Harnessing Incomplete, Noisy, and Multi-level Labels for Classification and Annotation Tasks

Dongyu Zhang

A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Data Science

April 2024

Committee Members:
Dr. Elke A. Rundensteiner, Professor, WPI, Advisor.
Dr. Xiangnan Kong, Associate Professor, WPI.
Dr. Nima Kordzadeh, Assistant Professor, WPI.
Dr. Liang Wang, Principal Research Scientist, Visa Research.

Abstract

Deep learning models, known for deciphering complex patterns, consistently excel across various machine learning tasks. Training these models typically demands a large amount of accurately labeled data. However, the aspiration for perfect datasets often meets the challenges of real-world data collection. Acquiring precise labels is a daunting task due to prohibitive costs, limited labeling resources, and the need for domain expertise. Some datasets are characterized by *incomplete labels*, where a sizable portion remains unlabeled, creating informational voids. Others are marred by *noisy*, where the provided annotations deviate from the ground truth, posing risks of misdirection during model training. In more challenging scenarios, datasets might suffer from both incomplete and noisy labels, further complicating the training process.

The concept of multi-level labeled data reflects the multifaceted nature of data and the different depths at which insights can be extracted. For instance, in detecting foodborne illness from social media posts, the task at the post level is to determine if a post indicates a potential foodborne illness event. At the word level, the aim is to identify specific entities such as symptoms or food items related to the incident. Depending on the objective, labels at both two levels might be necessary, the difficulty and cost of obtaining these labels can differ greatly. Multi-level labeled datasets often suffer from incomplete or noisy labels. However, the interconnections between the two levels offer a unique advantage. The overall context of a post provides valuable insights to better identify specific word-level slots, while recognizing specific entities offers clarity on the broader message of the post. The label quality, completeness issue, and the connection across levels emphasize the importance of adaptive strategies to achieve improved outcomes.

Large Language Models (LLMs) have demonstrated remarkable performance across a broad spectrum of tasks. They excel in in-context learning (ICL), where they make predic-

tions based on a limited number of examples without needing updates to their parameters. The Chain-of-Thought (CoT) method, which prompts LLMs to articulate their reasoning steps in addition to providing the final answer, further enhances their capability to tackle complex problems. LLMs can serve as annotators to mitigate the challenge of label scarcity. However, the labels they generate may be noisy due to the models' tendency to hallucinate. The effectiveness and potential issues of using LLMs for annotation warrant further exploration.

My dissertation research centers on three directions to address the challenges posed by incomplete, noisy, and multi-level labeled datasets. These directions are: 1) learning from two-level labeled datasets with one level having complete labels and the other having incomplete labels, 2) learning from datasets with noisy labels, and 3) in context learning from two-level labeled datasets with incomplete labels.

Direction 1: Learning from two-level labeled datasets with one level having complete labels and the other having incomplete labels. In practical scenarios, obtaining labels at a more fine-grained level is often more resource-intensive and challenging than at broader levels. For instance, Human Attention Maps (HAMs) in text classification comprise detailed word-level labels from human annotators. These labels serve as explanations, derived from the influence each word has on human predictions. The process of collecting HAMs is considerably more demanding than obtaining mere classification labels about the sentence as a whole. This is because it necessitates annotators to invest significant effort and time in evaluating every word within an extensive dataset.

In this context, we introduce a novel problem, which we call explainable text classification with limited human attention supervision. The goal is to craft a classifier that offers human-like explanations when comprehensive text classification labels are at hand, but only a sparse set of HAMs are available. To address this challenge, we present HELAS, a pioneering solution that seamlessly integrates joint learning of tasks at both levels. This enhances

both the text classification and human-like explanation tasks, even when faced with limited supervision labels for the latter.

Direction 2: Learning from datasets with noisy labels. Obtaining meticulously labeled data is both expensive and time-consuming, leading many researchers and practitioners to turn to alternate non-expert labeling sources, such as crowd-sourcing or automated annotations using pre-trained models. While these methods enhance efficiency and curtail expenses, they often compromise the integrity of the labels, resulting in noisy labels. The challenge of learning with such labels has been a focal point of numerous studies. Yet existing techniques presuppose knowledge about the proportion of noisy labels or their specific characteristics. In practical settings, this information is often unavailable, hindering the effective application of these techniques. Further complications arise when noisy labels are mishandled or mis-corrected, leading to compounded errors that can adversely impact representation learning and induce overfitting.

Addressing these intricacies, we introduce CoLafier. Unlike existing models, this solution harnesses the local intrinsic dimensionality (LID) score, derived from the enhanced representation of training samples' features and label, to discern between accurate and noisy labels. CoLafier utilizes the LID score for adaptive instance weighting and for correcting noisy labels during the training phase. Enhancing its resilience, CoLafier integrates two augmented views for each sample, using their LID scores to counteract error propagation.

Direction 3: In context learning from two-level labeled datasets with incomplete labