

$$p(\boldsymbol{A}_{i,g}|z_i, r_g, b_g) = \mathcal{N}(z_i + b_g, \frac{1}{r_g}) \tag{4.5}$$

The conditional probability in Eq. (4.5) formalizes the following intuitions: *i)* a grader with a bias $b_g > 0$ (or $b_g < 0$) is likely to overestimate (or underestimate) the conformity of a review, whereas a grader with a bias $b_g \approx 0$ has a more accurate estimation of review conformity; and *ii)* a grader with a high reliability r_g is likely to give a conformity score with a small deviation from the true conformity.

4.3.3 Variational Inference for PGPR

Learning the parameters of PGPR resorts to maximizing the following likelihood function:

$$p(\boldsymbol{A}) = \int p(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}|\boldsymbol{X}; \mathcal{W})\, d\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}, \tag{4.6}$$

where $\boldsymbol{z}$ is the latent conformity scores for all the reviews, and $\boldsymbol{r}$ and $\boldsymbol{b}$ are the latent reliability scores and biases for all graders. $\boldsymbol{X}$ represents the feature matrix of all reviews and $\mathcal{W}$ is the set of machine learning parameters.

Since Eq. (4.6) contains more than one latent variable, it is computationally infeasible to optimize [195]. Therefore, we consider the log of the likelihood function, i.e.,

$$\log p(\boldsymbol{A}) = \underbrace{\int q(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}) \frac{p(\boldsymbol{A}, \boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}|\boldsymbol{X}; \mathcal{W})}{q(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b})} d\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}}_{\mathcal{L}(\mathcal{W}, q)} + \underbrace{\int q(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}) \frac{q(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b})}{p(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}|\boldsymbol{A}, \boldsymbol{X}; \mathcal{W})} d\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}}_{KL(q||p)}, \tag{4.7}$$

where $KL(\cdot)$ is the Kullback Leibler divergence between two distributions. The log likelihood function in Eq. (4.7) is composed of two terms. Using the variational expectation-maximization algorithm [195], we can optimize the objective function iteratively in two steps: 1) the E-step, where we minimize the KL-divergence to approximate $p(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}|\boldsymbol{A}, \boldsymbol{X}; \mathcal{W})$ with the variational distribution $q(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b})$; and 2) the M-step, where we maximize the first term $\mathcal{L}(\mathcal{W}, q)$ given the newly inferred latent variables. In the following, we describe both steps.

E-step Using the mean-field variational inference approach [26], we assume that $q(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b})$ factorizes over the latent variables:

$$q(\boldsymbol{z}, \boldsymbol{r}, \boldsymbol{b}) = \prod_{i \in \mathscr{I}} q(z_i) \prod_{g \in \mathscr{G}} q(r_g) \prod_{g \in \mathscr{G}} q(b_g). \tag{4.8}$$

To minimize the KL divergence, we choose the following forms for the factor functions:

$$q(z_i) = \mathscr{N}(\mu_i, \sigma_i), q(r_g) = \Gamma(A_g, B_g), q(b_g) = \mathscr{N}(m_g, \frac{1}{\alpha_g}), \tag{4.9}$$

where $\mu_i, \sigma_i, A_g, B_g, m_g, \alpha_g$ are variational parameters used to perform the optimization and minimize the KL-divergence.

In the following, we give the update rules for each of the latent variables. We first give the update rules for review conformity z_i by the following lemma.[1]

Lemma 3 (Incremental Update for Review Conformity)**.** *The conformity distribution $q(z_i)$ follows a Gaussian distribution and can be incrementally computed using the grade, the grader reliability, and the review conformity from the previous iteration:*

$$q(z_i) \sim \mathscr{N}(\frac{W}{V}, \frac{1}{V}), \tag{4.10}$$

where:

$$\begin{cases} W = \sum_g \frac{A_g}{B_g}(\boldsymbol{A}_{i,g} - m_g) + \frac{\mu_i}{\sigma_i^2}, \\ V = (\sum_g \frac{A_g}{B_g} + \frac{1}{\sigma_i^2}). \end{cases}$$

Next, we show the updating rules of grader's reliability and bias.

Lemma 4 (Incremental Update for Grader Reliability)**.** *The update of the grader reliability $q(r_g)$ follows a Gamma distribution with parameters that can be incrementally updated using the conformity of reviews she graded, her bias and her reliability from the previous iteration:*

$$q(r_g) \sim Gamma(X, Y), \tag{4.11}$$

where:

$$\begin{cases} X = A_g + \frac{|\mathscr{I}_g|}{2}, \\ Y = B_g + \frac{1}{2}(\frac{|\mathscr{I}_g|}{\alpha_g} + \sum_i [\boldsymbol{A}_{i,g}^2 + \sigma_i^2 + 2\mu_i(m_g - \boldsymbol{A}_{i,g}) - 2\boldsymbol{A}_{i,g} m_g]). \end{cases}$$

Lemma 5 (Incremental Update for Grader Bias)**.** *The bias of the graders $q(b_g)$ follows a Gaussian distribution with parameters that can be incrementally updated using the review confor-*

[1] Proofs for all the lemmas are given in the appendix.

```
Algorithm 3: Learning PGPR Parameters
Input  :Grader-Review matrix $\boldsymbol{A}$, Review features matrix $\boldsymbol{X}$
Output :Parameters of the PGPR framework:
        $\mu_i, \sigma_i, A_g, B_g, m_g, \alpha_g, \mathscr{W}$
Initialize PGPR parameters ;
while $\log p(\boldsymbol{A})$ has not converged do
    for $i \in \mathscr{I}$ do
        update $q(z_i)$ using Lemma 3;
    for $g \in \mathscr{G}$ do
        update $q(r_g)$ using Lemma 4;
        update $q(b_g)$ using Lemma 5;
    for $i \in \mathscr{I}$ do
        Update $\mathscr{W}$ using back-propagation;
```

mity, the grader reliability and her bias from the previous iteration:

$$q(b_g) \sim \mathscr{N}(\frac{L}{K}, \frac{1}{K}), \tag{4.12}$$

where:

$$\begin{cases} K = \frac{A_g|\mathscr{I}_g|}{B_g} + \alpha_g, \\ L = \alpha_g m_g + \frac{A_g}{B_g}\sum_i(\boldsymbol{A}_{i,g} - \mu_i). \end{cases}$$

M-step Given the conformity of a review, the grader reliability and bias inferred in the E-step, the M-step maximizes the first term of Eq. (4.7) to learn the parameter $\mathscr{W}$ of the machine learning model:

$$\begin{aligned} \mathscr{L}(\mathscr{W}, q) &= \int q(z_i, r_g, b_g) \log p(\boldsymbol{A}_{i,g}, z_i, r_g, b_g|\boldsymbol{x}_i; \mathscr{W})\, dz_i, r_g, b_g + C \\ &= \int q(z_i, r_g, b_g) \log[p(\boldsymbol{A}_{i,g}|z_i, r_g, b_g) p(z_i|\boldsymbol{x}_i; \mathscr{W})]\, dz_i, r_g, b_g + C \\ &= \underbrace{\int q(z_i, r_g, b_g) \log p(\boldsymbol{A}_{i,g}|z_i, r_g, b_g)\, dz_i, r_g, b_g}_{\mathscr{M}_1} + \underbrace{\int q(z_i) \log p(z_i|\boldsymbol{x}_i; \mathscr{W}) dz_i}_{\mathscr{M}_2} + C \end{aligned} \tag{4.13}$$

where $C = \mathbb{E}_{q(z_i, r_g, b_g)} log(\frac{1}{q(z_i, r_g, b_g)})$ is a constant. Only the second part of $\mathscr{L}(\mathscr{W}, q)$, i.e., $\mathscr{M}_2$, depends on the model's parameters. $\mathscr{M}_2$ is exactly the inverse of the cross-entropy between $q(z_i)$ and $p(z_i|\boldsymbol{x}_i; \mathscr{W})$, which is widely used as the loss function for many classifiers. $\mathscr{M}_2$ can, therefore, be optimized using back-propagation.

4.3.4 Algorithm

The overall optimization algorithm is given in Algorithm 3. We start by initializing the parameters of each probability distribution and of the machine learning model. Then, we iterate between the E step (rows 3-7) and the M step (rows 8-9). The E step consists of updating the variational distributions of the review conformity $q(z_i)$, the grader reliability $q(r_g)$ and her bias $q(b_g)$. The M step consists of updating the parameters $\mathscr{W}$ of the machine learning model using back-propagation. The convergence is reached when the review conformity $q(z_i)$ is no longer modified by the grader reliability and bias. Note that when some reviews with ground truth conformity are available, the machine learning model can be trained first to obtain an initialization of $\mathscr{W}$, which will then be updated further by Algorithm 3. Once the learning algorithm terminates, the machine learning model of PGPR can be taken out to assess the conformity of any review.

The iterations in rows 3-4 require a time complexity of $|\mathscr{I}|$ and the iterations through all graders yield a time complexity of $|\mathscr{G}|$. The overall complexity of our algorithm is $O(\#iter(|\mathscr{I}|+|\mathscr{G}|+\mathscr{C}_W)$ where $\#iter$ is the total number of iterations needed until convergence and $\mathscr{C}_W$ is the complexity to learn the parameters of the machine learning model.

4.4 Task Design for Grading Reviews

In this section, we present our design for the review grading task, which is used to collect data for evaluating our proposed framework. Due to the privacy concern, submissions and review information in most venues are not publicly available. Fortunately, we have access to such an information in two venues, on which we conduct a small-scale experiment with expert reviewers to evaluate the effectiveness of peer grading in measuring review conformity. Evaluating our proposed PGPR framework, however, requires more grading than those we can collect from expert reviewers. We conduct a larger-scale crowdsourcing study, in which we collect worker grading to approximate the grading from expert reviewers and use those grading for evaluating PGPR.

This section focuses on the task design of grading reviews for both expert and crowd scenarios. We present an analysis on the effectiveness of grading from both expert reviewers and crowd workers in the next section. In the following, we first identify a set of criteria for review conformity assessment and then describe the setup of the grading task.

4.4.1 Criteria for Review Conformity

We compile a list of eight criteria for review conformity from the literature, a set of review guidelines published by journals and conferences [72, 1, 152], and guidelines from publishers such as Springer [183] or Nature Research [133]. Those criteria are grouped into the following three categories.

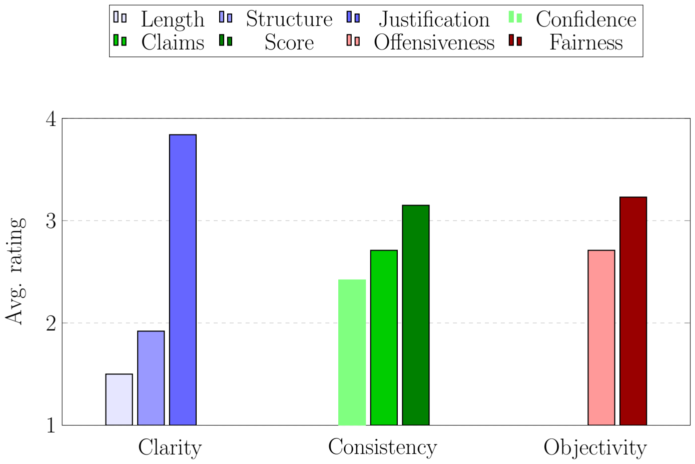


Figure 4.2: Ranking of review criteria.

- **Clarity.** The clarity of a review resides in three main aspects. 1) *Structure*: it is often imposed that the review should contain a summary of the paper, the decision, and supporting arguments for this decision. 2) *Length*: a review should be of adequate length to provide sufficient information for the meta-reviewer to understand the reviewer's recommendation [140]. 3) *Justification*: a review should include supporting arguments of the decision, by including pointers to prior work as well as references to specific parts of the paper on which the score is based [72].

- **Consistency.** The consistency of a review is defined by three aspects. 1) *Score*: the recommended score should be supported by at least one or two justifications. 2) *Claims*: there should be no contradiction between the summary and the stated weak or strong claims. 3) *Confidence*: the reviewer should make a clear acknowledgement when certain aspects of a paper are beyond her expertise [72].

- **Objectivity.** A review should be fair and provide constructive critiques. 1) *Fairness*: A review should not be biased towards irrelevant factors such as assigning a low score because of missing references from the reviewer's own work only[2]. 2) *Offensiveness*: A review should cover the technical work rather than giving personal statements and/or offensive terms [72].

To understand the importance of those criteria, we initially conducted an online survey with 38 expert reviewers from two international venues: SEMANTICS (SEM) (2019 edition) and the International Workshop on Decentralizing the Semantic Web (DSW) (2017 and 2018 editions). We asked the expert reviewers to rate the importance of each individual criterion and the three categories on a 5-point Likert scale and show the results in Figure 4.2. We observe that clarity ranked the highest (by 28 expert reviewers) and, in particular, that justification is viewed as the most important aspect of a high-conformity review. Consistency is equally important to

[2] https://www.seas.upenn.edu/ nenkova/AreaChairsInstructions.pdf

objectivity. While many agree that objectivity is not a deterministic aspect, half of the experts admit having received or read unfair reviews while few have received offensive ones. In fact, 24 expert reviewers rank fairness as equally or more important than offensiveness. These results indicate that the experts consider review fairness as an important concern.

4.4.2 Task Design

For each of the reviews we consider in our work, we ask participants to provide ratings for each of the eight conformity criteria, grouped in three sections corresponding to the three categories introduced above. In the crowdsourcing scenario, we recruit from Amazon MTurk workers with a "Master" qualification, i.e. workers who have demonstrated high degree of success in performing a wide range of tasks across a large number of requesters. The task starts by explaining how a scholarly review is presented, the criteria (on a category level), followed by a positive and a negative example. Then, we show workers a review and ask them to rate each criterion from 1 to 4 with 4 being the best rating. We set the range to be 1-4 instead of 1-5 as we found in a preliminary study that workers tend to favor 3 in the latter case. Each rating question is accompanied with an information box that explains the aspect to rate. For questions regarding justification, fairness and offensiveness, we ask workers to provide a snippet from the review as a rationale justifying their grading decision [125]. The rationale can be used as an explanation for the conformity score assigned to the review. For attention check, we ask workers to identify the recommendation decision from the review; results of workers who fail at recognizing review decisions are excluded. After getting their ratings, we ask the workers to enter feedback in free text. Each review is rated by three different workers. The task takes approximately 12 min to complete. Workers who completed the task received a reward of 1.8 USD.

In the expert scenario, the task is simplified to include only the rating for each of the criteria. The peer grading of scholarly reviews is implicit in the current peer review systems: each reviewer is supposed to read the reviews from other reviewers and decide whether to keep her original recommendation or not; however, they are typically not required to express their opinion about other reviews explicitly. We assume explicit peer grading can stimulate reviewers to look into other reviews and promote the quality of the discussions afterwards. We show in the next section through an experiment with real expert reviewers that the peer grading is effective when used to weight the reviewers' recommendation scores in score aggregation, which approximates meta-decisions better than existing aggregation methods, e.g., weighted average by reviewers' self-indicated confidence.

Table 4.2: Description of the ICLR Datasets. #Misalign. sub. is the number of submissions to which there is at least a review with decision misaligned with meta-decision; #Misalign. reviews is the overall number of not-aligned reviews.

Edition	#sub.	#Misalign. sub.	#Misalign. reviews
2017	506	169	530
2018	846	355	1072
2019	1565	670	2060

4.5 Experimental Results

This section presents the results of our empirical evaluation[3]. We first conduct a preliminary analysis to understand the effectiveness of expert and worker grading, then evaluate the performance of our PGPR framework by comparing it against the state of the art. Finally, we perform an in-depth analysis of PGPR's main properties. We answer the following questions:

- Q1: How effective is expert and worker grading in assessing review conformity? (Section 4.5.2).
- Q2: How effective is our proposed human-AI approach in predicting review conformity? (Section 4.5.3).
- Q3: How effective is our framework in leveraging peer grading compared to majority voting? (Section 4.5.4).
- Q4: How effective is peer grading in improving the conformity prediction over time when more reviews with ground truth decisions become available? (Section 4.5.5).

4.5.1 Experimental Setup

Datasets

We collect data from the ICLR conference, which provides open access to reviews and evaluation scores for all submissions through OpenReview[4]. We collected reviews for all submissions to the ICLR conference from 2017 until 2019. Our ICLR dataset contains in total 2917 submissions and 8838 reviews. 1194 papers have at least one review that is misaligned with the meta-decision. In our study, we are mainly interested in those cases as they require some additional effort when reaching a final decision. Key statistics on the collected dataset are reported in Table 4.2.

[3] Source code and data are available at https://github.com/eXascaleInfolab/pgpr.

[4] https://openreview.net/

Active Selection of Reviews for Grading

We leverage active learning to select a subset of the most informative reviews from the ICLR-2018 and 2019 datasets for grading: for each year, we apply the model trained in the previous year to all reviews in the current year, and select the reviews on which the model prediction is most uncertain (measured by the entropy of the predicted probability) for crowdsourcing. We select the top-30% (321) reviews and top-5% (103) reviews from ICLR-2018 and ICLR-2019, respectively, and show in our experiments that those numbers are sufficient for the model to converge to optimal performance. We refer to the selected reviews as “uncertain” reviews and the rest as “certain” ones. We investigate in our experiments the performance of PGPR on both categories as well as the impact of the number of graded uncertain reviews on model training. In total, we crowdsourced a subset of 444 reviews in 2018 and 2019 and collected 1093 grades from 64 crowd workers on those selected reviews.

Data Split

To simulate the real-world application of PGPR, we evaluate it on different editions of the ICLR conference as follows: for each year (2018 or 2019), we assume the reviews and the ground truth from previous years are known, while for the current year only the reviews are available without the ground truth. For a subset of the reviews in the current year, we collect grading from workers. The training data, therefore, contains reviews and decisions from the previous years, and some reviews with crowd labels from the current year. We take reviews with the ground truth of the current year and equally split it into validation and test sets.

Label Extraction

We consider the ground truth of a review conformity as a binary variable indicated by the alignment between a reviewer decision and the meta-reviewer decision: when both the reviewer and the meta-reviewer decide to accept or reject a paper, the ground truth for the review is set to 1, otherwise to 0. Our model predicts for each review a value between 0 and 1 describing the probability of the review being conform. The higher the value, the higher the likelihood of the review to be conform. For the grades collected from crowd workers, we map it to the interval $[0,1]$ using the function $t(x) = (x-1)/4$, so that the range of valid grading matches the range of our model’s predictions.

Neural Architecture and Features.

The inputs of our machine learning model are hand-engineered features along with embeddings of the sentences in a review. For the hand-engineered features, we extract for each review the decision score, the confidence score, and their difference with the decision and confidence scores of the other reviews on the same paper. We also compute the review’s length, the number of citations within the review, and the number of keywords referring to a paper’s content (e.g., equation, section, figure). For the textual embeddings, we represent each sentence as a fixed-size vector by leveraging the pre-trained language model SciBERT [20]. These inputs are fed to the machine learning component of our framework consisting of a multi-input model we call "Mix-model". It includes both an attention-based model for the review's embedding and a logistic regression for the review's statistical features. We concatenate the output of the attention-based model and logistic regression and use a fully connected layer with tanh activation followed by a linear layer; the output is generated by a softmax function (Eq. 4.2).

Comparison Methods.

We compare our approach against the most applicable techniques for review's conformity assessment. We first compare against classification methods designed for the scholarly domain: 1) MILNET [207], a Multiple Instance Learning (MIL) neural model used to classify scholarly reviews (originally for sentiment analysis). 2) SciBERT [20], a self-attention-based neural language model pre-trained on scientific text consisting of publications from the computer science and biomedical domains. 3) DoesMR [66], a Logistic Regression model that takes hand-engineered features from scholarly reviews for prediction. In addition, we compare against models developed for non-scholarly review tasks, including a general-purpose language model and two models originally developed for predicting the helpfulness of product reviews: 4) ALBERT [104], a pre-trained language model for various NLP tasks, taking into account inter-sentence coherence to capture fine-grained information in documents including reviews. 5) PCNN [139], a convolutional neural model with context encoding. 6) RAHP [232], an attention-based model relying on a bidirectional LSTM to capture the sequential dependencies in text. For DoesMR, in addition to the original features, we include all features used by our method, such as the number of citations within the review and the number of keywords referring to a paper's content. All other methods use only textual data and hence cannot leverage hand-engineered features.

We also compare PGPR with its variant Mix-model that only consists of the machine learning component. Note that in Mix-model, the attention-based model used for the review's embedding is the same model used to evaluate SciBERT and the logistic regression used for the hand-engineered features is similar to DoesMR. All the comparison methods are trained using the same training data, i.e., historical reviews with decisions and new reviews with worker grading, which are aggregated by majority voting.

Parameter Settings.

For all the comparison methods, we tune the hyperparameters on the validation set. This includes the learning rate searched in {1e-5, 1e-4, 1e-3, 1e-2, 1e-1}, and the batch size in {8, 16, 32, 64}. For RAHP and PCNN, we vary the dimension of the embedding vector in {50, 100, 200, 300}. We train the models for a maximum of 500 epochs and take the versions that achieve the best performance on the validation set. For PGPR, after concatenating the output from

Table 4.3: Accuracy of approximating meta-decisions with average review scores and weighted average by self-reported confidence, expert grading, and worker grading.

Method	SEM	DSW	ICLR
Average	0.33	0.60	0.69
Confidence-weighted	0.50	*NA*	0.70
Grade-weighted (Experts)	0.83	0.80	*NA*
Grade-weighted (Workers)	*NA*	0.80	0.73

the attention-based model and logistic regression, we use a fully connected layer with tanh activation and ten neurons.

Evaluation Metrics

We measure the effectiveness of expert and worker grading in assessing review conformity by the accuracy of approximating meta-decisions with the grading-weighted average of reviewers' recommendation scores. Given a set of reviews $\mathscr{R}$ on the same paper, we denote the recommendation score of a review $r \in \mathscr{R}$ to the paper by s_r and the average grading the review receives by g_r. The aggregated score of $\mathscr{R}$ is given by:

$$s_{\mathscr{R}} = \frac{\sum_{r \in \mathscr{R}} g_r s_r}{\sum_{r \in \mathscr{R}} g_r}. \tag{4.14}$$

To measure the performance of PGPR and our baselines, we use accuracy, precision, recall and F1-score over the positive class. Higher values indicate better performance.

4.5.2 Preliminary Analysis on Peer Grading

We verify the effectiveness of peer grading on review conformity by expert reviewers and by crowd workers. We use the grading to weight reviewers' recommendation scores in score aggregation, and compare to other aggregation methods. We compute the accuracy of approximating meta-decision with the aggregation result.

Grading Reviews by Experts

For our first experiments, we select seven and five borderline papers from SEM and DSW, respectively. We only consider the borderline papers on which reviewers have some disagreement over their recommendations. Reviews from DSW papers are publicly available through OpenReview. For SEM, as the reviews are not publicly available, we contacted the reviewers to get their consent before sharing them with their peers. For both venues, we asked the original reviewers of the same paper to grade each other's reviews. 21 reviewers were involved for SEM providing one review each and 12 reviewers were involved for the DSW papers providing in

Table 4.4: Performance (Accuracy, Precision, Recall and F1-score) comparison with baseline methods. The best performance is highlighted in bold; the second best performance is marked by ‘*’.

Method	ICLR-2018				ICLR-2019			
	Accuracy	**Precision**	**Recall**	**F1-score**	**Accuracy**	**Precision**	**Recall**	**F1-score**
MILNET	0.533	0.580	0.770	0.660	0.528	0.560	0.860*	0.670
DoesMR	0.678*	0.710*	0.782	0.740*	0.747*	0.752*	0.838	0.792*
SciBERT	0.540	0.678	0.434	0.524	0.583	0.604	0.778	0.680
ALBERT	0.548	0.652	0.516	0.570	0.567	0.590	0.782	0.670
PCNN	0.523	0.624	0.508	0.562	0.516	0.570	0.645	0.605
RAHP	0.593	0.612	0.784*	0.688	0.501	0.570	0.515	0.540
PGPR	**0.781**	**0.822**	**0.810**	**0.810**	**0.799**	**0.770**	**0.917**	**0.840**

total 16 reviews. Results are shown in Table 4.3. We observe that the grade-weighted average of the reviews’ recommendations is better at approximating meta-decisions than other means of aggregating review scores. The result verifies that peer grading is a better indicator of review conformity than self-reported confidence scores and can be leveraged to better approximate meta-decisions than existing aggregation methods.

Grading Reviews by Crowd Workers

For this experiment, we use the DSW and ICLR datasets. We do not consider the reviews from SEM since those reviews are not public. Results are shown in Table 4.3. We observe that for both venues, the weighted average leveraging worker grading better approximates meta-decisions than the weighted average by self-reported confidence scores or the average without the weighting. Worker grading achieves comparable results to expert grading on DSW reviews. To further compare worker grading with expert grading in ICLR, we derive the peer grading according to the agreement between the reviews’ recommendation scores: the mutual grading between two reviewers is set to 4 if they gave the same score; if two reviewers have the same decision (e.g., an accept) with different scores, then we set their mutual grading to 3; if two reviewers have different decisions with a small difference between their scores (e.g., a weak accept and a weak reject), we set their mutual grading to 2; otherwise the mutual grading is set to 1. We calculate the average grading to the same review by workers and experts and observe that on 67% of the reviews, worker grading is similar to expert grading (difference < 1). We also observe that workers and experts have a higher agreement on assigning high grades rather than low ones and that workers tend to be more “generous" in grading reviews. Overall, those results are aligned with related work showing that crowd workers in carefully-designed tasks can provide satisfying outcomes on domain-specific problems [24, 97].

4.5.3 Comparison with the State of the Art

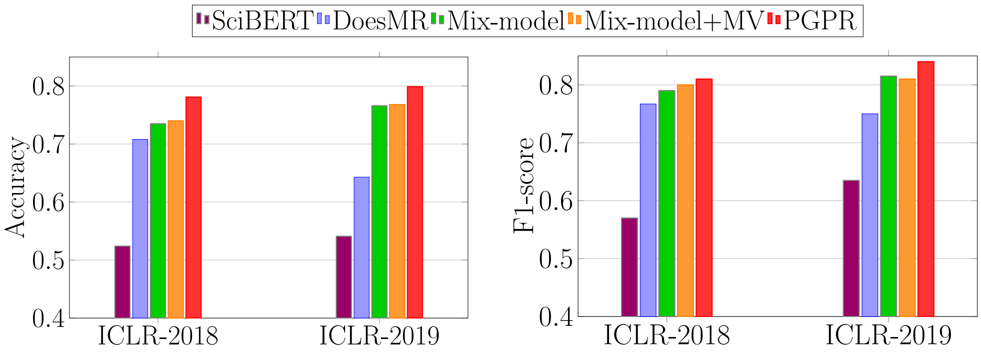
Table 4.4 summarizes the performance of PGPR against all the comparison methods on both ICLR-2018 and ICLR-2019. We make several observations.

First, we observe that among the comparison methods, DoesMR outperforms the other embedding or deep neural network models. Recall that DoesMR relies on hand-engineered features from scholarly reviews. The result indicates the effectiveness of hand-engineered features as compared to automatically-learned representations in predicting review conformity. This is likely due to the similarity of the vocabulary used in most reviews, making review content alone not highly predictive of review conformity. In contrast, we find through DoesMR that hand-engineered features such as the relative strength of a review recommendation (and confidence) with respect to other reviews on the same paper are highly predictive of the review conformity. Second, we observe that methods developed for modeling scholarly reviews generally outperform those for modeling non-scholarly reviews. In particular, deep neural networks for predicting the helpfulness of product reviews, i.e., PCNN and RAHP, generally reach the lowest performance. These results indicate that models developed in other domains cannot be easily transferred to assess review conformity. Among the two pre-trained language models SciBERT and ALBERT, we observe that SciBERT, which is pre-trained on corpora including computer science publications, does not necessarily outperform ALBERT. Such a result indicates that language models pre-trained on scientific publications are not necessarily effective for modeling scholary reviews.

Most importantly, PGPR achieves the best performance on both datasets. Overall, it improves the second best method by 15.19% accuracy and 9.46% F1-score on ICLR-2018 and by 6.51% accuracy and 6.06% F1-score on ICLR-2019. Such a result underlines the effectiveness of our approach in integrating peer grading into model training. The relatively lower improvement on ICLR-2019 compared to that on ICLR-2018 is likely due to the larger historical data with ground truth available for training, which we investigate latter in our experiments.

4.5.4 Ablation Studies & Uncertain Reviews

The comparison between PGPR and machine learning baselines is shown in Figure 4.3. The Mix-model, which consists of the machine learning component of PGPR, outperforms both DoesMr and SciBERT by 11.5% and 40.9% accuracy and by 5.8% and 33.5% F1-score, respectively. These results show the complementary predictive power of hand-engineered features and embeddings. We observe that PGPR outperforms the Mix-model by 5.3% accuracy and by 2.8% F1-score on average on both datasets. This result indicates that using worker's grading improves substantially the model performance. We also observe that PGPR outperforms Mix-model additionally trained with workers grading (aggregated by majority voting), i.e., Mix-model+MV, by 4.8% accuracy and 2.47% F1-score. These results show that PGPR is better at utilizing worker's grading for conformity prediction by taking into account worker reliability.



(a) Accuracy

(b) F1-score

Figure 4.3: Comparison between PGPR and machine learning baselines measured by (a) Accuracy and (b) F1-score.

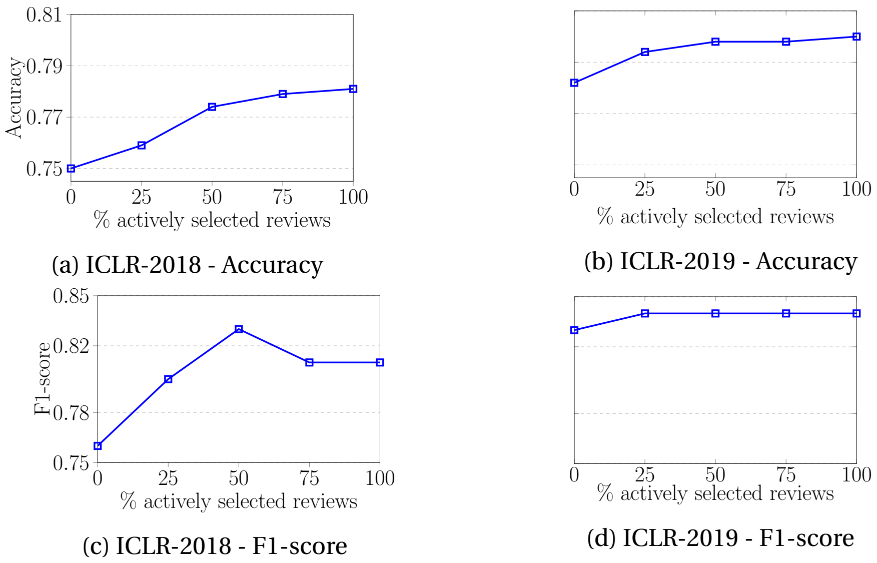
Table 4.5: Analysis of PGPR performance in terms of accuracy on certain and uncertain reviews.

Method	**Dataset**					
	ICLR-18			**ICLR-19**		
	all	**certain**	**uncertain**	**all**	**certain**	**uncertain**
Mix-model	0.752	0.845	0.510	0.793	0.801	0.764
PGPR	0.781	0.846	0.630	0.799	0.801	0.807

Table 4.5 shows a breakdown comparison between the performance of Mix-model and PGPR using the uncertain (actively selected) and certain reviews. We observe that PGPR outperforms the Mix-model by 23.53% and by 5.63% on the uncertain reviews from ICLR-2018 and ICLR-2019, respectively. We also observe that PGPR has little improvement over Mix-model on the certain reviews. These results show that considering workers' grading is important in predicting the conformity of uncertain reviews accurately while having little effect on certain ones. We also find that despite the importance of worker's grading in PGPR, the grading alone is not sufficient to predict the conformity of reviews. Using a majority aggregation of grading on the uncertain reviews leads to an accuracy of 0.61 and 0.73 on ICLR-2018 and ICLR-2019, respectively; i.e., less by 3.17% and 9.54% than our framework's performance. This result shows that combining workers grading with machine learning is crucial for an accurate prediction of review's conformity.

4.5.5 Grading Effect Over Time

The key advantage of our framework is leveraging peer grading for conformity prediction. In what follows, we study the impact of varying the amount of graded reviews on the performance of our framework. We measure the impact on PGPR performance by varying the percentage of the actively selected reviews. We split the graded reviews by s_{act} where we vary s_{act} between 0% and 100%, where $s_{act} = 50\%$ means that we use 50% of the graded reviews in addition to



(a) ICLR-2018 - Accuracy

(b) ICLR-2019 - Accuracy

(c) ICLR-2018 - F1-score

(d) ICLR-2019 - F1-score

Figure 4.4: Performance of PGPR over the two years (2018-2019) with an increasing number of actively graded reviews.

the historical data for training. The results are shown in Figure 4.4 where we use the same y-scale for ICLR-2018 and ICLR-2019 for ease of comparison. We observe that the performance of our framework increases along with the increase of s_{act} on the ICLR-2018 dataset while it gradually stabilizes with the increase of s_{act} on the ICLR-2019 data. This on one hand, confirms the effectiveness of integrating peer grading for model performance. On the other hand, using PGPR in subsequent editions of the same conference requires less grading from one year to the next, as it gradually “learns" the conformity standards of the conference. This property is highly desirable in real-world scenarios as with the increase of the number of submissions (and consequently the number of reviews) our model improves its prediction on the conformity of reviews while requiring fewer reviews to be graded.

4.6 Conclusion

In this chapter, we presented a human-AI approach that estimates the conformity of scholarly reviews by leveraging both human and machine intelligence. We introduced peer grading mechanisms that involve peer reviewers to grade each others’ reviews anonymously and a Bayesian framework that seamlessly integrates peer grading with a machine learning model for review conformity assessment. The peer grading mechanism can be easily incorporated into current peer review systems without inducing much extra effort from the reviewers. The machine learning model trained by the Bayesian framework can continuously learn from new grading from peer reviewers over time. Through a crowdsourced, longitudinal study over a three years-worth dataset, we showed that our approach substantially improves the state of the art and that the machine learning in our framework can largely improve the performance over three consecutive years.

The proposed frameworks OpenCrowd and PGPR allow us to evaluate open-ended answers and estimate the reliability of workers. We showed that they perform better than machine learning or human contributors working alone. Nonetheless, end-users without prior knowledge might still have reservations about using these frameworks in real-world scenarios because they do not fully understand the method's reasoning. This lack of transparency is not a specific problem of our frameworks but common to many state-of-the-art methods. This problem is raising significant attention, and many efforts are dedicated to pushing for explainable methods. In the next chapter, we contribute to this line of work and propose a hybrid human-AI approach that incorporates human rationales into a machine learning model to improve the explainability of its results.

5 MARTA: Leveraging Human Rationales for Explainable Text Classification

Explainability is a key requirement for text classification in many application domains ranging from sentiment analysis to medical diagnosis or legal reviews. Existing methods often rely on “attention" mechanisms for explaining classification results by estimating the relative importance of input units. However, recent studies have shown that such mechanisms tend to misidentify irrelevant input units in their explanation.

In this chapter, we propose a hybrid human-AI approach that incorporates human rationales into attention-based text classification models to improve the explainability of classification results. Specifically, we ask workers to provide rationales for their annotation by selecting relevant pieces of text. By doing so, we couple a boolean task (i.e., text classification) with an open-ended task (i.e., text justification) and leverage both annotations in our approach.

At the technical level, we introduce MARTA, a Bayesian framework that jointly learns an attention-based model and the reliability of workers while injecting human rationales into model training. We derive a principled optimization algorithm based on variational inference with efficient updating rules for learning MARTA parameters. Extensive validation on real-world datasets shows that our framework significantly improves state-of-the-art both in terms of classification explainability and accuracy.

5.1 Introduction

Text classification is a fundamental task in natural language processing (NLP) [233, 221, 10]. State-of-the-art methods are dominated by neural network models, which are generally considered as “black boxes” by end-users. The opaqueness of those models has become a major obstacle for their development, deployment, and improvement, particularly in critical tasks such as medical diagnosis [102] and legal document review [44, 122]. Explainable text classification has, therefore, emerged as an important topic, where the goal is to present end-users with human-readable descriptions of the classification rationale [160, 184, 32, 116].

Among existing explainability methods, a popular approach is the *attention* mechanism,

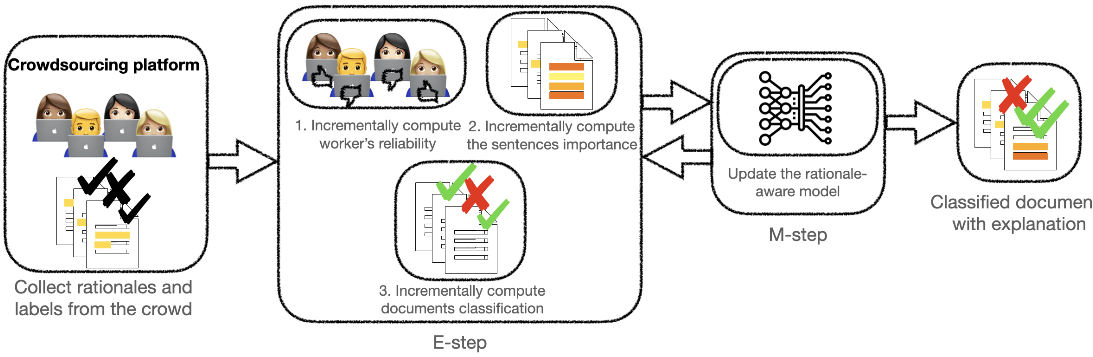

Crowdsourcing platform
Collect rationales and labels from the crowd
1. Incrementally compute worker's reliability
2. Incrementally compute the sentences importance
3. Incrementally compute documents classification
E-step
Update the rationale-aware model
M-step
Classified documents with explanation

Figure 5.1: The MARTA Framework

which identifies important parts of the input for the prediction task by providing a distribution over attended-to input units (e.g., tokens or sentences) [217, 13]. Attention-based models have resulted in impressive performance across many NLP tasks including text classification, question-answering, and entity recognition [13, 147, 209]; in particular, the self-attention mechanism that underlies the Transformer architecture [197, 53] has been playing a central role in many NLP systems. Despite that, recent studies have shown that the learned attention weights are often uncorrelated with the importance of input components measured by other explainability methods (e.g., gradients [178]), and that one can identify different attention distributions that nonetheless yield equivalent predictions [78, 215].

A promising approach to enhance the explainability of attention-based models is integrating human rationales as extra supervision information for attention learning. Prior research [226, 233] has shown that human rationales represent valuable input for improving model performance and for identifying explainable input features in model prediction [13, 132]. Coincidentally, recent studies in human computation [125] have demonstrated that asking workers to provide annotation rationales – by highlighting supporting text excerpts from the given text – brings no extra annotation efforts. Human rationales are, therefore, easy-to-obtain information with great potential in improving model explainability and performance. Existing work [13, 132], however, takes human rationales as gold information that is entirely trustworthy, which is typically not the case in practice; indeed, studies from human computation have found the reliability of human-contributed rationales to be a key problem that requires careful treatment [226, 157].

In this work, we tackle the problem of rationale reliability by introducing a human-AI computational approach that integrates human rationales into an attention-based model while weighing individual reliability. We crowdsource the task of annotating documents and ask workers to justify their labels using text excerpts from the document. We introduce MARTA, a Bayesian framework that jointly learns the workers' reliability and the attention-based model parameters while MApping human Rationales To Attention. The model parameters and worker reliability are updated in an iterative manner, allowing their learning processes to benefit from each other until agreements on both the label and rationales are reached. The overall process is depicted in Figure 5.1. We formalize such a learning process with a principled optimization algorithm based on variational expectation-maximization. In particular, we derive efficient updating rules that allow both model parameters and worker reliability to be updated incrementally at each iteration.

In summary, we make the following contributions:

- We propose MARTA, a Bayesian framework for explainable text classification that integrates human rationales into attention-based models.
- We derive an efficient learning algorithm based on variational inference with incremental updating rules for MARTA parameter estimation.
- We conduct an extensive evaluation on two real-world datasets and show that MARTA substantially outperforms the state of the art by 5.76% F1-score while offering a human-understandable explanation.

5.2 Related Work

5.2.1 Explainable Text Classification

Driven by the need for transparency, machine learning explainability has drawn significant attention recently [160, 56]. Existing explainability methods fall into two broad categories: post-hoc explainability and intrinsic explainability. Post-hoc explainability aims at providing explanations for an existing model. A representative method is LIME [160], which approximates model decisions with an explainable model (e.g., a linear model) in the local area of the feature space. A recent development of this topic is GEF [116], which is designed to explain a generic encoder-predictor architecture by jointly generating explanations and classification results. Another class of methods identifies important features by calculating the gradient of an output with respect to an input feature to derive the contribution of the various features [178, 163, 6]. Intrinsic explainability aims at constructing self-explanatory models. This can be achieved by adding explainability constraints in model learning to enforce feature sparsity [63], representation disentanglement [230], or sensitivity towards input features [184]. Our work falls into this second category by injecting human rationales into model learning through a unified Bayesian framework.

To explain individual predictions, a more popular approach is attention mechanisms, which identify parts of the input that are attended by the model for specific predictions [217, 13]. These attention mechanisms have been playing an important role in NLP not only for explainability but also for the enhancement they bring to model performance [147, 197, 53]. Their effectiveness in explainability, however, has recently been questioned by an empirical study, which points to the facts that attention distributions are inconsistent with the importance of input units as measured by gradient-based methods and that adversarial distributions can be found yielding similar model performance [78]. Those findings have triggered heated discussions, e.g., it has been shown that attention mechanisms attribute higher weights to important input units for a given task even when the model architecture for prediction changes [215]. Our work contributes to the discussion by showing that human rationales, when properly injected into the attention-based models, can enhance the model explainability and performance.

5.2.2 Human Rationale in Machine Learning

The idea of incorporating human rationales for model improvement can be traced back to Zaidan et al. (2007) [226], where a human teacher highlights pieces of text in a document as a rationale to justify label annotation. The rationale is fused into the loss function of an SVM classifier by constraining the prediction labels. Similar ideas have been explored for neural network models [233] and through different ways of human rationale integration, e.g., by learning a mapping between human rationales and machine attention [17] or ensuring the diversity among the hidden representations learned at different time steps [132]. The idea of finding a small subset of input units capable of generating the same output has resulted in various selective rationalization techniques [106, 109, 40, 38, 224]. Despite all existing efforts, few studies have addressed the potential issues in human involvement, such as controlling the quality of rationales contributed by humans with varying levels of expertise and motivation. Unlike them, our framework offers a principled method to model human reliability in integrating human rationales.

A separated line of research in human computation and crowdsourcing has investigated the task design for soliciting human rationales in crowdsourcing settings. When gathering relevance judgments for search results, McDonnell et al. [125] found out that by asking crowd workers to provide 2-3 sentences of document excerpts for justification, annotation quality can be largely enhanced without the task completion time being increased. However, it is also known that the quality of human-contributed rationales remains a challenging issue, especially for subjective and complex tasks [226, 157]. Aligned with these works, our work offers a computational approach that integrates human rationales for explainable text classification while addressing the reliability issue of rationales through a principled learning algorithm.

5.3 Method

MARTA is a unified Bayesian Framework that integrates an attention-based model with labels and rationales contributed by workers. In this section, we first formally define our problem, and then introduce our framework, followed by a presentation of our algorithm for learning MARTA parameters.

5.3.1 Problem Formulation

Notations.

We use boldface lowercase letters to denote vectors and boldface uppercase letters to denote matrices. For an arbitrary matrix $\mathbf{M}$, we use $\mathbf{M}_{i,j}$ to denote the entry at the i-th row and j-th column. We denote the set of documents as $\mathscr{I}$, the set of sentences composing all documents as $\mathscr{S}$, and the set of sentences belonging to document i as $\mathscr{S}_i$. We denote the set of workers who provide noisy labels with rationales as $\mathscr{J}$. The subset of workers who label document i is denoted as $\mathscr{J}_i$. We consider binary classification and use $\mathbf{A}_{i,j} = 1$ to denote that a document i is classified as positive by worker j, and $\mathbf{A}_{i,j} = 0$ otherwise. We use $\mathbf{B}_{s,j} = 1$ to denote that a sentence s is selected as a rationale by worker j. The subset of workers who select the sentence s as a rationale for their annotations is denoted as $\mathscr{J}_s$.

Problem Definition.

Let $\mathscr{I}$ be a set of documents, each assigned to a unique binary label representing its relevance to a topic. Each document $i \in \mathscr{I}$ is composed of a set of sentences $\mathscr{S}_i$ that can be used as rationale in determining the relevance of a document to the topic. Let $\mathscr{J}$ be a set of workers who annotate the documents with labels $\mathbf{A}$ and rationales $\mathbf{B}$. Our goal is to infer the true label of a document denoted as z_i while estimating the importance of each sentence $s \in \mathscr{S}_i$, denoted as α_s, in the inference.

5.3.2 The MARTA Framework

MARTA is a probabilistic framework that models the process of worker-provided labels (i.e., $\mathbf{A}$) and rationales (i.e., $\mathbf{B}$), conditioned on the true labels (i.e., z), the importance of rationales (i.e., α), and the reliability of workers (i.e., r). The overall framework is depicted as a graphical model in Figure 5.2. In the following, we first describe how an attention-based model is embedded into MARTA to allow the integration of human rationales, and then describe the process of worker-provided labels and answers for their integration.

Rationale-Aware Attention Model.

Given the true label of a document $z_i \in \{0,1\}$ as a binary variable, we model it with a Bernoulli distribution. The underlying intuition of our rationale-aware attention model is that the label of a document is determined by its sentences, and that each sentence contributes differently in determining the overall label of the document. Formally, we have:

$$z_i \sim Ber(\theta_i), \theta_i = \sum_{s \in \mathscr{S}_i} a_s P_s, \tag{5.1}$$

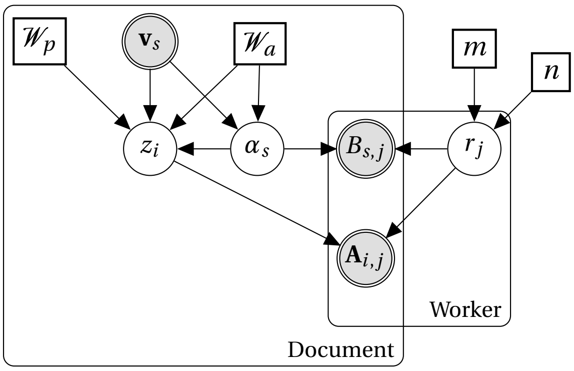


Figure 5.2: Graphical representation of MARTA. Double (greyed) circles represent observed variables, while single circles represent latent variables. Squares represent model parameters. Edges represent conditional relationships in text classification. On the left side, an attention-based model parameterized by $\{\mathcal{W}_a, \mathcal{W}_p\}$ predicts the label z_i for a document. Each document is composed of sentences v_s, with an importance α_s in the classification. On the right side, a worker is represented with her reliability distribution r_j with parameters m and n. The work annotates a document with label $A_{i,j}$ and rational $B_{s,j}$.

where θ_i is the parameter of the distribution, modeled as the weighted sum of the sentence-level label P_s with attention weight a_s. The sentence-level label P_s is predicted from the contents of the sentence through a neural network of arbitrary architecture:

$$P_s = \text{softmax}(f^{W_p, b_p}(\mathbf{v}_s)), \tag{5.2}$$

where $\mathbf{v}_s$ is the embedding vector of the sentence s, and $f^{W_p, b_p}(\mathbf{v}_s)$ models the output of the network layers preceding the softmax layer, parameterized by W_p and b_p and shared across all sentences.

To model the attention weight a_s for each sentence, we use a Bidirectional LSTM (BiLSTM) [169] to account for the sequential dependencies among sentences. Specifically, each sentence vector is transformed into a hidden vector h_s through BiLSTM:

$$h_s = \text{BiLSTM}(\mathbf{v}_s). \tag{5.3}$$

Then, the attention weight of a sentence is modeled through a fully-connected layer and a softmax normalization:

$$a_s = \text{softmax}(h'_s), h'_s = \tanh(W_a h_s + b_a). \tag{5.4}$$

Finally, we model if a sentence can be viewed as a rationale for the document label as a binary variable $\alpha_s \in \{0, 1\}$ that follows a Bernoulli distribution, parameterized by a_s:

$$\alpha_s \sim Ber(a_s). \tag{5.5}$$

Integrating Labeling Rationales.

We represent worker reliability by $r_j \in [0, 1]$ where $r_j = 1$ indicates that the worker is fully reliable and $r_j = 0$ otherwise. In practice, we would like to measure our confidence for an estimate r_j as dependent on the number of answers of worker j, i.e., the more annotations a worker provides, the more confident we would like to be about her reliability estimate r_j. To quantify the confidence of our estimates, we adopt a Bayesian treatment of r_j by modeling it with a Beta distribution:

$$r_j \sim Beta(m, n), \tag{5.6}$$

where m and n are the parameters of the distribution.

We use the reliability of a worker to define the likelihood of her rationale being a true support of the document label:

$$p(\mathbf{B}_{s,j}|\alpha_s, r_j) = r_j^{\mathbb{1}[\alpha_s = \mathbf{B}_{s,j}]}(1 - r_j)^{\mathbb{1}[\alpha_s \neq \mathbf{B}_{s,j}]}, \tag{5.7}$$

where $\mathbb{1}[\cdot]$ is an indicator function returning 1 if the statement is true and 0 otherwise.

Similarly, we use the reliability of a worker to define the likelihood of her provided label being the true label:

$$p(\mathbf{A}_{i,j}|z_i, r_j) = r_j^{\mathbb{1}[z_i = \mathbf{A}_{i,j}]}(1 - r_j)^{\mathbb{1}[z_i \neq \mathbf{A}_{i,j}]}. \tag{5.8}$$

5.3.3 Variational Inference

Learning the parameters of MARTA resorts to maximizing the following likelihood function:

$$p(\mathbf{A},\mathbf{B}) = \int p(\mathbf{A},\mathbf{B},\mathbf{z},\mathbf{r},\alpha,|\mathscr{W},\mathbf{V})d\mathbf{z},\mathbf{r},\alpha, \tag{5.9}$$

where $\mathbf{z}$, $\mathbf{r}$ and α are latent variables, $\mathscr{W}$ represents the set of parameters of the model, i.e. $\mathscr{W} = \{\mathscr{W}_a, \mathscr{W}_p\}$, and $\mathbf{V}$ is the embedding of all the sentences composing the documents. Since Eq. (5.9) contains more than one latent variable, it is computationally infeasible to optimize [195]. Therefore, we consider the log of our likelihood function, i.e.,

$$\log(p(\mathbf{A},\mathbf{B})) = \underbrace{\int q(\mathbf{z},\mathbf{r},\alpha)\log\frac{p(\mathbf{A},\mathbf{B},\mathbf{z},\mathbf{r},\alpha|\mathscr{W},\mathbf{V})}{q(\mathbf{z},\mathbf{r},\alpha)}d\mathbf{z},\mathbf{r},\alpha}_{\mathscr{L}(\mathscr{W},q)} + \underbrace{\int q(\mathbf{z},\mathbf{r},\alpha)\log\frac{q(\mathbf{z},\mathbf{r},\alpha)}{p(\mathbf{z},\mathbf{r},\alpha|\mathbf{A},\mathbf{B},\mathscr{W},\mathbf{V})}d\mathbf{z},\mathbf{r},\alpha}_{KL(q||p)}, \tag{5.10}$$

where $q(\mathbf{z},\mathbf{r},\alpha)$ is any probability density function and $KL(\cdot)$ is the KL divergence between two distributions. By doing so, the two parts of the objective function can then be optimized iteratively with a variational expectation-maximization method [195]. Specifically, we iterate between two steps: 1) the E-step, where we approximate the latent variables $p(\mathbf{z},\mathbf{r},\alpha|\mathbf{A},\mathbf{B},\mathscr{W},\mathbf{V})$ with the variational distribution $q(\mathbf{z},\mathbf{r},\alpha)$, by minimizing the KL-divergence. 2) the M-step, where we maximize the term $\mathscr{L}(\mathscr{W},q)$ given the newly inferred latent variables.

E step.

We use the mean field variational inference approach [26] by assuming that $q(\mathbf{z},\mathbf{r},\alpha)$ factorizes over the latent variables.

$$q(\mathbf{z},\mathbf{r},\alpha) = \prod_i q(z_i)\prod_s q(\alpha_s)\prod_j q(r_j). \tag{5.11}$$

To minimize the KL divergence, we choose following forms for the factor functions:

$$q(z_i){=}Ber(\theta_i); q(\alpha_s){=}Ber(a_s); q(r_j){=}Beta(m_j,n_j), \tag{5.12}$$

where θ_i, a_s, m_j and n_j are variational parameters used to minimize the KL divergence. The latter can be minimized by updating one latent variable at a time and keeping all others fixed.

In the following, we derive the updating rules for $q(z_i)$, $q(\alpha_s)$ and $q(r_j)$. To do so, we simplify Eq.(5.10) and obtain for each latent variable the following inference equations:

$$
\begin{aligned}
q(z_i) =& \prod_{s\in\mathcal{S}_i, j\in\mathcal{J}_i} p(z_i|\mathbf{v}_s,\mathcal{W}) g_{q(r_j)}(p(\mathbf{A}_{i,j}|z_i,r_j)),\\
q(\alpha_s) =& \prod_{j\in\mathcal{J}_s} p(\alpha_s|\mathbf{v}_s,\mathcal{W}_a) g_{q(r_j)}(p(\mathbf{B}_{s,j}|\alpha_s,r_j)),\\
q(r_j) =& \prod_{i\in\mathcal{I}_j, s\in\mathcal{S}_j} p(r_j) g_{q(z_i,\alpha_s)}(p(\mathbf{A}_{i,j}|z_i,r_j)p(\mathbf{B}_{s,j}|\alpha_s,r_j)),
\end{aligned}
\tag{5.13}
$$

where $\mathcal{S}_i$ are the sentences in a document i. $\mathcal{J}_i$ and $\mathcal{J}_s$ are the workers annotating document i and those choosing sentence s as a rationale, respectively. $\mathcal{I}_j$ and $\mathcal{S}_j$ are the documents annotated by worker j and the sentences chosen by her as rationales, respectively. We use $g_x(\cdot)$ to denote the exponential of expectation term $\exp\{\mathbb{E}_x[\log(\cdot)]\}$ with x being a variational distribution. With the above equations, we obtain the updating rules for all latent variables. We first give the updating rules of the document label z_i and the sentence's importance α_s by the following lemmas.

Lemma 6 (Incremental Document Classification)**.** *The true label distribution $q(z_i)$ can be incrementally computed using the predicted label by the attention-based model θ_i, and the parameters m_j and n_j of the worker reliability distribution r_j.*

$$
q(z_i=1) \propto \begin{cases} \theta_i \prod_{j\in\mathcal{J}_i} \exp\{\Psi(n_j)-\Psi(m_j+n_j)\}, & if\mathbf{A}_{i,j}=0,\\ \theta_i \prod_{j\in\mathcal{J}_i} \exp\{\Psi(m_j)-\Psi(m_j+n_j)\}, & if\mathbf{A}_{i,j}=1, \end{cases}
\tag{5.14}
$$

where Ψ is the Digamma function. If $q(z_i=0)$ then we replace θ_i by $1-\theta_i$.

Lemma 7 (Incremental Sentence Importance)**.** *The importance of a sentence for document classification can be incrementally computed using the attributed attention weight by the attention-based model a_s and the parameters m_j and n_j of the worker reliability distribution r_j.*

$$
q(\alpha_s=1) \propto \begin{cases} a_s \prod_{j\in\mathcal{J}_s} \exp\{\Psi(n_j)-\Psi(m_j+n_j)\}, & if\mathbf{B}_{s,j}=0,\\ a_s \prod_{j\in\mathcal{J}_s} \exp\{\Psi(m_j)-\Psi(m_j+n_j)\}, & if\mathbf{B}_{s,j}=1. \end{cases}
\tag{5.15}
$$

Next, we show the updating rule for the worker reliability $q(r_j)$ with the following lemma.

Lemma 8 (Incremental Worker Reliability)**.** *The worker reliability distribution $q(r_j)$ can be incrementally computed using her annotation and rationale quality, and the reliability parameters m_j and n_j from the previous iteration.*

$$q(r_j) \propto \begin{cases} Beta(m'_j + \sum_{s\in\mathscr{S}_j}(1-a_s), n'_j + \sum_{s\in\mathscr{S}_j} a_s), if \mathbf{B}_{s,j}=0, \\ Beta(m'_j + \sum_{s\in\mathscr{S}_j} a_s, n'_j + \sum_{s\in\mathscr{S}_j}(1-a_s)), if \mathbf{B}_{s,j}=1, \end{cases} \tag{5.16}$$

where $m'_j = m_j + \sum_{i\in\mathscr{I}_j}\theta_i$ *and* $n'_j = n_j + \sum_{i\in\mathscr{I}_j}(1-\theta_i)$, *if* $\mathbf{A}_{i,j}=1$ *and* $m'_j = m_j + \sum_{i\in\mathscr{I}_j}(1-\theta_i)$ *and* $n'_j = n_j + \sum_{i\in\mathscr{I}_j}\theta_i$, *if* $\mathbf{A}_{i,j}=0$.

We provide proofs for all lemmas in the appendix.

M step.

Given the true labels of the documents, the importance of sentences, and the worker reliability inferred by the E-step, the M-step maximizes the first term of Eq. (5.10) to learn the parameters $\mathscr{W} = \{\mathscr{W}_a, \mathscr{W}_p\}$.

$$\begin{aligned} \mathscr{L}(\mathscr{W}, q) &= \int q(z,\alpha,r)\log[p(\mathbf{A},\mathbf{B},z,\alpha,r|\mathbf{V},\mathscr{W})]dr + C_1 \\ &= \underbrace{\sum_{z_i} q(z_i)\log[p(z_i|\mathbf{V};\mathscr{W}_a,\mathscr{W}_p)]}_{\mathscr{T}_1} + \underbrace{\sum_{\alpha_s} q(\alpha_s)\log[p(\alpha_s|\mathbf{v}_s;\mathscr{W}_a)]}_{\mathscr{T}_2} + C_1 + C_2, \end{aligned} \tag{5.17}$$

where $C_1 = \exp\{\mathbb{E}_{q(z,\alpha,r)}[\log((\frac{1}{q(z,\alpha,r)})]$ is a constant and C_2 are the terms that do not depend on the parameters $\mathscr{W}$. The term $\mathscr{T}_1$ is equivalent to the inverse of the cross entropy between the target labels of a document $q(z_i)$ and the predicted label $p(z_i|\mathbf{V};\mathscr{W}_a,\mathscr{W}_p)$. Similarly, the term $\mathscr{T}_2$ is equivalent to the inverse of the cross entropy between the indication of a sentence as a rationale and the predicted importance $p(\alpha_s|\mathbf{v}_s;\mathscr{W}_a)$. Given the shared parameter $\mathscr{W}_a$, we minimize the prediction loss $\mathscr{T}_1$ together with the loss $\mathscr{T}_2$.

5.3.4 Algorithm

The overall optimization algorithm is given in Algorithm 4. We initialize MARTA's parameters and iterate between an E step (rows 2-6) and an M-step (rows 7-8). The E-step consists of updating the variational distribution of the document labels, the sentence importance and the worker reliability. Our framework is semi-supervised in the sense that when ground truth labels are available, we fix them in the E-step. The M steps consists in updating the parameters of the attention-based model by jointly learning the document labels and the sentence's importance. It is worth noting that for this step, the loss between human rationales and the attention generated by the model is minimized. The convergence is reached when the documents label $q(z_i)$ and the sentences relevance $q(\alpha_s)$ are no longer modified by the workers' reliability and the model's parameters stabilize.

The iterations through the documents (rows 2-4) yields a time complexity of $|\mathscr{I}|$ while the

Algorithm 4: Learning MARTA Parameters

```
  Input     :A, B, $\mathcal{S}_i$ ($\forall i \in \mathcal{I}$)
  Output    :Variational distributions: $q(z_i)$, $q(\alpha_s)$ and $q(r_j)$
  Initialize:MARTA parameters: $\theta_i$, $m_j$, $n_j$, $\mathcal{W}$
1 while Eq. (5.10) has not converged do
2     for $i \in \mathcal{I}$ do
3         update $q(z_i)$ using Lemma 6;
4         update $q(\alpha_s)$ using Lemma 7;
5     for $j \in \mathcal{J}$ do
6         update $q(r_j)$ using Lemma 8;
7     for $i \in \mathcal{I}$ do
8         Update $\mathcal{W}$;
```

Dataset	#Docs	%Positive	#Judgments	#Workers
Wiki-Tech	1413	17.26%	4488	58
Amazon	400	50%	6744	449

Table 5.1: Datasets Description

iterations through the workers (rows 5-6) yields a time complexity of $|\mathcal{J}|$. The overall complexity of our algorithm is $O(\#iter \times (|\mathcal{I}| + |\mathcal{J}| + \mathcal{C}_W))$, where $\#iter$ is the number of iterations needed to converge and $\mathcal{C}_W$ is the complexity of learning the parameters $\mathcal{W} = \{\mathcal{W}_a, \mathcal{W}_p\}$ of the attention-based model.

5.4 Experiments

5.4.1 Experimental Setup

Datasets. We use two datasets for our experiments: *Wiki-Tech* and *Amazon*[1]. *Wiki-Tech* contains 1413 Wikipedia articles with expert annotations on their relevance with respect to the topic "technologies commonly used by companies". We crowdsourced this dataset to collect worker rationales. *Amazon* is developed and published by Ramirez et al. (2019) [157]. It contains 400 reviews with ground truth labels about "reviews written about books"; this dataset is released with worker's rationales. Key statistics about both datasets are reported in Table 5.1.

Crowdsourcing Task. Worker annotations in *Wiki-Tech* were collected through a crowdsourcing task that we published on Amazon Mechanical Turk[2]. We asked workers the following predicate: *Does the Wikipedia article describe a technology commonly used by companies?*. We

[1] Source code and data are available at `https://github.com/eXascaleInfolab/MARTA`.

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

Method	Wiki-Tech				Amazon			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
MILNET	0.683	0.340	0.890	0.490	0.840	0.850	0.820	0.840
fastText	0.829	0.521	0.268	0.349	0.780	0.750	0.888	0.804
SciBERT	0.779	0.440	**0.970**	0.600	0.920	0.940	0.900	0.92
ALBERT	0.882*	**0.708**	0.560	0.618*	0.946	0.960*	0.932	0.946
LSTM-ortho	0.799	0.464	0.829	0.590	0.822	0.699	0.756	0.725
LSTM-diversity	0.649	0.365	0.928*	0.506	0.952*	0.960*	**0.944**	0.952*
InvRAT	0.717	0.220	0.210	0.230	0.720	0.750	0.720	0.710
RA-CNN	0.813	0.428	0.442	0.432	0.667	0.652	0.68	0.661
MARTA	**0.886**	0.660*	0.700	**0.680**	**0.960**	**0.980**	0.940*	**0.960**

Table 5.2: Performance (Accuracy, Precision, Recall and F1-score) comparison with baseline methods. The best performance is highlighted in bold; the second best performance is marked by '*'.

chose workers with a HIT approval rate above 70%. The task started by explaining the concept of “Technology" and provided a positive and a negative example. Then, workers were asked to annotate the article and provide a snippet from the text as a justification. Workers took on average 1 minute to complete the task and were rewarded 16 cents per answer (we made sure that we pay over 8USD per hour). The crowdsourcing task used to collect the *Amazon* dataset consisted in asking workers: *Is this review written on a book?* The full experiment is described in length in [157].

Representation Learning. The inputs of our machine learning model are the sentences from the documents. We represent each sentence as a fixed-size vector $\mathbf{v}_s$ by leveraging pre-trained language models. We use SciBERT as pre-trained word embeddings for *Wiki-Tech* since the language in Wikipedia is formal and contains scientific terms and ALBERT for *Amazon* as it contains reviews with less formal language compared to the documents used to train SciBERT. Considering the size of the datasets, we use a neural network with one fully-connected layer for sentence-level label prediction (Eq. (5.2)).

Comparison Methods. We compare our approach to a wide range of baselines. First, we compare against a set of recent text classification methods: 1) MILNET [7], a Multiple Instance Learning (MIL) neural network model. 2) fastText [83], a linear model for text classification that uses bags of n-grams as additional features to capture information about the local word order. 3) SciBERT [20], a language model trained on scientific text consisting of scholar papers from the computer science and biomedical domains. 4) ALBERT [103], a pre-trained language model that takes into account the inter-sentence coherence, which allows to capture fine-grained information in documents.

In addition, we compare against rational-aware models: 1) LSTM-ortho and LSTM-diversity, both proposed in [132]. These methods extend an LSTM to learn diverse hidden representations at different time steps through an orthogonality and a diversity constraint for hidden states. 2) InvRat [39], a game-theoretic approach that is designed to identify and remove

Method	Implementation
MILNET	github.com/stangelid/oposum
fastText	fasttext.cc/docs/en/supervised-tutorial.html
SciBERT	github.com/allenai/scibert
ALBERT	huggingface.co/transformers/model_doc/albert.html
LSTM	github.com/akashkm99/Interpretable-Attention
InvRAT	github.com/code-terminator/invariant_rationalization
RA-CNN	github.com/yezhang-xiaofan/Rationale-CNN

Table 5.3: Methods Implementation

Method	Wiki-Tech				Amazon			
	#epochs	lr	Batch Size	Other	#epochs	lr	Batch Size	Other
MILNET	25	1e-3	50	-	20	1e-3	50	-
fastText	20	1	-	N-gram = 5	20	9e-1	-	N-gram = 5
SciBERT	75	1e-4	4	-	250	1e-7	8	-
ALBERT	10	1e-4	8	-	10	1e-4	8	-
LSTM-ortho	8	-	32	-	8	-	32	-
LSTM-diversity	8	-	32	diversity weight=0.5	8	-	32	diversity weight=0.5
InvRAT	20	1e-5	20	embedding dim=300	20	1e-4	20	embedding dim=300
RA-CNN	15	-	50	-	20	-	50	-
MARTA	20	1e-3	50	$m_j = 2.5, n_j = 2$	20	1e-3	50	$m_j = 2.5, n_j = 2$

Table 5.4: Choice of hyperparameters for baseline methods on the *Wiki-Tech* and *Amazon* datasets. We use '-' to denote if the hyperparameter is not applicable. 'lr' denotes the learning rate and the value reported for m_j and n_j in MARTA, is the value used for initialization.

features with spurious correlation with the output. 3) RA-CNN [233], a sentence-level convolutional model that estimates the probability of a given sentence being a rationale. We note that the LSTM variants (LSTM-ortho and LSTM-diversity) and InvRat generate rationales automatically from the models, while RA-CNN uses the rationale provided by workers. In our experiment, we use the sentences indicated by the majority of workers as rationales to train RA-CNN.

Comparison Methods Implementation

In our experiments, we compare with text classification and rationale-aware methods. We use the authors' implementation for all methods except MILNET, for which we re-implemented the original code in Python. Table 5.3 summarizes all methods implementations used in our experiments. For each method, we tune the hyperparameters including the learning rate in {0.00001,0.0001,0.001,0.01,0.1,1}, the batch size in {10,20,50,100} and the number of epochs in {10,20,50,100,500}. For fastText, we also vary the word n-grams in [1,5]. For InvRat, we vary the embedding dimension in {50,100,200,300}. We use the hyperparameters that led to the optimal results on the validation set. We report the optimal settings in Table 5.4.

Parameter Settings for MARTA

The parameters of our framework are empirically set. We search for the best architecture for our attention-based model by applying a grid search in {10, 20, 50, 100} for the batch size and in {0.00001, 0.0001, 0.001, 0.01, 0.1, 1} for the learning rate. We also test different optimization methods including stochastic gradient descent, ADAM and RMSprop. We initialize the priors m_j and n_j by sampling from a uniform distribution $[0,10]$ and update them in the E-step according to Lemma 8. The optimal parameter settings we found through the validation set are reported in Table 5.4.

Hardware and Software

For our framework, we used a Ubuntu 16.04 machine with 32 CPUs and 128GB RAM. For experiments that required GPU, we used a Ubuntu 18.04 with 9 GPUs (1 TITAN V and 8 GeForce RTX 2080 Ti), 64 CPUs and 395GB RAM. In our code repository, we provide a file (requirement.txt) that specifies the versions of all required libraries; this file can be used to install them automatically.

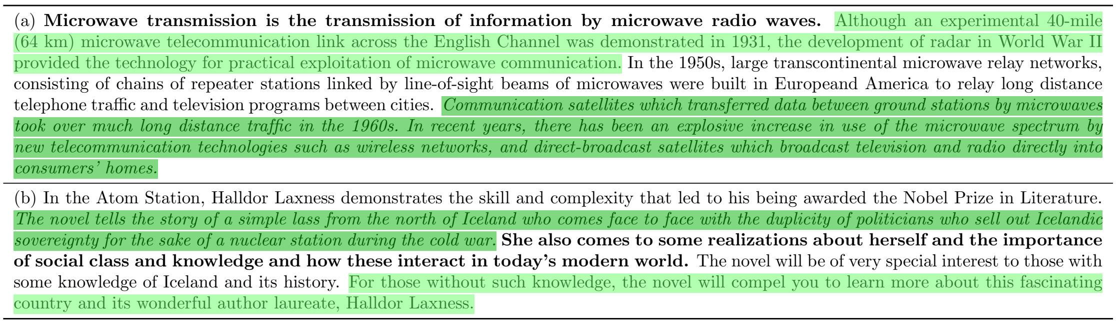
Evaluation Protocol.

We split the datasets into training, validation, and test sets. We use 50% of the data for training and the rest for validation and test with equal split. We report the average over 10 runs for each method. Note that we only use worker's annotations and rationales in the training and validation sets. We use accuracy, precision, recall and F1-score over the positive class to measure the performance. Higher values indicate better performance.

5.4.2 Results and Discussion

Table 5.2 summarizes the performance of MARTA against baseline methods on both *Wiki-Tech* and *Amazon*.

First, we observe that ALBERT and SciBERT perform relatively well compared to the other baseline methods, especially on the *Wiki-Tech* dataset. Recall that both ALBERT and SciBERT leverage textual context for representation learning, which is useful in fine-grained classification tasks, such as *Wiki-Tech* where the model has to capture the relationship between technologies and companies. Second, we observe that among the rationale-aware models, the two LSTM variants, i.e., LSTM-ortho and LSTM-diversity, achieve the highest performance. This confirms the advantage of attention mechanisms and shows the effectiveness of learning non-redundant hidden states for model performance. We also observe that RA-CNN, which uses human rationales, does not necessarily perform well. This is probably due to the way textual data is handled by RA-CNN: as opposed to the LSTM variants where the sequential order in the textual data is modeled (with attention), the textual data is considered as independent



(a) **Microwave transmission is the transmission of information by microwave radio waves.** Although an experimental 40-mile (64 km) microwave telecommunication link across the English Channel was demonstrated in 1931, the development of radar in World War II provided the technology for practical exploitation of microwave communication. In the 1950s, large transcontinental microwave relay networks, consisting of chains of repeater stations linked by line-of-sight beams of microwaves were built in Europeand America to relay long distance telephone traffic and television programs between cities. *Communication satellites which transferred data between ground stations by microwaves took over much long distance traffic in the 1960s. In recent years, there has been an explosive increase in use of the microwave spectrum by new telecommunication technologies such as wireless networks, and direct-broadcast satellites which broadcast television and radio directly into consumers' homes.*

(b) In the Atom Station, Halldor Laxness demonstrates the skill and complexity that led to his being awarded the Nobel Prize in Literature. *The novel tells the story of a simple lass from the north of Iceland who comes face to face with the duplicity of politicians who sell out Icelandic sovereignty for the sake of a nuclear station during the cold war.* **She also comes to some realizations about herself and the importance of social class and knowledge and how these interact in today's modern world.** The novel will be of very special interest to those with some knowledge of Iceland and its history. For those without such knowledge, the novel will compel you to learn more about this fascinating country and its wonderful author laureate, Halldor Laxness.

Figure 5.3: Examples from the *Wiki-Tech* (a) and *Amazon* (b) datasets. Bold letters refer to the weight attributed by an attention-based model. Italic letters indicate a rationale given by a worker. The shades of green refer to the weights given by MARTA: a stronger shade means a higher weight.

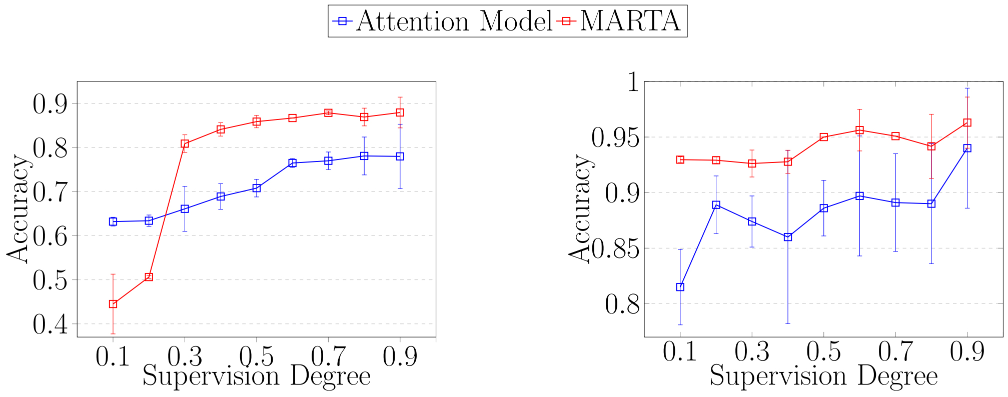
tokens by RA-CNN, which can lead to a loss of contextual meaning.

Most importantly, MARTA achieves the best performance in terms of accuracy and F1-score on both datasets. Overall, it improves ALBERT by 0.97% accuracy and 5.76% F1-score and LSTM-diversity by 18.68% accuracy and 17.61% F1-score on average on both datasets. To further confirm that our way of integrating human rationales is effective, we conducted an ablation study comparing MARTA to a simplified version with only the attention-based model (with pre-trained sentence embeddings). Results show that MARTA improves the performance by 23% accuracy and 28% F1-score in the *Wiki-Tech* dataset and by 12.5% accuracy and F1-score in the *Amazon* dataset. Such a result underlines the effectiveness of weighting the reliability of human rationales when integrating them into attention-based models.

5.4.3 MARTA Properties

In addition to better classification performance, MARTA exhibits a number of properties highly desirable in terms of accountability and deployment. In the following, we present some of these properties.

Explainability. MARTA provides explanations to classification results by incorporating human rationales. A comparison of the overlap between the rationales chosen by annotators and those highlighted by MARTA shows a recall of 71.1% on the *Wiki-Tech* dataset and 61.3% on the *Amazon* dataset, which is respectively 45.1% and 22.6% higher than the attention-based model alone. The precision is 21.7% on *Wiki-Tech* and 27.0% on *Amazon*. These results are due to the fact that MARTA typically tends to select multiple relevant sentences as a rationale, while workers tend to select only one sentence. Figure 5.3 shows two examples from the test sets of the *Wiki-Tech* and the *Amazon* datasets, respectively. The first example describes a technology used by companies. The attention-based model attributes a high weight to the first sentence (in bold), which defines the concept as a technology but not as used by companies. In comparison, our framework attributes high weights to the last two sentences given as



(a) Wiki-Tech - Accuracy

(b) Amazon - Accuracy

Figure 5.4: Performance of MARTA with varying s_deg.

rationales by workers (in italic), as they clearly show the relationship between the technology and companies. In addition, MARTA attributes a high weight to the second sentence that is relevant to the task. These results show that MARTA learns to generalize from human rationales how to identify important sentences. We observe similar results in the example from the *Amazon* dataset: our framework identifies both worker-provided rationales and other relevant sentences for the task.

Adjustable Supervision Degree. Our framework is highly effective even with a relatively small amount of ground truth labels for training. In what follows, we study the impact of the supervision degree to determine the minimum amount of ground truth needed. We split our datasets by s_{deg} where we vary s_{deg} between 10% and 90%, where s_{deg} = 50% means that we use 50% of the ground truth labels for training. We compare with a variant of our model with only the attention-based model described in the Method section. The results are shown in Figure 5.4. We observe that the performance of our framework increases along with the increase of s_{deg} on the *Wiki-Tech* data while it is overall stable for the *Amazon* data. This shows that the amount of ground truth needed to train our framework varies across tasks: compared with *Amazon*, *Wiki-Tech* is a more complex task that requires the model to capture fine-grained information; consequently, it requires more labels in model training. We observe that MARTA has better performance than the attention-based model starting from a supervision degree of 30% on the *Wiki-Tech* dataset and 10% on the *Amazon* dataset. This on one hand, confirms the effectiveness of integrating human rationales for model performance. On the other hand, the fact that a small proportion of labels (less than 30%) does not help to improve model performance on the *Wiki-Tech* dataset indicates that when the task is complex, a small proportion of ground truth labels might not be sufficient to correctly identify the workers' reliability, and that the benefits of having the labels might be over-weighted by the disadvantage of the extra parameters to be learned in that case.

In addition, we measure the performance variation across 10 runs with different data split for each s_deg. The results are shown in Figure 5.4, where the standard deviation is 0.023

and 0.011 on average on *Wiki-Tech* and *Amazon*, respectively. The standard deviation is small compared to the absolute accuracy which demonstrates MARTA's robustness.

5.5 Limitations and Future Work

The main limitation of our technique consists in two assumptions. The first one assumes that a rationale given by a worker can be directly extracted from text. While this assumption is valid in our context, there are applications where the rationale is expressed in a syntax different from the original text. For example, in [125], the rationale is expressed as reasoning. We also assume that a document is composed of many sentences, and that the sentences have different levels of importance. In case the document is short (1-3 sentences), the importance of the sentences does not vary a lot, and hence the attention scores assigned by our framework are almost all equal. In future work, we plan to represent the workers' rationales by embedding and leverage the similarity between the rationale and the original text to derive the sentence's importance. Using embedding to represent rationales would allow us to capture the worker reliability on different topics. We also plan to leverage the rationales generated through our framework to learn other domains' rationale through transfer learning.

5.6 Conclusion

In this chapter, we presented MARTA, a Bayesian framework leveraging human rationales to improve the performance of attention-based models and provide a human-understandable explanation of classification results. Our proposed method incrementally updates the attention distribution by learning from human rationales while taking into account the workers' reliability. Extensive validation on two real-world datasets shows that MARTA is an effective and robust framework that substantially outperforms state-of-the-art methods while providing better, human-understandable explanations.

In the next chapter, we conclude our thesis by summarizing the main findings and contributions of our proposed methods. Then, based on them, we discuss ways to extend our frameworks and define future research directions.

6 Conclusions

In this thesis, we focused on the problem of open-ended data curation while considering humans' contribution. We developed human-AI collaborative approaches that integrate machine learning with human computation such that their learning processes benefit from each other. We tackled the problem from different aspects, such as cleaning and evaluating open-ended answers and injecting them into a model's learning to improve its explainability. Each of these aspects is discussed in a separate chapter of the thesis. In this chapter, we summarize the main contributions and define future research directions.

6.1 Summary

We started by tackling the problem of open-ended answers aggregation in Chapter 3. This problem is particularly challenging in crowdsourcing, where workers provide their answers as free text. In addition to being prone to errors, workers' open-ended answers are sparse and positive-only. These properties contrast with the conventional crowdsourcing setting, where worker's answers are binary or categorical and the overlap between them is sufficient to infer the truth.

To address this problem, we developed a human-AI collaborative approach that integrates machine learning and crowdsourcing for aggregating open-ended answers. In our framework, we cope with the sparsity and positive-only properties of open-ended answers by combining two strategies: 1) training a machine learning model on the answer's features and worker's reliability to infer the truth for unseen data, and 2) using random negative sampling to assign a random set of answers as workers' negative answers. More specifically, our approach consists of a Bayesian framework that learns a feature-based model for the answers' quality and the crowd workers' reliability. We use variational inference, where we model the answer's quality as a discrete variable and the worker's reliability as a continuous variable. We iterate between updating the parameters of our probabilistic model and those of a machine learning model such that their learning processes benefit from each other. Experimental results show that our framework substantially improves the state of the art by 11.5% AUC.

In an effort to improve the evaluation of open-ended answers, we designed in Chapter 4 a framework for peer grading peer reviews where the open-ended answers are, in this context, the scholarly reviews. In this work, we estimate the conformity of scholarly reviews concerning conference standards. This problem is challenging because the task is complex and requires assessing reviews from many dimensions. In addition, fairness is crucial in such a high stake domain, and therefore, uncertain cases should be handled cautiously.

To tackle this problem, we proposed a human-AI approach that estimates the conformity of reviews to the conference standards. Our approach is based on active learning, where we designed a multi-input network that takes both the review's textual content and statistical features to estimate its conformity, and used peer grading for the reviews where the model's prediction is most uncertain. We developed a Bayesian framework for the selected graded reviews to estimate the grader's reliability and bias while updating the model's parameters and conformity estimation. Similar to OpenCrowd, our method is based on variational inference, where we incrementally update both model parameters and graders' reliability at each iteration. The framework's design differs from OpenCrowd as we model the review conformity with a continuous variable ands the grader's bias in addition to her reliability. These design choices allow us to handle the task's complexity while ensuring fairness in handling uncertain cases. Overall, our framework infers the conformity of reviews from the peer grading process, historical reviews, and conference decisions while considering grading reliability and bias. Our approach helps meta-reviewers identify reviews that require clarification while not inducing additional overhead from reviewers. Through a large-scale crowdsourced study where crowd workers are recruited as graders, we show that the proposed approach outperforms existing methods by 11.6% accuracy. Moreover, our method exhibit improvement over subsequent editions of the same conference, a property highly desirable in real-world scenarios with the increase in the number of submissions and consequently the number of reviews.

Finally, we turned our focus to explainability in AI. In Chapter 5, we improve the explainability of text classification methods that rely on a specific machine learning mechanism: attention. Although they supposedly provide an explanation through an attention distribution that reflects the importance of input units, end-users often consider these methods as black boxes because their explanation differs from human rationale.

We address this problem with a framework named "MARTA" for explainable text classification. Our framework is a human-AI hybrid system that seamlessly incorporates human rationale into an attention-based model by using them to guide the learning of their distribution. In this work, we ask workers to provide rationales for their annotation by selecting relevant pieces of text. Similarly to the other frameworks, we use variational inference to update a probabilistic model's parameters and an attention-based model. Nonetheless, MARTA substantially differs from the previous frameworks as we update the model's result and intrinsic learning process by modifying the attention distribution based on the human rationale. Through experiments, we showed that such an approach boosts the explainability and the classification performance of an attention-based model.

6.2 Future Work

This thesis contributed novel frameworks for open-ended answers curation from several important aspects. There are different ways to extend and improve the proposed methodologies. In this section, we outline future research directions for open-ended data curation with crowdsourcing. In addition, we discuss the challenges and research opportunities that we believe could enhance the broad field of data curation.

6.2.1 Research Directions for Open-ended Data Curation with Crowdsourcing

Merging Open-ended Answers

In Chapter 3, we developed a solution for open-ended answers aggregation using annotators' reliability and answers' features to identify high quality answers. Our solution performs well for filtering open-ended answers in tasks such as data enumeration (e.g., collection of social influencers names) and extracting text spans (e.g., rationale extraction). Other types of open-ended tasks such as text translation or audio transcription might require merging workers' answer into one solution in case the best answer is a combination of all collected answers. There are some ongoing efforts to develop such a solution. For instance, in the very large database conference (VLDB) 2021, a crowd science contest [199] was organized by Toloka to aggregate multiple audio transcriptions into a single high-quality transcription, and the winning solution used an extractive text summarization method. Such a solution provides a first step towards merging open-ended answers, however it omits workers' modeling, which could lead to incoherent answers. Open research questions to be addressed in this context include: 1) how to model worker's style in writing open-ended answers; 2) how to integrate workers' modeling in answers merging; 3) how to ensure consistency and coherence of the merged solution when using multiple answers. To deepen our understanding of open-ended answers' aggregation, one can investigate how individuals collaborate in writing, how they split roles and how they combine different contributions and correct mistakes.

Build a Competent Crowd

In the thesis, workers complete complex tasks relying mainly on their background knowledge with little training. It would be valuable to investigate workers' learning curve and how their performance improve over time. We foresee two main challenges: First, how to identify a trade-off between the cost, the learning time, and the quality of answers. In fact, workers invest time in learning which should be rewarded, although they do not contribute yet to the task at hand. Nonetheless, the better workers understand the task, the more likely the answers' quality improve. Second, how to adapt the task to the workers' learning speed. Workers have different background, level of expertise and learning curves, which implies that their improvement over time depends on how well the task is adapted to them. To address this problem, we could draw inspiration from e-learning and teaching techniques [154, 222] and investigate how to treat workers as students to learn the task incrementally by increasing the difficulty level and correcting their mistakes each time.

Modeling Workers' Performance with Time Series

In our work, workers' reliability is represented as a variable dependent on the quality of her answers or on the task's difficulty. Up until now, we did not investigate the temporal dimension in workers' performance, which has important applications in sequential labeling tasks where the annotation of one task can affect the interpretation of subsequent ones. Open research questions to address include: 1) how to model worker's performance over time; 2) what external factors impact workers' performance over time and how to take them into account; 3) how to measure inter-worker agreement in time-dependent tasks. To tackle these questions, we could investigate time series techniques to model workers performance and analyze their patterns to derive inter-workers' agreement and infer the truth.

Mapping Human Reasoning to Model Learning

In Chapter 5, we developed a solution to align the attention distribution of an attention-based model with a human's explanation. While our solution improves model's performance and explanation, it assumes a one-to-one mapping between model learning and human's understanding. Such an assumption sums up the models' learning to quantifying the importance of input units in the model's results. While such an approach allows end-users to see the impact of certain data instances on the model's performance, it does not reflect what the model has learned and missed. Future work could investigate data propagation in model learning and map it to logic rules understandable by end-users. Another direction could be exploring ways to rectify the model's learning such that it follows human reasoning. We foresee as a main challenge the feasibility of mapping model learning to logic rules. Model's learning can be barely traced in small neural networks and it is almost impossible to fully understand the model's parameters update in large models with millions of parameters. One way to proceed could be to use a logic rule-based approach [156, 234] as a translator between a neural network and humans. Such an approach would allow to check if the models had assimilated all patterns needed to perform correctly the task.

6.2.2 Research Directions for Data Curation

Weak supervision for Sequence labeling

In our work, the main source of labeling are workers in crowdsourcing platforms with different levels of reliability. In some circumstances, cost is an issue and instead of relying on manual labor, developers/researchers might choose to use other weak sources such as gazetteers, pre-trained models and heuristic functions. Depending on the data instance, some sources are more reliable than others. A future direction could draw inspiration from the crowdsourcing domain and treat weak sources as workers with different levels of reliability and model the confusion of a weak source as dependent on both its annotation and the input sequence. Such a solution could be applied for sequence labeling tasks such as entity recognition, part-of-speech tagging and phrase extraction.

Taxonomy Enrichment

Taxonomies are a particular type of knowledge graph in which concepts are organized in a tree-like structure and connected through an "is-a" relation. They are widely used in e-commerce, web search, and scientific domains. These taxonomies are usually curated manually by experts. With the emergence of new concepts, they have to be constantly updated. Several methods [173, 228, 123] have been proposed to address this taxonomy expansion problem. Many of them simplify the problem to a hypernym finding problem. Such a mapping does not take into account the semantic similarity between concepts and the hierarchical structure of the taxonomy. Open research questions to tackle include: 1) how to define the semantic similarity between concepts as reflective of their distance in the taxonomy; 2) How to establish hierarchical relationships between concepts that share similar semantic meaning; 3) How to rectify taxonomies' inner structure after the emergence of new concepts. One way to proceed is to map the problem to a semantic search problem where, given a query and a corpus, we identify the closest concept from the corpus close to the query. There are many efforts in this direction, mainly applied in question-answering systems. The mapping should consider the hierarchical structure of the taxonomy and the distance between nodes in the taxonomy.

In this section, we summarized our work and went through some future directions that could use the developed methodologies within this thesis as a foundation to build interpretable methods. We contributed to the human-AI field by proposing systems where the interaction between the human computation and the AI model is bidirectional, which was fundamental to ensure the effectiveness of the human-AI team. In establishing such collaborative systems, we paved the way for better collaboration between artificial intelligence and human computation for open-ended data curation.

A Appendix

A.1 Proofs for Chapter Peer Grading the Peer Reviews: A Dual-Role Approach for Lightening the Scholarly Paper Review Process

We apply the same notational conventions as in Chapter 4. We use the symbol $\propto$ to denote that two variables are proportionally related.

A.1.1 Proof of Lemma Incremental Update for Review Conformity

Proof. To minimize the KL divergence, we assume the variational distribution follows the same distribution as the latent variable [195]. For $q(z_i)$, we obtain

$$q(z_i) \propto h_{q(r_g,b_g)}[p(z_i, r, b, \boldsymbol{A}_{i,*}, \mathscr{W})], \tag{A.1}$$

where $h_{q(r_g,b_g)}$ denotes the exponential of expectation $\exp\{\mathbb{E}_x[\log(\cdot)]\}$ with x being a variational distribution. According to the mean field approximation, the probability $p(z_i, r, b, \boldsymbol{A}_{i,*}, \mathscr{W})$ factorizes over $\mathscr{G}_i$ and Eq.(A.19) can be written as:

$$q(z_i) \propto \prod_{g \in \mathscr{G}_i} h_{q(r_g,b_g)}[p(z_i, r_g, b_g, \boldsymbol{A}_{r,g}, \mathscr{W})]. \tag{A.2}$$

where $\mathscr{G}_i$ represents the graders relevant to a review i. By applying the chain rule on the probability $p(z_i, r, b, \boldsymbol{A}_{i,*}, \mathscr{W})$ and keeping only the terms that depend on z_i, we get:

$$q(z_i) \propto p(z_i|\boldsymbol{x}_i, \mathscr{W}) \prod_{g \in \mathscr{G}_i} \underbrace{h_{q(r_g,b_g)}[p(\boldsymbol{A}_{r,g}|z_i, r_g, b_g)]}_{\mathscr{T}_1} \tag{A.3}$$

The term $\mathscr{T}_1$ can be expressed using the probability density function of a Gaussian distribution of $p(\boldsymbol{A}_{r,g}|z_i, r_g, b_g)$ where the logarithm of the Gaussian distribution is given by

$$\log(\mathcal{N}(z_i + b_g, \frac{1}{r_g}) \propto \frac{1}{2}\log r_g - \frac{r_g}{2}(\boldsymbol{A_{i,g}} - z_i - b_g)^2 \tag{A.4}$$

Then, we keep the terms dependent on z_i and apply $\mathbb{E}_{r_g,b_g}$:

$$\mathbb{E}_{r_g,b_g}[\log(\mathcal{N}(z_i + b_g, \frac{1}{r_g})] \propto \mathbb{E}_{r_g,b_g}[\frac{r_g}{2}] \times \mathbb{E}_{r_g,b_g}[(\boldsymbol{A_{i,g}} - z_i - b_g)^2] \tag{A.5}$$

We expand the second term by the square factor and get:

$$(\boldsymbol{A_{i,g}} - z_i - b_g)^2 = \boldsymbol{A}^2_{\boldsymbol{i,g}} + z_i^2 + b_g^2 + 2z_ib_g - 2\boldsymbol{A_{i,g}}z_i - 2\boldsymbol{A_{i,g}}b_g \tag{A.6}$$

We eliminate the terms independent from z_i and apply $\mathbb{E}_{r_g,b_g}$:

$$\begin{aligned}\mathbb{E}_{r_g,b_g}[(\boldsymbol{A_{i,g}} - z_i - b_g)^2] = \mathbb{E}_{r_g,b_g}[z_i^2] + 2\mathbb{E}_{r_g,b_g}[z_i]\mathbb{E}_{r_g,b_g}[b_g] \\ - 2\mathbb{E}_{r_g,b_g}[\boldsymbol{A_{i,g}}]\mathbb{E}_{r_g,b_g}[z_i]\end{aligned} \tag{A.7}$$

Using the properties of b_g distribution and since $\boldsymbol{A_{i,g}}$ and z_i do not depend on b_g, the terms in Eq.(A.7) are expressed as follows:

$$\mathbb{E}_{b_g}[\boldsymbol{A_{i,g}}] = \boldsymbol{A_{i,g}}, \mathbb{E}_{b_g}[z_i^2] = z_i^2, \mathbb{E}_{b_g}[z_i] = z_i, \mathbb{E}_{b_g}[b_g] = m_g \tag{A.8}$$

The first term in Eq.(A.5) is the mean of r_g's distribution, i.e., $\frac{A_g}{B_g}$. We replace the second term by the expressions in Eqs.(A.7)-(A.8):

$$\mathbb{E}_{r_g,b_g}[\log(\mathcal{N}(z_i + b_g, \frac{1}{r_g})] \propto \frac{A_g}{2B_g} \times (z_i^2 + 2z_i(m_g - \boldsymbol{A_{i,g}})) \tag{A.9}$$

We now replace in Eq.(A.3) $p(z_i|\boldsymbol{x}_i, \mathcal{W})$ by the probability density function of z_i and the term $\mathcal{T}_1$ by its simplification in Eq.(A.9).

$$\begin{aligned} q(z_i) &\propto \mathcal{N}(\mu_i, \sigma_i) \prod_{g \in \mathcal{G}_i} \exp\{\frac{A_g}{2B_g} \times (z_i^2 + 2z_i(m_g - \boldsymbol{A_{i,g}}))\} \\ &\propto \exp\{\frac{-1}{2}[(\sum_g \frac{A_g}{B_g} + \frac{1}{\sigma_i^2})z_i^2 - 2(\sum_g \frac{A_g}{B_g}(\boldsymbol{A_{i,g}} - m_g) + \frac{\mu_i}{\sigma_i^2})]z_i\} \\ &\propto \mathcal{N}(\frac{W}{V}, \frac{1}{V}), \end{aligned}$$

where $W = \sum_g \frac{A_g}{B_g}(\boldsymbol{A}_{i,g} - m_g) + \frac{\mu_i}{\sigma_i^2}$ and $V = (\sum_g \frac{A_g}{B_g} + \frac{1}{\sigma_i^2})$, which concludes the proof. □

A.1.2 Proof of Lemma Incremental Update for Grader Reliability

Proof. Following the reasoning in Eq.(A.19)-(A.3) for r_g, we get:

$$q(r_g) \propto p(r_g|A_g, B_g) \prod_{i\in\mathscr{I}_g} \underbrace{h_{q(z_i,b_g)}[p(\boldsymbol{A}_{r,g}|z_i, r_g, b_g)]}_{\mathscr{T}_2} \tag{A.10}$$

To incrementally update the grader reliability, we simplify the term $\mathscr{T}_2$ in Eq.(A.10). First, we use Eq.(A.6) to expand the term $(\boldsymbol{A_{i,g}} - z_i - b_g)^2$ and apply the expectation $\mathbb{E}_{z_i,b_g}(\cdot)$. Then, using the properties of the Gaussian distribution of z_i and b_g, we get:

$$\mathbb{E}_{z_i,b_g}[z_i]{=}\mu_i, \mathbb{E}_{z_i,b_g}[z_i^2]{=}\sigma_i^2, \mathbb{E}_{z_i,b_g}[b_g]{=}m_g, \mathbb{E}_{z_i,b_g}[b_g^2]{=}\frac{1}{\alpha_g} \tag{A.11}$$

The term $\mathbb{E}_{z_i,b_g}[(\boldsymbol{A_{i,g}} - z_i - b_g)^2]$ can be simplified using the expressions in Eq.(A.11). We denote the simplification with M_i

$$M_i = \boldsymbol{A_{i,g}^2} + \sigma_i^2 + \frac{1}{\alpha_g} + 2(\mu_i m_g - \boldsymbol{A_{i,g}}\mu_i - \boldsymbol{A_{i,g}} m_g) \tag{A.12}$$

The expectation of Eq.(A.4) conditioned on z_i and b_g can be simplified using Eq.(A.12):

$$\mathbb{E}_{z_i,b_g}[\log(\mathcal{N}(z_i + b_g, \frac{1}{r_g})] \propto \frac{1}{2}\log r_g - \frac{r_g}{2} M_i \tag{A.13}$$

Now, we can replace the term $\mathscr{T}_2$ in Eq.(A.10) by its expression in Eq.(A.13) and the probability $p(r_g|A_g, B_g)$ by its density function.

$$\begin{aligned}
q(r_g) &\propto \Gamma(A_g, B_g) \prod_{i\in\mathscr{I}_g} \exp\{\frac{1}{2}\log r_g - \frac{r_g}{2} M_i\} \\
&\propto \frac{1}{\Gamma(A_g)} B_g^{A_g} r_g^{A_g + \frac{|\mathscr{I}_g|}{2} - 1} \exp\{-(b_g + \frac{1}{2}\sum_{i\in\mathscr{I}_g} M_i) r_g\} \\
&\propto Gamma(X, Y)
\end{aligned}$$

where $X = A_g {+} \frac{|\mathscr{I}_g|}{2}$ and $Y = B_g {+} \frac{1}{2}(\frac{|\mathscr{I}_g|}{\alpha_g} {+} \sum_i [\boldsymbol{A}_{i,g}^2 + \sigma_i^2 + 2\mu_i(m_g - \boldsymbol{A}_{i,g}) - 2\boldsymbol{A}_{i,g} m_g])$ which concludes the proof. □

A.1.3 Proof of Lemma Incremental Update for Grader Bias

Proof. Following the reasoning in Eq.(A.19)-(A.3) for b_g, we get:

$$q(b_g) \propto p(b_g|m_g, \alpha_g) \prod_{i\in\mathscr{I}_g} \underbrace{h_{q(z_i,r_g)}[p(\boldsymbol{A}_{r,g}|z_i, r_g, b_g)]}_{\mathscr{T}_3} \tag{A.14}$$

To incrementally update worker's bias, we simplify the term $\mathscr{T}_3$ in Eq.(A.14). In order to do that, we use Eq.(A.6) to expand the term $(\boldsymbol{A_{i,g}} - z_i - b_g)^2$ and apply the expectation $\mathbb{E}_{z_i,r_g}(\cdot)$. Then, we use the properties of the Gaussian distribution of z_i and the independence property of b_g with respect to z_i and r_g and get:

$$\mathbb{E}_{z_i,r_g}[z_i] = \mu_i, \mathbb{E}_{z_i,r_g}[z_i^2] = \sigma_i^2, \mathbb{E}_{z_i,r_g}[b_g] = b_g, \mathbb{E}_{z_i,r_g}[b_g^2] = b_g^2 \tag{A.15}$$

Using the expressions in Eq.(A.15) and by eliminating the terms that do not depend on b_j, the expectation $\mathbb{E}_{z_i,b_g}[(\boldsymbol{A_{i,g}} - z_i - b_g)^2]$ can be simplified as follows.

$$\mathbb{E}_{z_i,b_g}[(\boldsymbol{A_{i,g}} - z_i - b_g)^2] = b_g^2 + 2(\mu_i - \boldsymbol{A_{i,g}})b_g \tag{A.16}$$

The expectation term $\mathbb{E}_{z_i,r_g}[\log(\mathcal{N}(z_i + r_g, \frac{1}{r_g})]$ is given by:

$$\mathbb{E}_{z_i,r_g}[\log(\mathcal{N}(z_i + r_g, \frac{1}{r_g})] \propto -\frac{A_g}{2B_g}(b_g^2 + 2(\mu_i - \boldsymbol{A_{i,g}})b_g), \tag{A.17}$$

where $\frac{A_g}{B_g}$ is the mean of the reliability density function. We can now replace the term $\mathscr{T}_3$ in Eq.(A.14) by its expression in Eq.(A.17) and the probability $p(b_g|m_g, \alpha_g)$ by its density function.

$$\begin{aligned} q(b_g) &\propto \mathcal{N}(m_g, b_g) \prod_{i \in \mathscr{I}_g} \exp\{-\frac{A_g}{2B_g}(b_g^2 + 2(\mu_i - \boldsymbol{A_{i,g}})b_g))\} \\ &\propto \exp(-\frac{1}{2}((\frac{A_g|\mathscr{I}_g|}{b_g} + \alpha_g)b_g^2 - 2[\alpha_g m_g + \frac{A_g}{b_g}\sum_i(\boldsymbol{A_{i,j}} - \mu_i)]b_g)) \\ &\propto \mathcal{N}(\frac{L}{K}, \frac{1}{K}) \end{aligned}$$

where $K = \frac{A_g|\mathscr{I}_g|}{B_g} + \alpha_g$ and $L = \alpha_g m_g + \frac{A_g}{B_g}\sum_r(\boldsymbol{A}_{i,g} - \mu_i)$ which concludes the proof. □

A.2 Proofs for Chapter MARTA: Leveraging Human Rationales for Explainable Text Classification

In this section, we present the proofs of our lemmas in Chapter 5. We use the same notational conventions as in the chapter.

A.2.1 Proof of Lemma Incremental Document Classification

The true label distribution $q(z_i)$ can be incrementally computed using the predicted label by the attention-based model θ_i, and the parameters m_j and n_j of the worker reliability distribution r_j.

$$q(z_i = 1) \propto \begin{cases} \theta_i \prod_{j \in \mathscr{J}_i} \exp\{\Psi(n_j) - \Psi(m_j + n_j)\}, & \text{if } \mathbf{A}_{i,j} = 0, \\ \theta_i \prod_{j \in \mathscr{J}_i} \exp\{\Psi(m_j) - \Psi(m_j + n_j)\}, & \text{if } \mathbf{A}_{i,j} = 1, \end{cases} \tag{A.18}$$

where Ψ is the Digamma function. If $q(z_i = 0)$, then we replace θ_i by $1 - \theta_i$.

Proof. To minimize the KL divergence, we assume the variational distribution follows the same distribution as the latent variable [195]. For $q(z_i)$, we obtain Eq. (A.19).

$$q(z_i) \propto g_{q(r_j,\alpha_s)}[p(z_i, r, \alpha, \mathbf{A}_{i,*}, \mathbf{B}, \mathscr{W})], \tag{A.19}$$

where, we use $g_x(\cdot)$ to denote the exponential of expectation term $\exp\{\mathbb{E}_x[\log(\cdot)]\}$ with x being a variational distribution and $x \propto y$ to denote that the two variables x and y are proportionally related (i.e., $x = ky$, where k is a constant). According to the mean field approximation, the probability $p(z_i, r, \alpha, \mathbf{A}_{i,*}, \mathbf{B}, \mathscr{W})$ factorizes over $\mathscr{S}_i$ and $\mathscr{J}_i$ and Eq. (A.19) can be written as Eq. (A.20).

$$q(z_i) \propto g_{q(r_j,\alpha_s)}[\prod_{s \in \mathscr{S}_i, j \in \mathscr{J}_i} p(z_i, r_j, \alpha_s, \mathbf{A}_{i,j}, \mathbf{B}_{s,j}, \mathscr{W})], \tag{A.20}$$

where $\mathscr{S}_i$ and $\mathscr{J}_i$ represent respectively the sentences and the workers relevant to document i. Using the properties of the exponential and the logarithm functions in $g_x(\cdot)$, we get Eq. (A.21):

$$q(z_i) \propto \prod_{s \in \mathscr{S}_i, j \in \mathscr{J}_i} g_{q(r_j,\alpha_s)}[p(z_i, r_j, \alpha_s, \mathbf{A}_{i,j}, \mathbf{B}_{s,j}, \mathscr{W})]. \tag{A.21}$$

By applying the chain rule on $p(z_i, r_j, \alpha_s, \mathbf{A}_{i,j}, \mathbf{B}_{s,j}, \mathscr{W})$, we obtain Eq. (A.22).

$$\begin{aligned} p(z_i, r_j, \alpha_s, \mathbf{A}_{i,j}, \mathbf{B}_{s,j}, \mathscr{W}) = {} & p(\mathbf{A}_{i,j} | z_i, r_j) \times p(\mathbf{B}_{s,j} | r_j, \alpha_s) \\ & \times p(z_i | \mathbf{v}_s, \mathscr{W}) \times p(r_j | m_j, n_j) \\ & \times p(\alpha_s | \mathbf{v}_s, \mathscr{W}_a). \end{aligned} \tag{A.22}$$

Next we replace the probability in Eq. (A.21) by the chain rule and keep only the terms that depend on z_i, we get Eq. (A.23).

$$q(z_i) \propto \prod_{s \in \mathscr{S}_i, j \in \mathscr{J}_i} g_{q(r_j,\alpha_s)}[p(z_i | \mathbf{v}_s, \mathscr{W}) p(\mathbf{A}_{i,j} | z_i, r_j)]. \tag{A.23}$$

As the probability $p(z_i | \mathbf{v}_s, \mathscr{W})$ is independent from $p(\mathbf{A}_{i,j} | z_i, r_j)$, we get the following:

$$q(z_i) \propto \prod_{s \in \mathscr{S}_i, j \in \mathscr{J}_i} g_{q(r_j,\alpha_s)}[p(z_i | \mathbf{v}_s, \mathscr{W})] g_{q(r_j,\alpha_s)}[p(\mathbf{A}_{i,j} | z_i, r_j)]. \tag{A.24}$$

Since the probability of z_i does not depend on r_j and α_s, we can simplify Eq. (A.24) to the following:

$$\begin{aligned} q(z_i) &\propto \prod_{s\in\mathscr{S}_i, j\in\mathscr{J}_i} p(z_i|\mathbf{v}_s,\mathscr{W}) g_{q(r_j,\alpha_s)}[p(\mathbf{A}_{i,j}|z_i,r_j)] \\ &\propto \prod_{s\in\mathscr{S}_i} p(z_i|\mathbf{v}_s,\mathscr{W}) \prod_{j\in\mathscr{J}_i} g_{q(r_j)}[p(\mathbf{A}_{i,j}|z_i,r_j)] \\ &\propto p(z_i|\mathbf{V},\mathscr{W}) \prod_{j\in\mathscr{J}_i} g_{q(r_j)}[p(\mathbf{A}_{i,j}|z_i,r_j)]. \end{aligned} \tag{A.25}$$

We show the proof only for $z_i = 1$ since the proof for $z_i = 0$ follow similarly. Using the definition of $q(z_i)$, we have Eq.(A.26):

$$p(z_i = 1|\mathbf{V},\mathscr{W}) = \theta_i. \tag{A.26}$$

We substitute the probability $p(z_i|\mathbf{V},\mathscr{W})$ and $p(\mathbf{A}_{i,j}|z_i,r_j)$ by their respective definitions in Eq.(A.26) and Eq.(8) from Section Method:

$$q(z_i = 1) \propto \begin{cases} \theta_i \prod_{j\in\mathscr{J}_i} g_{q(r_j)}[1-r_j], & \text{if } \mathbf{A}_{i,j} = 0, \\ \theta_i \prod_{j\in\mathscr{J}_i} g_{q(r_j)}[r_j], & \text{if } \mathbf{A}_{i,j} = 1. \end{cases} \tag{A.27}$$

By computing the geometric mean of the beta distribution, we can evaluate the exponential terms $g_{q(r_j)}[\cdot]$ as follows:

$$\begin{aligned} g_{q(r_j)}[1-r_j] &= \exp\{\Psi(n_j) - \Psi(m_j+n_j)\}, \\ g_{q(r_j)}[r_j] &= \exp\{\Psi(m_j) - \Psi(m_j+n_j)\}. \end{aligned} \tag{A.28}$$

Putting (A.28) into (A.27), the update equation can be simplified:

$$q(z_i = 1) \propto \begin{cases} \theta_i \prod_{j\in\mathscr{J}_i} \exp\{\Psi(n_j) - \Psi(m_j+n_j)\}, & \text{if } \mathbf{A}_{i,j} = 0, \\ \theta_i \prod_{j\in\mathscr{J}_i} \exp\{\Psi(m_j) - \Psi(m_j+n_j)\}, & \text{if } \mathbf{A}_{i,j} = 1, \end{cases} \tag{A.29}$$

which concludes the proof. □

A.2.2 Proof of Lemma Incremental Sentence Importance

The importance of a sentence for document classification can be incrementally computed using the attributed attention weight by the attention-based model a_s and the parameters m_j and n_j of the worker reliability distribution r_j.

$$q(\alpha_s = 1) \propto \begin{cases} a_s \prod_{j\in\mathscr{J}_s} \exp\{\Psi(n_j) - \Psi(m_j+n_j)\}, & \text{if } \mathbf{B}_{s,j} = 0, \\ a_s \prod_{j\in\mathscr{J}_s} \exp\{\Psi(m_j) - \Psi(m_j+n_j)\}, & \text{if } \mathbf{B}_{s,j} = 1. \end{cases} \tag{A.30}$$

Proof. Similarly to Eq.(A.19), we assume the variational distribution $q(\alpha_s)$ follows the same distribution as the latent variable α_s. We obtain Eq. (A.31):

$$q(\alpha_s) \propto g_{q(r_j,z_i)}[p(z,r,\alpha_s,\mathbf{A},\mathbf{B}_{s,*},\mathscr{W})]. \tag{A.31}$$

Using the mean field approximation, the probability $p(z,r,\alpha_s,\mathbf{A},\mathbf{B}_{s,*},\mathscr{W})$ factorizes as follows:

$$q(\alpha_s) \propto g_{q(r_j,z_i)}[\prod_{j\in\mathscr{J}_s, i\in\mathscr{I}_s} p(z_i,r_j,\alpha_s,\mathbf{A}_{i,j},\mathbf{B}_{s,j},\mathscr{W})], \tag{A.32}$$

where $\mathscr{J}_s$ and $\mathscr{I}_s$ represent the workers and the documents relevant to sentence s. Using the properties of the exponential and logarithm functions, we get:

$$q(\alpha_s) \propto \prod_{j\in\mathscr{J}_s, i\in\mathscr{I}_s} g_{q(r_j,z_i)}[p(z_i,r_j,\alpha_s,\mathbf{A}_{i,j},\mathbf{B}_{s,j},\mathscr{W})]. \tag{A.33}$$

By applying the chain rule of Eq. (A.22) and keeping only the terms that depend on α_s, Eq. (A.33) is simplified as follows:

$$q(\alpha_s) = \prod_{j\in\mathscr{J}_s} g_{q(r_j)}(p(\alpha_s|\mathbf{v}_s,\mathscr{W}_a)\,p(\mathbf{B}_{s,j}|\alpha_s,r_j)). \tag{A.34}$$

Since α_s does not depend on r_j, we get:

$$q(\alpha_s) = p(\alpha_s|\mathbf{v}_s,\mathscr{W}_a)\prod_{j\in\mathscr{J}_s} g_{q(r_j)}[p(\mathbf{B}_{s,j}|\alpha_s,r_j)]. \tag{A.35}$$

Here as well, we show the proof for $\alpha_s = 1$, as the proof for $\alpha_s = 0$ follows similarly. Using the definition of α_s, we get:

$$p(\alpha_s = 1|\mathbf{v}_s,\mathscr{W}_a) = a_s. \tag{A.36}$$

Using the definition of $p(\alpha_s = 1|\mathbf{v}_s,\mathscr{W}_a)$ in Eq. (A.36) and the definition of $p(\mathbf{B}_{s,j}|\alpha_s,r_j)$ from Section Method Eq. (7), we get the following:

$$q(\alpha_s = 1) \propto \begin{cases} a_s \prod_{j\in\mathscr{J}_s} g_{q(r_j)}[1-r_j], & \text{if } \mathbf{B}_{s,j} = 0, \\ a_s \prod_{j\in\mathscr{J}_s} g_{q(r_j)}[r_j], & \text{if } \mathbf{B}_{s,j} = 1. \end{cases} \tag{A.37}$$

We replace in Eq. (A.37), $g_{q(r_j)}[1-r_j]$ and $g_{q(r_j)}[r_j]$ by the expressions given in Eq. (A.28):

$$q(\alpha_s = 1) \propto \begin{cases} a_s \prod_{j\in\mathscr{J}_s} \exp\{\Psi(n_j) - \Psi(m_j+n_j)\}, & \text{if } \mathbf{B}_{s,j} = 0, \\ a_s \prod_{j\in\mathscr{J}_s} \exp\{\Psi(m_j) - \Psi(m_j+n_j)\}, & \text{if } \mathbf{B}_{s,j} = 1, \end{cases} \tag{A.38}$$

which concludes the proof. □

A.2.3 Proof of Lemma Incremental Worker Reliability

The worker reliability distribution $q(r_j)$ can be incrementally computed using her annotation and rationale quality, and the reliability parameters m_j and n_j from the previous iteration.

$$q(r_j) \propto \begin{cases} Beta(m'_j + \sum_{s \in \mathscr{S}_j} (1 - a_s), (n'_j + \sum_{s \in \mathscr{S}_j} a_s), \text{if } \mathbf{B}_{s,j} = 0, \\ Beta(m'_j + \sum_{s \in \mathscr{S}_j} a_s, n'_j + \sum_{s \in \mathscr{S}_j} (1 - a_s)), \text{if } \mathbf{B}_{s,j} = 1, \end{cases} \tag{A.39}$$

where $m'_j = m_j + \sum_{i \in \mathscr{I}_j} \theta_i$ and $n'_j = n_j + \sum_{i \in \mathscr{I}_j} (1 - \theta_i)$, if $\mathbf{A}_{i,j} = 1$ and $m'_j = m_j + \sum_{i \in \mathscr{I}_j} (1 - \theta_i)$ and $n'_j = n_j + \sum_{i \in \mathscr{I}_j} \theta_i$, if $\mathbf{A}_{i,j} = 0$.

Proof. We assume the variational distribution $q(r_j)$ follows the same distribution as the latent variable r_j which translates to Eq.(A.40).

$$q(r_j) \propto g_{q(z_i, \alpha_s)}[p(z, r_j, \alpha, \mathbf{A}_{*,j}, \mathbf{B}_{*,j}, \mathscr{W})]. \tag{A.40}$$

Using the mean field approximation, the probability $p(z, r_j, \alpha, \mathbf{A}_{*,j}, \mathbf{B}_{*,j}, \mathscr{W})$ factorizes over $|\mathscr{I}_j|$ and $|\mathscr{S}_j|$ representing respectively the documents and the sentences relevant to worker j.

$$q(r_j) \propto g_{q(z_i, \alpha_s)}[\prod_{i \in \mathscr{I}_j, s \in \mathscr{S}_j} p(z_i, r_j, \alpha_s, \mathbf{A}_{i,j}, \mathbf{B}_{s,j}, \mathscr{W})]. \tag{A.41}$$

Using the properties of the exponential and logarithm, we get:

$$q(r_j) \propto \prod_{i \in \mathscr{I}_j, s \in \mathscr{S}_j} g_{q(z_i, \alpha_s)}[p(z_i, r_j, \alpha_s, \mathbf{A}_{i,j}, \mathbf{B}_{s,j}, \mathscr{W})]. \tag{A.42}$$

By applying the chain rule in Eq.(A.22) and keeping only the terms that depend on r_j, we get:

$$q(r_j) \propto \prod_{i \in \mathscr{I}_j, s \in \mathscr{S}_j} g_{q(z_i, \alpha_s)}[p(r_j | m_j, n_j) p(\mathbf{A}_{i,j} | z_i, r_j) p(\mathbf{B}_{s,j} | r_j, \alpha_s)] \tag{A.43}$$

Since the probability $p(r_j | m_j, n_j)$ does not depend on z_i and α_s, we can simplify Eq.(A.43) to the following:

$$q(r_j) \propto p(r_j | m_j, n_j) \times \prod_{i \in \mathscr{I}_j, s \in \mathscr{S}_j} g_{q(z_i, \alpha_s)}[p(\mathbf{A}_{i,j} | z_i, r_j) p(\mathbf{B}_{s,j} | r_j, \alpha_s)] \tag{A.44}$$

By replacing the exponential of expectation term $g_x(\cdot)$ by its expression, we can simplify the term $g_{q(z_i, \alpha_s)}[p(\mathbf{A}_{i,j} | z_i, r_j) p(\mathbf{B}_{s,j} | r_j, \alpha_s)]$ as follows:

$$\begin{aligned} g_{q(z_i, \alpha_s)}[p(\mathbf{A}_{i,j} | z_i, r_j) p(\mathbf{B}_{s,j} | r_j, \alpha_s)] &= \exp\{\mathbb{E}_{q(z_i, \alpha_s)}[\log(p(\mathbf{A}_{i,j} | z_i, r_j) p(\mathbf{B}_{s,j} | r_j, \alpha_s))]\} \\ &= \exp\{\mathbb{E}_{q(z_i, \alpha_s)}[\log(p(\mathbf{A}_{i,j} | z_i, r_j)) + \log(p(\mathbf{B}_{s,j} | r_j, \alpha_s))]\} \\ &= \exp\{\mathbb{E}_{q(z_i)}[\log(p(\mathbf{A}_{i,j} | z_i, r_j))] + \mathbb{E}_{q(\alpha_s)}[\log(p(\mathbf{B}_{s,j} | r_j, \alpha_s))]\} \\ &= g_{q(z_i)}[p(\mathbf{A}_{i,j} | z_i, r_j))] \times g_{q(\alpha_s)}[p(\mathbf{B}_{s,j} | r_j, \alpha_s)] \end{aligned} \tag{A.45}$$

A.2 Proofs for Chapter MARTA: Leveraging Human Rationales for Explainable Text Classification

We distinguish two main cases depending on the values of $\mathbf{A}_{i,j} \in \{0,1\}$. Let's start with the case where $\mathbf{A}_{i,j} = 1$. This case covers all documents i a worker j has annotated as positive. In such a case, the probability $p(\mathbf{A}_{i,j}|z_i, r_j)$ can be written as a function of θ_i and r_j as given by eq.(A.46):

$$g_{q(z_i)}[p(\mathbf{A}_{i,j} = 1|z_i, r_j) = r_j^{\theta_i}(1-r_j)^{(1-\theta_i)} \tag{A.46}$$

The documents annotated as positive by worker j include two sets of sentences: a set of sentences annotated as rationales, where $\mathbf{B}_{s,j} = 1$, and a set of non-rationales where $\mathbf{B}_{s,j} = 0$. The probability $p(\mathbf{B}_{s,j}|r_j, \alpha_s)$ is simplified as follows:

$$g_{q(\alpha_s)}[p(\mathbf{B}_{s,j}|r_j, \alpha_s)] \propto \begin{cases} r_j^{a_s}(1-r_j)^{(1-a_s)}, \text{if } \mathbf{B}_{s,j} = 1, \\ r_j^{(1-a_s)}(1-r_j)^{a_s}, \text{if } \mathbf{B}_{s,j} = 0, \end{cases} \tag{A.47}$$

If we take the case of $\mathbf{B}_{s,j} = 1$, by putting (A.46) and (A.47) in (A.43), we get:

$$q(r_j) \propto p(r_j|m_j, n_j) \prod_{i \in \mathscr{I}_j, s \in \mathscr{S}_j} r_j^{\theta_i + a_s}(1-r_j)^{(1-\theta_i+1-a_s)} \tag{A.48}$$

The probability $p(r_j|m_j, n_j)$ is a beta distribution and hence its probability density function is given by Eq.(A.49).

$$\begin{aligned} p(r_j|m_j, n_j) &= Beta(m_j, n_j) \\ &= r_j^{(m_j-1)}(1-r_j)^{(n_j-1)} \end{aligned} \tag{A.49}$$

By substituting the probability $p(r_j|m_j, n_j)$ by its expression in Eq.(A.49), we get the following result:

$$\begin{aligned} q(r_j) &\propto r_j^{(m_j-1)}(1-r_j)^{(n_j-1)} \times \prod_{i \in \mathscr{I}_j, s \in \mathscr{S}_j} r_j^{\theta_i + a_s}(1-r_j)^{(1-\theta_i+1-a_s)} \\ &\propto r_j^{m_j - 1 + \sum_{i \in \mathscr{I}_j} \theta_i + \sum_{s \in \mathscr{S}_j} a_s} \times (1-r_j)^{(n_j - 1 + \sum_{i \in \mathscr{I}_j}(1-\theta_i) + \sum_{s \in \mathscr{S}_j}(1-a_s))} \end{aligned} \tag{A.50}$$

Similarly for $\mathbf{B}_{s,j} = 0$, by putting (A.46), (A.47) and (A.49) in (A.44), we get:

$$\begin{aligned} q(r_j) &\propto r_j^{(m_j-1)}(1-r_j)^{(n_j-1)} \times \prod_{i \in \mathscr{I}_j, s \in \mathscr{S}_j} r_j^{\theta_i + 1 - a_s}(1-r_j)^{(1-\theta_i+a_s)} \\ &\propto r_j^{m_j - 1 + \sum_{i \in \mathscr{I}_j} \theta_i + \sum_{s \in \mathscr{S}_j}(1-a_s)} \times (1-r_j)^{(n_j - 1 + \sum_{i \in \mathscr{I}_j}(1-\theta_i) + \sum_{s \in \mathscr{S}_j} a_s)} \end{aligned} \tag{A.51}$$

Using Eq.(A.48) and Eq.(A.51), we get the updating rules for when the documents are labeled as positive by a worker, i.e., $\mathbf{A}_{i,j} = 1$:

$$q(r_j) \propto \begin{cases} r_j^{m_j - 1 + \sum_i \theta_i + \sum_s a_s}(1-r_j)^{(n_j - 1 + \sum_i (1-\theta_i) + \sum_s (1-a_s))}, \text{if } \mathbf{A}_{i,j} = \mathbf{B}_{s,j}, \\ r_j^{(m_j - 1 + \sum_i \theta_i + \sum_s (1-a_s))}(1-r_j)^{(n_j - 1 + \sum_i (1-\theta_i) + \sum_s a_s)}, \text{if } \mathbf{A}_{i,j} \neq \mathbf{B}_{s,j}, \end{cases} \tag{A.52}$$

where the documents i and sentences s are relevant to worker j, i.e., $i \in \mathscr{I}_j$ and $s \in \mathscr{S}_j$. For the second case, where $\mathbf{A}_{i,j} = 0$, the term $g_{q(z_i)}[p(\mathbf{A}_{i,j}|z_i, r_j)$ can be written as a function of θ_i and r_j as given by eq.(A.53):

$$g_{q(z_i)}[p(\mathbf{A}_{i,j} = 0|z_i, r_j)] = r_j^{(1-\theta_i)}(1 - r_j)^{\theta_i} \tag{A.53}$$

Using the same reasoning of distinguishing the two cases $\mathbf{B}_{s,j} = 0$ and $\mathbf{B}_{s,j} = 1$ and then replacing the probability $p(r_j|m_j, n_j)$ by its probability density function, we get the following results:

$$q(r_j) \propto \begin{cases} r_j^{(m_j-1+\sum_i(1-\theta_i)+\sum_s(1-a_s))}(1 - r_j)^{(n_j-1+\sum_i\theta_i+\sum_s a_s)}, \text{if } \mathbf{A}_{i,j}{=}\mathbf{B}_{s,j}, \\ r_j^{m_j-1+\sum_i(1-\theta_i)+\sum_s a_s}(1 - r_j)^{(n_j-1+\sum_i\theta_i+\sum_s(1-a_s))}, \text{if } \mathbf{A}_{i,j}{\neq}\mathbf{B}_{s,j}, \end{cases} \tag{A.54}$$

which concludes the proof. □

Bibliography

[1] ACM. Policy on roles and responsibilities in acm publishing. https://www.acm.org/publications/policies/roles-and-responsibilities, 2018. Accessed: 2020-02-26.

[2] Nitin Agarwal, Huan Liu, Lei Tang, and Philip S. Yu. Identifying the influential bloggers in a community. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM)*, pages 207–218, Palo Alto, California, USA, 2008. ACM.

[3] Anastasia Ailamaki, Periklis Chrysogelos, Amol Deshpande, and Tim Kraska. The sigmod 2019 research track reviewing system. *ACM SIGMOD Record*, 48(2):47–54, 2019.

[4] Bruce Alberts, Brooks Hanson, and Katrina L. Kelner. Reviewing peer review. *Science*, 321(5885):15–15, 2008.

[5] Ammar Ammar and Devavrat Shah. Efficient rank aggregation using partial data. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):355–366, 2012.

[6] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*. OpenReview.net, 2018.

[7] Stefanos Angelidis and Mirella Lapata. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31, 2018.

[8] Appen. About us. https://appen.com/about-us/, 2022. Accessed: 2022-05-29.

[9] Ines Arous, Jie Yang, Mourad Khayati, and Philippe Cudré-Mauroux. Opencrowd: A human-ai collaborative approach for finding social influencers via open-ended answers aggregation. In *Proceedings of The Web Conference 2020*, pages 1851–1862, 2020.

[10] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. “ what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142, 2017.

[11] Kyohei Atarashi, Satoshi Oyama, and Masahito Kurihara. Semi-supervised learning from crowds using deep generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[12] Yukino Baba and Hisashi Kashima. Statistical quality estimation for general crowdsourcing tasks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–562, 2013.

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[14] Agathe Balayn, Panagiotis Soilis, Christoph Lofi, Jie Yang, and Alessandro Bozzon. What do you mean? interpreting image classification with crowdsourced concept extraction and analysis. In *Proceedings of the Web Conference 2021*, pages 1937–1948, 2021.

[15] Agathe Balayn, Gaole He, Andrea Hu, Jie Yang, and Ujwal Gadiraju. Ready player one! eliciting diverse knowledge using a configurable game. In *Proceedings of the ACM Web Conference 2022*, pages 1709–1719, 2022.

[16] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. Beyond accuracy: The role of mental models in human-ai team performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 2–11, 2019.

[17] Yujia Bao, Shiyu Chang, Mo Yu, and Regina Barzilay. Deriving machine attention from human rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1903–1913. Association for Computational Linguistics, 2018.

[18] Shane Barker. The ultimate guide to micro-influencers. `https://shanebarker.com/blog/micro-influencers-guide/`, 2019. Accessed: 2019-10-11.

[19] Hannah Bast. How objective is peer review? `https://cacm.acm.org/blogs/blog-cacm/248824-how-objective-is-peer-review`, 2020. Accessed: 2021-02-02.

[20] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: Pretrained language model for scientific text. In *EMNLP*, pages 3613–3618, USA, 2019. Association for Computational Linguistics.

[21] Phil Bernstein, Michael Brodie, Stefano Ceri, David DeWitt, Mike Franklin, Hector Garcia-Molina, Jim Gray, Jerry Held, Joe Hellerstein, HV Jagadish, et al. The asilomar report on database research. *ACM Sigmod record*, 27(4):74–80, 1998.

[22] Rajat Bhatnagar, Ananya Ganesh, and Katharina Kann. Don't rule out monolingual speakers: A method for crowdsourcing machine translation data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1099–1106, 2021.

[23] Bin Bi, Yuanyuan Tian, Yannis Sismanis, Andrey Balmin, and Junghoo Cho. Scalable topic-specific influence analysis on microblogs. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 513–522, New York, NY, USA, 2014. ACM.

[24] Roi Blanco, Harry Halpin, Daniel M Herzig, Peter Mika, Jeffrey Pound, Henry S Thompson, and Thanh Tran Duc. Repeatable and reliable search system evaluation using crowdsourcing. In *SIGIR*, pages 923–932, USA, 2011. ACM.

[25] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[26] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[27] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295, 2012.

[28] Kendrick Boyd, Kevin H Eng, and C David Page. Area under the precision-recall curve: point estimates and confidence intervals. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 451–466, Prague, Czech Republic, 2013. Springer.

[29] Alessandro Bozzon, Marco Brambilla, and Stefano Ceri. Answering search queries with crowdsearcher. In *Proceedings of the 21st international conference on World Wide Web*, pages 1009–1018, 2012.

[30] Steven Burrows, Iryna Gurevych, and Benno Stein. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25(1): 60–117, 2015.

[31] Carrie J Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. " hello ai": uncovering the onboarding needs of medical practitioners for human-ai collaborative decision-making. *Proceedings of the ACM on Human-computer Interaction*, 3(CSCW):1–24, 2019.

[32] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, pages 9539–9549, 2018.

[33] Yu-Rong Cao, Xiao-Han Li, Jia-Yu Pan, and Wen-Chieh Lin. Visguide: User-oriented recommendations for data event extraction. In *CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2022.

[34] Alejandro Uriel Carbonara, Anupam Datta, Arunesh Sinha, and Yair Zick. Incentivizing peer grading in moocs: an audit game approach. In *IJCAI*, pages 497–503, USA, 2015. AAAI Press.

[35] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Fourth International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 10–17, Washington, DC, USA, 2010. The AAAI Press.

[36] Chengliang Chai, Lei Cao, Guoliang Li, Jian Li, Yuyu Luo, and Samuel Madden. Human-in-the-loop outlier detection. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 19–33, 2020.

[37] NeurIPS 2020 Program Chairs. Getting started with neurips 2020. https://medium.com/@NeurIPSConf/getting-started-with-neurips-2020-e350f9b39c28, 2020. Accessed: 2021-02-02.

[38] S. Chang, Y. Zhang, Mo Yu, and T. Jaakkola. A game theoretic approach to class-wise selective rationalization. In *Advances in neural information processing systems*, pages 10055–10065, 2019.

[39] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi S Jaakkola. Invariant rationalization. *CoRR*, abs/2003.09772, 2020.

[40] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. *CoRR*, abs/1808.02610, 2018.

[41] Yuyan Chen, Yanghua Xiao, and Bang Liu. Grow-and-clip: Informative-yet-concise evidence distillation for answer explanation. *arXiv preprint arXiv:2201.05088*, 2022.

[42] Justin Cheng and Michael S Bernstein. Flock: Hybrid crowd-machine learning classifiers. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 600–611, 2015.

[43] Zhiyuan Cheng, James Caverlee, Himanshu Barthwal, and Vandana Bachani. Who is the barbecue king of texas?: A geo-spatial approach to finding local experts on twitter. In *Proceedings of the 37th ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 335–344, Gold Coast, Queensland, Australia, 2014. ACM.

[44] Rishi Chhatwal, Peter Gronvall, Nathaniel Huber-Fliflet, Robert Keeling, Jianping Zhang, and Haozhen Zhao. Explainable text classification in legal document review a case study of explainable predictive coding. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1905–1911. IEEE, 2018.

[45] Sven Coppers, Kris Luyten, Davy Vanacken, David Navarre, Philippe Palanque, and Christine Gris. Fortunettes: feedforward about the future state of gui widgets. *Proceedings of the ACM on Human-Computer Interaction*, 3(EICS):1–20, 2019.

[46] Alvaro HC Correia and Freddy Lecue. Human-in-the-loop feature selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2438–2445, 2019.

[47] Laurie Cutrone, Maiga Chang, et al. Auto-assessor: computerized assessment system for marking student's short-answers automatically. In *2011 IEEE International Conference on Technology for Education*, pages 81–88. IEEE, 2011.

[48] Peng Dai, Daniel Sabey Weld, et al. Decision-theoretic control of crowd-sourced workflows. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[49] P. Dawid, A. M. Skene, A. P. Dawidt, and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28:20–28, 1979.

[50] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st International Conference on World Wide Web (WWW)*, pages 469–478, Lyon, France, 2012. ACM.

[51] Hongbo Deng, Irwin King, and Michael R Lyu. Formal models for expert finding on dblp bibliography data. In *ICDM*, pages 163–172, USA, 2008. IEEE.

[52] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[53] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[54] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G Ipeirotis, and Philippe Cudré-Mauroux. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proceedings of the 24th international conference on world wide web*, pages 238–247, 2015.

[55] Yue Dong, Andrei Mircea, and Jackie Chi Kit Cheung. Discourse-aware unsupervised summarization for long scientific documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1089–1102, 2021.

[56] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *CoRR*, abs/1702.08608, 2017.

[57] Susan T Dumais and Jakob Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *SIGIR*, pages 233–244, USA, 1992. ACM.

[58] Laura J Falkenberg and Patricia A Soranno. Reviewing reviews: An evaluation of peer reviews of journal article submissions. *Limnology and Oceanography Bulletin*, 27(1): 1–5, 2018.

[59] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. icrowd: An adaptive crowdsourcing framework. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1015–1030, Melbourne, Victoria, Australia, 2015. ACM.

[60] Ju Fan, Jiarong Qiu, Yuchen Li, Qingfei Meng, Dongxiang Zhang, Guoliang Li, Kian-Lee Tan, and Xiaoyong Du. Octopus: An online topic-aware influence analysis system for social networks. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1569–1572, Paris, France, 2018. IEEE Computer Society.

[61] Hui Fang and ChengXiang Zhai. Probabilistic models for expert finding. In *ECIR*, pages 418–430, Switzerland, 2007. Springer.

[62] Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 61–72, 2011.

[63] Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

[64] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov): 933–969, 2003.

[65] Casalino Gabriella, Barbara Cafarelli, Emiliano Del Gobbo, Fontanella Lara, Luca Grilli, Alfonso Guarino, Pierpaolo Limone, Schicchi Daniele, and Taibi Davide. Framing automatic grading techniques for open-ended questionnaires responses. a short survey. In *Second Workshop on Technology Enhanced Learning Environments for Blended Education-The Italian E-Learning Conference 2021 (teleXbe 2021)*, volume 3025, pages 1–20. CEUR Workshop Proceedings-, 2021.

[66] Yang Gao, Steffen Eger, Ilia Kuznetsov, Iryna Gurevych, and Yusuke Miyao. Does my rebuttal matter? insights from a major nlp conference. In *NAACL-HLT*, pages 1274–1290, USA, 2019. Association for Computational Linguistics.

[67] Zihan Gao and Jiepu Jiang. Evaluating human-ai hybrid conversational systems with chatbot message suggestions. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 534–544, 2021.

[68] Hector Garcia-Molina, Manas Joglekar, Adam Marcus, Aditya Parameswaran, and Vasilis Verroios. Challenges in data crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):901–911, 2016.

[69] Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, volume 36, Quebec City, Canada, 2014. cognitivesciencesociety.org.

[70] Jonathan Grudin. From tool to partner: The evolution of human-computer interaction. *Synthesis Lectures on Human-Centered Interaction*, 10(1):i–183, 2017.

[71] Behnam Hajian and Tony White. Modelling influence in a social network: Metrics and evaluation. In *2011 IEEE Third International Conference on Privacy, Security, Risk and*

Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), pages 497–500, Boston,MA, USA, 2011. IEEE Computer Society.

[72] I Hames. Cope ethical guidelines for peer reviewers. *COPE Council*, 1:1–5, 2013.

[73] Kenji Hata, Ranjay Krishna, Li Fei-Fei, and Michael S Bernstein. A glimpse far into the future: Understanding long-term crowd worker quality. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 889–901, 2017.

[74] Qiuxiang He, Guoping Huang, Qu Cui, Li Li, and Lemao Liu. Fast and accurate neural machine translation with translation memory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3170–3180, 2021.

[75] Seth Hettich and Michael J Pazzani. Mining for proposal reviewers: lessons learned at the national science foundation. In *KDD*, pages 862–871, USA, 2006. ACM.

[76] Leading Global Influencer Marketing Agency Relatable in collaboration with 350 Brands and Agencies. The 2019 state of influencer marketing report. `https://www.cancerresearchuk.org/get-involved/citizen-science/the-projects#citizenscience4`, 2019. Accessed: 2022-05-20.

[77] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, 2017.

[78] Sarthak Jain and Byron C Wallace. Attention is not explanation. *CoRR*, abs/1902.10186, 2019.

[79] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *arXiv preprint arXiv:2202.03629*, 2022.

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

[81] Jian Jin, Qian Geng, Qian Zhao, and Lixue Zhang. Integrating the trend of research interest for reviewer assignment. In *WWW Companion*, pages 1233–1241, USA, 2017. IW3C2, ACM.

[82] Sally Jordan. Short-answer e-assessment questions: five years on. 2012.

[83] Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, 2017.

[84] Ece Kamar. Directions in hybrid intelligence: Complementing ai systems with human intelligence. In *IJCAI*, pages 4070–4073, 2016.

[85] Maryam Karimzadehgan and ChengXiang Zhai. Constrained multi-aspect expertise matching for committee review assignment. In *CIKM*, pages 1697–1700, USA, 2009. ACM.

[86] Maryam Karimzadehgan, ChengXiang Zhai, and Geneva Belford. Multi-aspect expertise matching for review assignment. In *CIKM*, pages 1113–1122, USA, 2008. ACM.

[87] Alexy Khrabrov and George Cybenko. Discovering influence in communication networks using dynamic graph analysis. In *2010 IEEE Second International Conference on Social Computing (SocialCom)*, pages 288–294, Minneapolis, Minnesota, USA, 2010. IEEE Computer Society.

[88] Hoyoung Kim, Seunghyuk Cho, Dongwoo Kim, and Jungseul Ok. Robust deep learning from crowds with belief propagation. In *International Conference on Artificial Intelligence and Statistics*, pages 2803–2822. PMLR, 2022.

[89] Joy Kim, Sarah Sterman, Allegra Argent Beal Cohen, and Michael S Bernstein. Mechanical novel: Crowdsourcing complex work through reflection and revision. In *Proceedings of the 2017 acm conference on computer supported cooperative work and social computing*, pages 233–245, 2017.

[90] Richard Klein, Angelo Kyrilov, and Mayya Tokman. Automated assessment of short free-text responses in computer science using latent semantic analysis. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, pages 158–162, 2011.

[91] Janin Koch, Andrés Lucero, Lena Hegemann, and Antti Oulasvirta. May ai? design ideation with cooperative contextual bandits. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[92] Janin Koch, Nicolas Taffin, Michel Beaudouin-Lafon, Markku Laine, Andrés Lucero, and Wendy E Mackay. Imagesense: an intelligent collaborative ideation tool to support diverse human-computer partnerships. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–27, 2020.

[93] Rafal Kocielnik, Saleema Amershi, and Paul N Bennett. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

[94] Ngai Meng Kou, U Leong Hou, Nikos Mamoulis, and Zhiguo Gong. Weighted coverage based reviewer assignment. In *SIGMOD*, pages 2031–2046, USA, 2015. ACM.

[95] Josua Krause, Adam Perer, and Kenney Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 5686–5697, 2016.

[96] Evgeny Krivosheev, Fabio Casati, Valentina Caforio, and Boualem Benatallah. Crowdsourcing paper screening in systematic literature reviews. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*, 2017.

[97] Evgeny Krivosheev, Fabio Casati, and Boualem Benatallah. Crowd-based multi-predicate screening of papers in literature reviews. In *WWW*, pages 55–64, USA, 2018. ACM.

[98] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 126–137, 2015.

[99] Igor Labutov and Christoph Studer. Jag: a crowdsourcing framework for joint assessment and peer grading. In *AAAI*, pages 1010–1016, USA, 2017. AAAI Press.

[100] Vivian Lai, Samuel Carton, Rajat Bhatnagar, Q Vera Liao, Yunfeng Zhang, and Chenhao Tan. Human-ai collaboration via conditional delegation: A case study of content moderation. In *CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2022.

[101] Himabindu Lakkaraju, Jure Leskovec, Jon Kleinberg, and Sendhil Mullainathan. A bayesian framework for modeling human evaluations. In *Proceedings of the 2015 SIAM International Conference on Data Mining (SDM)*, pages 181–189, Vancouver, BC, Canada, 2015. SIAM.

[102] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1675–1684. ACM, 2016.

[103] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*. OpenReview.net, 2019.

[104] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*. OpenReview.net, 2020.

[105] Janette Lehmann, Carlos Castillo, Mounia Lalmas, and Ethan Zuckerman. Finding news curators in twitter. In *Companion Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pages 863–870, Rio de Janeiro, Brazil, 2013. ACM.

[106] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *CoRR*, abs/1606.04155, 2016.

[107] Daifeng Li, Xin Shuai, Guozheng Sun, Jie Tang, Ying Ding, and Zhipeng Luo. Mining topic-level opinion influence in microblog. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1562–1566, Maui, Hawaii, USA, 2012. ACM.

[108] Guoliang Li. Human-in-the-loop data integration. *Proceedings of the VLDB Endowment*, 10(12):2006–2017, 2017.

[109] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220, 2016.

[110] Shao-Yuan Li, Yuan Jiang, Nitesh V Chawla, and Zhi-Hua Zhou. Multi-label learning from crowds. *IEEE Transactions on Knowledge and Data Engineering*, 31(7):1369–1382, 2018.

[111] Y. Li, J. Fan, Y. Wang, and K. Tan. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 30(10):1852–1872, 2018.

[112] Daniel J Liebling, Katherine Heller, Margaret Mitchell, Mark Díaz, Michal Lahav, Niloufar Salehi, Samantha Robertson, Samy Bengio, Timnit Gebru, and Wesley Deng. Three directions for the design of human-centered machine translation. 2021.

[113] Christopher H Lin, Mausam Mausam, and Daniel S Weld. Crowdsourcing control: Moving beyond multiple choice. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[114] Yuyu Lin, Jiahao Guo, Yang Chen, Cheng Yao, and Fangtian Ying. It is your turn: collaborative ideation with a co-creative robot through sketch. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.

[115] Haochen Liu, Joseph Thekinen, Sinem Mollaoglu, Da Tang, Ji Yang, Youlong Cheng, Hui Liu, and Jiliang Tang. Toward annotator group bias in crowdsourcing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1797–1806, 2022.

[116] Hui Liu, Qingyu Yin, and William Yang Wang. Towards explainable nlp: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5570–5581. Association for Computational Linguistics, 2019.

[117] Qiang Liu, Jian Peng, and Alexander T Ihler. Variational inference for crowdsourcing. *Advances in neural information processing systems*, 25, 2012.

[118] Jianhua Han Li'ang Yin, Weinan Zhang, and Yong Yu. Aggregating crowd wisdoms with label-aware autoencoders. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence. AAAI Press*, pages 1325–1331, 2017.

[119] Cheng Long, Raymond Chi-Wing Wong, Yu Peng, and Liangliang Ye. On good and fair paper-reviewer assignment. In *ICDM*, pages 1145–1150, USA, 2013. IEEE.

[120] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 745–754, Sydney, NSW, Australia, 2015. ACM.

[121] Zhanyu Ma and Arne Leijon. Bayesian estimation of beta mixture models with variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33: 2160–2173, 2011.

[122] Christian J Mahoney, Jianping Zhang, Nathaniel Huber-Fliflet, Peter Gronvall, and Haozhen Zhao. A framework for explainable text classification in legal document review. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1858–1867. IEEE, 2019.

[123] Yuning Mao, Tong Zhao, Andrey Kan, Chenwei Zhang, Xin Luna Dong, Christos Faloutsos, and Jiawei Han. Octet: Online catalog taxonomy enrichment with self-supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2247–2257, 2020.

[124] Adam Marcus, Eugene Wu, David Karger, Samuel Madden, and Robert Miller. Human-powered sorts and joins. *Proceedings of the VLDB Endowment*, 5(1):13–24, 2011.

[125] Tyler McDonnell, Matthew Lease, Mucahid Kutlu, and Tamer Elsayed. Why is that relevant? collecting annotator rationales for relevance judgments. In *HCOMP*, pages 139–148, USA, 2016. AAAI Press.

[126] Fei Mi and Dit-Yan Yeung. Probabilistic graphical models for boosting cardinal and ordinal peer grading in moocs. In *AAAI*, pages 454–460, USA, 2015. AAAI Press.

[127] Dennis P Michalopoulos, Jessica Jacob, and Alfredo Coviello. Ai-enabled litigation evaluation: Data-driven empowerment for legal decision makers. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pages 264–265, 2019.

[128] Joseph Victor Michalowicz, Jonathan M Nichols, and Frank Bucholtz. *Handbook of differential entropy*. Chapman and Hall/CRC, 2013.

[129] David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *KDD*, pages 500–509, USA, 2007. ACM.

[130] Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517, 2017.

[131] Ioannis Mitliagkas, Aditya Gopalan, Constantine Caramanis, and Sriram Vishwanath. User rankings from comparisons: Learning permutations in high dimensions. In *Allerton*, pages 1143–1150, New York City, USA, 2011. IEEE.

[132] Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216. Association for Computational Linguistics, 2020.

[133] Springer Nature. Focus on peer review. https://masterclasses.nature.com/focus-on-peer-review-online-course/16605550, 2020. Accessed: 2021-02-02.

[134] An Nguyen, Aditya Kharosekar, Matthew Lease, and Byron Wallace. An interpretable joint graphical model for fact-checking from crowds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[135] Andy Nguyen, Christopher Piech, Jonathan Huang, and Leonidas Guibas. Codewebs: scalable homework search for massive open online programming courses. In *Proceedings of the 23rd international conference on World wide web*, pages 491–502, 2014.

[136] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA:, 2015.

[137] Besmira Nushi, Ece Kamar, Eric Horvitz, and Donald Kossmann. On human intellect and machine failures: troubleshooting integrative machine learning systems. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, pages 1017–1025, San Francisco, California, USA, 2017. AAAI Press.

[138] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. I lead, you help but only with enough details: Understanding user experience of co-creation with artificial intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[139] Iyiola E Olatunji, Xin Li, and Wai Lam. Context-aware helpfulness prediction for online product reviews. In *AIRS*, pages 56–65, Switzerland, 2019. Springer.

[140] ACM Transactions on Social Computing. Review guidelines. https://dl.acm.org/journal/tsc/review-guidelines, 2019. Accessed: 2020-10-17.

[141] Luciana Padua, Hendrik Schulze, Krešimir Matković, and Claudio Delrieux. Interactive exploration of parameter space in data mining: Comprehending the predictive quality of large decision tree collections. *Computers & Graphics*, 41:99–113, 2014.

[142] Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 45–54, Hong Kong, China, 2011. ACM.

[143] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008.

[144] Aditya Parameswaran, Stephen Boyd, Hector Garcia-Molina, Ashish Gupta, Neoklis Polyzotis, and Jennifer Widom. Optimal crowd-powered rating and filtering algorithms. *Proceedings of the VLDB Endowment*, 7(9):685–696, 2014.

[145] Aditya Parameswaran, Ming Han Teh, Hector Garcia-Molina, and Jennifer Widom. Datasift: a crowd-powered search toolkit. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pages 885–888, 2014.

[146] Aditya Parameswaran, Akash Das Sarma, and Vipul Venkataraman. Optimizing open-ended crowdsourcing: The next frontier in crowdsourced data management. *Bulletin of the Technical Committee on Data Engineering*, 39(4):26, 2016.

[147] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933, 2016.

[148] Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. The language demographics of amazon mechanical turk. *Transactions of the Association for Computational Linguistics*, 2:79–92, 2014.

[149] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, USA, 2014. ACL.

[150] Diana Pérez-Marín and Ismael Pascual-Nieto. Willow: a system to automatically assess students' free-text answers by using a combination of shallow nlp techniques. *International Journal of Continuing Engineering Education and Life Long Learning*, 21(2-3): 155–169, 2011.

[151] Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, 2020.

[152] IEEE Potentials. Reviewer guidelines. `https://www.ieee.org/content/dam/ieee-org/ieee/web/org/members/students/reviewer_guidelines_final.pdf`, 2020. Accessed: 2020-02-26.

[153] Simon Price and Peter A Flach. Computational support for academic peer review: A perspective from artificial intelligence. *Communications of the ACM*, 60(3):70–79, 2017.

[154] Jiezhong Qiu, Jie Tang, Tracy Xiao Liu, Jie Gong, Chenhui Zhang, Qian Zhang, and Yufei Xue. Modeling and predicting learning behavior in moocs. In *Proceedings of the ninth ACM international conference on web search and data mining*, pages 93–102, 2016.

[155] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. Deepinf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2110–2119. ACM, 2018.

[156] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnn-logic: Learning logic rules for reasoning on knowledge graphs. In *International Conference on Learning Representations*, 2020.

[157] Jorge Ramírez, Marcos Baez, Fabio Casati, and Boualem Benatallah. Understanding the impact of text highlighting in crowdsourcing tasks. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, pages 144–152. AAAI Press, 2019.

[158] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

[159] Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. Finding expert users in community question answering. In *Proceedings of the 21st International Conference on World Wide Web (WWW)*, pages 791–798, Lyon, France, 2012. ACM.

[160] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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 (KDD)*, pages 1135–1144, San Francisco, CA, USA, 2016. ACM.

[161] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 61–70, Edmonton, Alberta, Canada, 2002. ACM.

[162] Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[163] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *CoRR*, abs/1703.03717, 2017.

[164] Nasim Sabetpour, Adithya Kulkarni, Sihong Xie, and Qi Li. Truth discovery in sequence labels from crowds. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 539–548. IEEE, 2021.

[165] Md Abdus Salam, Mary E Koone, Saravanan Thirumuruganathan, Gautam Das, and Senjuti Basu Roy. A human-in-the-loop attribute design framework for classification. In *The World Wide Web Conference*, pages 1612–1622, 2019.

[166] Belen Saldias-Fuentes and Pavlos Protopapas. A full probabilistic model for yes/no type crowdsourcing in multi-class classification. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 756–764. SIAM, 2019.

[167] Bruno Schneider, Dominik Jäckle, Florian Stoffel, Alexandra Diehl, Johannes Fuchs, and Daniel Keim. Integrating data and model space in ensemble learning by visual analytics. *IEEE Transactions on Big Data*, 7(3):483–496, 2018.

[168] Hendrik Schuff, Heike Adel, and Ngoc Thang Vu. F1 is not enough! models and evaluation towards user-centered explainable question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7076–7095, 2020.

[169] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[170] Nihar B. Shah and Zachary Lipton. SIGMOD 2020 tutorial on fairness and bias in peer review and other sociotechnical intelligent systems. In *SIGMOD Conference*, pages 2637–2640, USA, 2020. ACM.

[171] Nihar B Shah, Behzad Tabibian, Krikamol Muandet, Isabelle Guyon, and Ulrike Von Luxburg. Design and analysis of the nips 2016 review process. *The Journal of Machine Learning Research*, 19(1):1913–1946, 2018.

[172] Shahin Sharifi Noorian, Sihang Qiu, Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. What should you know? a human-in-the-loop approach to unknown unknowns characterization in image recognition. In *Proceedings of the ACM Web Conference 2022*, pages 882–892, 2022.

[173] Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *WWW*, pages 486–497. ACM / IW3C2, 2020.

[174] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 614–622, Las Vegas, Nevada, USA, 2008. ACM.

[175] Aashish Sheshadri and Matthew Lease. Square: A benchmark for research on computing crowd consensus. In *First AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, pages 156–164, Palm Springs, CA,USA, 2013. AAAI.

[176] Wanli Shi, Victor S Sheng, Xiang Li, and Bin Gu. Semi-supervised multi-label learning from crowds via deep sequential generative model. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1141–1149, 2020.

[177] Tarique Siddiqui, Paul Luh, Zesheng Wang, Karrie Karahalios, and Aditya Parameswaran. Shapesearch: A flexible and efficient system for shape-based exploration of trendlines. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 51–65, 2020.

[178] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[179] Edwin D Simpson, Matteo Venanzi, Steven Reece, Pushmeet Kohli, John Guiver, Stephen J Roberts, and Nicholas R Jennings. Language understanding in the wild: Combining crowdsourcing and machine learning. In *Proceedings of the 24th international conference on world wide web*, pages 992–1002, 2015.

[180] Richard Snodgrass. Single-versus double-blind reviewing: An analysis of the literature. *ACM Sigmod Record*, 35(3):8–21, 2006.

[181] Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263, 2008.

[182] Chonggang Song, Wynne Hsu, and Mong-Li Lee. Temporal influence blocking: Minimizing the effect of misinformation in social networks. In *33rd IEEE International Conference on Data Engineering (TKDE)*, pages 847–858, San Diego, CA, USA, 2017. IEEE Computer Society.

[183] SPRINGER. Guidelines for reviewers. `https://www.springer.com/authors/manuscript+guidelines?SGWID=0-40162-6-1261021-0`, 2020. Accessed: 2020-02-26.

[184] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[185] Wei Tang and Matthew Lease. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR 2011 workshop on crowdsourcing for information retrieval (CIR)*, pages 1–6, 2011.

[186] Wenbin Tang, Jie Tang, and Chenhao Tan. Expertise matching via constraint-based optimization. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 34–41, USA, 2010. IEEE, IEEE.

[187] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 75–86, Snowbird, Utah, USA, 2014. ACM.

[188] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1539–1554, Melbourne, Victoria, Australia, 2015. ACM.

[189] Jaime Teevan, Shamsi T Iqbal, and Curtis Von Veh. Supporting collaborative writing with microtasks. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 2657–2668, 2016.

[190] Yuandong Tian and Jun Zhu. Learning from crowds in the presence of schools of thought. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–234, Beijing, China, 2012. ACM.

[191] Toloka. Our global crowdforce. `https://toloka.ai/global-crowd`, 2022. Accessed: 2022-05-29.

[192] Andrew Tomkins, Min Zhang, and William D Heavlin. Reviewer bias in single-versus double-blind peer review. *Proceedings of the National Academy of Sciences*, 114(48): 12708–12713, 2017.

[193] Beth Trushkowsky, Tim Kraska, Michael J Franklin, and Purnamrita Sarkar. Crowdsourced enumeration queries. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 673–684. IEEE, 2013.

[194] Amazon Mechanical Turk. Welcome to the amazon mechanical turk requester user interface guide. `https://docs.aws.amazon.com/AWSMechTurk/latest/RequesterUI/amt-ui.pdf#Introduction`, 2014. Accessed: 2022-05-29.

[195] Dimitris G Tzikas, Aristidis C Likas, and Nikolaos P Galatsanos. The variational approximation for bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146, 2008.

[196] Christophe Van den Bulte and Yogesh V Joshi. New product diffusion with influentials and imitators. *Marketing Science*, 26(3):400–421, 2007.

[197] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[198] Jennifer Wortman Vaughan. Making better use of the crowd: How crowdsourcing can advance machine learning research. *Journal of Machine Learning Research*, 18:193–1, 2018.

[199] VLDB. Nlp challenge: Audio recordings transcription. https://crowdscience.ai/challenges/vldb21, 2021. Accessed: 2022-07-02.

[200] Luis Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

[201] Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, 2004.

[202] Luis Von Ahn, Ruoran Liu, and Manuel Blum. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64, 2006.

[203] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321 (5895):1465–1468, 2008.

[204] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.

[205] Jiannan Wang, Guoliang Li, Tim Kraska, Michael J Franklin, and Jianhua Feng. Leveraging transitive relations for crowdsourced joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 229–240, 2013.

[206] Jing Wang, Panagiotis G. Ipeirotis, and Foster Provost. Managing crowdsourcing workers. In *In The 2011 Winter Conference on Business Intelligence*, pages 10–12, 2011.

[207] Ke Wang and Xiaojun Wan. Sentiment analysis of peer review texts for scholarly papers. In *SIGIR*, pages 175–184, USA, 2018. ACM, ACM.

[208] Wanyuan Wang, Bo An, and Yichuan Jiang. Optimal spot-checking for improving evaluation accuracy of peer grading systems. In *AAAI*, pages 833–840, USA, 2018. AAAI Press.

[209] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198. Association for Computational Linguistics, 2017.

[210] Wentao Wang, Guowei Xu, Wenbiao Ding, Yan Huang, Guoliang Li, Jiliang Tang, and Zitao Liu. Representation learning from limited educational data with crowdsourced labels. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[211] Wei Wei, Gao Cong, Chunyan Miao, Feida Zhu, and Guohui Li. Learning to find topic experts in twitter via different relations. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 28(7):1764–1778, 2016.

[212] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2424–2432, Vancouver, British Columbia, Canada, 2010. Curran Associates, Inc.

[213] Steven Euijong Whang, Peter Lofgren, and Hector Garcia-Molina. Question selection for crowd entity resolution. *Proceedings of the VLDB Endowment*, 6(6):349–360, 2013.

[214] Jacob Whitehill, Ting fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2035–2043, Vancouver, British Columbia, Canada, 2009. Curran Associates, Inc.

[215] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, 2019.

[216] Barrett Wissman. Micro-influencers: The marketing force of the future? `https://www.forbes.com/sites/barrettwissman/2018/03/02/micro-influencers-the-marketing-force-of-the-future`, 2019. Accessed: 2019-10-11.

[217] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. JMLR.org, 2015.

[218] Yan Yan, Rómer Rosales, Glenn Fung, Mark Schmidt, Gerardo Hermosillo, Luca Bogoni, Linda Moy, and Jennifer Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 932–939. JMLR Workshop and Conference Proceedings, 2010.

[219] Jie Yang, Thomas Drake, Andreas Damianou, and Yoelle Maarek. Leveraging crowdsourcing data for deep active learning an application: Learning intents in alexa. In *Proceedings of the 2018 World Wide Web Conference (WWW)*, pages 23–32, Lyon, France, 2018. ACM.

[220] Jie Yang, Alisa Smirnova, Dingqi Yang, Gianluca Demartini, Yuan Lu, and Philippe Cudré-Mauroux. Scalpel-cd: leveraging crowdsourcing and deep probabilistic modeling for debugging noisy training data. In *Proceedings of the 2019 World Wide Web Conference (WWW)*, pages 2158–2168, San Francisco, CA, USA, 2019. ACM.

[221] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics:*

human language technologies, pages 1480–1489. The Association for Computational Linguistics, 2016.

[222] Mengfan Yao, Siqian Zhao, Shaghayegh Sahebi, and Reza Feyzi Behnagh. Stimuli-sensitive hawkes processes for personalized student procrastination modeling. In *Proceedings of the Web Conference 2021*, pages 1562–1573, 2021.

[223] Wenlin Yao, Cheng Zhang, Shiva Saravanan, Ruihong Huang, and Ali Mostafavi. Weakly-supervised fine-grained event recognition on social media texts for disaster management. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 532–539, 2020.

[224] Mo Yu, S. Chang, Y. Zhang, and T. Jaakkola. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*. Association for Computational Linguistics, 2019.

[225] Omar Zaidan and Chris Callison-Burch. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 1220–1229, 2011.

[226] Omar Zaidan, Jason Eisner, and Christine Piatko. Using “annotator rationales” to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267. The Association for Computational Linguistics, 2007.

[227] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41 (8):2008–2026, 2018.

[228] Jieyu Zhang, Xiangchen Song, Ying Zeng, Jiaze Chen, Jiaming Shen, Yuning Mao, and Lei Li. Taxonomy completion via triplet matching network. In *AAAI*, pages 4662–4670. AAAI Press, 2021.

[229] Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, Graham Neubig, and Satoshi Nakamura. Guiding neural machine translation with retrieved translation pieces. In *NAACL-HLT*, 2018.

[230] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836. IEEE Computer Society, 2018.

[231] Rui Zhang, Nathan J McNeese, Guo Freeman, and Geoff Musick. " an ideal human" expectations of ai teammates in human-ai teaming. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–25, 2021.

[232] Wenxuan Zhang, Wai Lam, Yang Deng, and Jing Ma. Review-guided helpful answer identification in e-commerce. In *WWW*, pages 2620–2626, USA, 2020. ACM / IW3C2.

[233] Ye Zhang, Iain Marshall, and Byron C Wallace. Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 795. NIH Public Access, 2016.

[234] Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. Efficient probabilistic logic reasoning with graph neural networks. In *International Conference on Learning Representations*, 2019.

[235] Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. Qasca: A quality-aware task assignment system for crowdsourcing applications. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1031–1046, 2015.

[236] Yudian Zheng, Guoliang Li, and Reynold Cheng. Docs: a domain-aware crowdsourcing system using knowledge bases. *Proceedings of the VLDB Endowment*, 10(4):361–372, 2016.

[237] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, 2017.

[238] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, 2017.

[239] Jianlong Zhou and Fang Chen. Towards trustworthy human-ai teaming under uncertainty. In *IJCAI 2019 Workshop on Explainable AI (XAI)*, 2019.

[240] James Zou, Kamalika Chaudhuri, and Adam Kalai. Crowdsourcing feature discovery via adaptively chosen comparisons. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

INES AROUS

PERSONAL INFORMATION

Born : November 2, 1992 Tunisia
Location : Fribourg, Switzerland
Email : ines@exascale.info
Homepage : https://inesarous.github.io/
Github : InesArous
Google Scholar ID : Ines Arous
ORCID : 0000-0001-7513-6197
H-index of 4, 60 total citations (Google Scholar)

EDUCATION

Ph.D. in Computer Science, advisor: Prof. Philippe Cudré-Mauroux — Mar 2017 – Oct 2022
eXascale Infolab, University of Fribourg — Fribourg, Switzerland

Exchange Program for 2nd year M.Sc. in Telecommunication Engineering — Sep 2015 – Jul 2016
Department of Information Engineering, University of Padova — Padova, Italy

Engineering Degree in Information and Communications Technology — Sep 2011 – Jul 2016
Higher School of Communication of Tunis — Aryanah, Tunisia

EMPLOYMENT HISTORY

Researcher/Ph.D. student — Mar 2017 – Oct 2022
eXascale Infolab, University of Fribourg — Fribourg, Switzerland
Ph.D. thesis titled "Human-AI Collaborative Approaches for Open-Ended Data Curation"

Applied Scientist Intern — Jun 2021 – Sep 2021
Amazon – Alexa Shopping (Team of Vanessa Murdock) — Seattle, USA
Internship project titled "Weak supervision for sequence labeling."

M.Sc. Student — Sep 2015 – Jul 2016
Department of Information Engineering, University of Padova — Padova, Italy
M.Sc. thesis titled "Statistical analysis of differential group delay in few-mode spun fibers"

RESEARCH PUBLICATIONS

- **Ines Arous**, Jie Yang, Mourad Khayati, and Philippe Cudré-Mauroux. "*Peer Grading the Peer Reviews: A Dual-Role Approach for Lightening the Scholarly Paper Review Process.*" In Proceedings of the Web Conference (**WWW 2021**).
  Link: https://dl.acm.org/doi/10.1145/3442381.3450088
- Mourad Khayati, **Ines Arous**, Zakhar Tymchenko and Philippe Cudré-Mauroux. "*ORBITS: Online Recovery of Missing Values in Multiple Time Series Streams.*" In Proceedings of the VLDB Endowment, Vol. 14, 2021 (**pVLDB 2021**).
  Link: http://www.vldb.org/pvldb/vol14/p294-khayati.pdf
- **Ines Arous**, Ljiljana Dolamic, Jie Yang, Akansha Bhardwaj, Giuseppe Cuccu, and Philippe Cudré-Mauroux. "*MARTA: A Human-AI Approach for Explainable Text Classification.*" In Proceedings of the AAAI Conference on Artificial Intelligence (**AAAI 2021**).
  Link: https://ojs.aaai.org/index.php/AAAI/article/view/16734
- **Ines Arous**, Jie Yang, Mourad Khayati, and Philippe Cudré-Mauroux. "*OpenCrowd: A Human-AI Collaborative Approach for Finding Social Influencers via Open-Ended Answers Aggregation.*" In Proceedings of the Web Conference (**WWW 2020**).
  Link: https://dl.acm.org/doi/10.1145/3366423.3380254
- **Ines Arous**, Mourad Khayati, Philippe Cudré-Mauroux, Ying Zhang, Martin Kersten, and Svetlin Stalinlov. "*RecovDB: Accurate and Efficient Missing Blocks Recovery for Large Time Series.*" In Proceedings of the 35th IEEE International Conference on Data Engineering (**ICDE DEMO 2019**).
  Link: https://ieeexplore.ieee.org/document/8731357

SUPERVISION OF JUNIOR RESEARCHERS

I have supervised the following M.Sc. Students:

- Julia Eigenmann for her M.Sc. thesis titled "Evaluating Text Classification Models on Multilingual Documents".
  - Defense date: Sep 2021
  - The project was about comparing text classifiers to identify potential clients for a Swiss company.
- Louis Müller for his M.Sc. thesis titled "Multiclass classification of open-ended answers".
  - Defense date: Mar 2021
  - The project was about developing a method to perform multi-class classification of open-ended answers using Gibbs Sampling. The performance of the proposed approach was evaluated on three real-world datasets.
- Zeno Bardelli for his M.Sc. thesis titled "SwissFinder: Identifying Swiss Websites from Unstructured Content".
  - Defense date: Sep 2020
  - The project was a benchmark of machine learning classifiers to identify Swiss companies' websites given their properties and the results of a search engine. The results were published as a poster:
    Zeno Bardelli, Ines Arous, Philippe Cudré-Mauroux, and Ljiljana Dolamic. "*SwissFinder: Identifying Swiss Websites from Unstructured Content.*" In 2020 IEEE International Conference on Big Data (**IEEE Big Data 2020**).

TEACHING ACTIVITIES

- Teaching assistant for the B.Sc. course "operating systems" (2017-2021) and for the M.Sc. course "big data infrastructures" (2017)
  - Teach labs and explain hand out assignments
  - Tutor groups of students working on the course's project
  - Assist professor with exams

MEMBERSHIPS AND SCIENTIFIC REVIEWING ACTIVITIES

- PC member for the AAAI Conference on artificial intelligence **(AAAI 2021)**
- Session chair for the data mining session at the international conference on information and knowledge management **(CIKM 2020)**.
- External Reviewer for: CIKM 2017, EDBT 2017, ICDE 2017, WWW 2017, ICDE 2018, ICDM 2018, KDD 2018, CIKM 2019, WWW 2019, CIKM 2020, ICDE 2020, IJCAI 2020, WSDM 2021, CIKM 2021, WWW 2021, WWW 2022.
- Member of the Switzerland Section of IEEE
- Organizing Committee member of:
  - ACM-Tunisian Collegiate Programming Contest 2013
  - Arab Mobile App Chalenge 2013
  - TEDx SUPCOM 2014

PERSONAL SKILLS

Digital Competences

- Programming languages: Python, Matlab, C++
- Python frameworks: Numpy, Pandas, Tensorflow, Pytorch, Keras
- Crowdsourcing platforms: Mechanical Turk and Appen
- Statistical inference: Variational Inference, Expectation Maximization, Gibbs Sampling
- Data Mining: Matrix factorization for recovery of time series

Languages

- English (Fluent), French (Native), Arabic (Native)

RESEARCH PROJECTS

- Technology Market Monitoring with **armasuisse** (Mar 2020 - Aug 2022): build a system to identify technology-related Wikipedia pages.
- Cancer diagnosis with **Hospital of Fribourg** (Mar 2019 - Jun 2020): develop a framework to leverage non-medical participants in cancer diagnosis.
- FashionBrain (**EU** project) (Mar 2017 - Feb 2020): design a human-AI framework to detect fashion influencers.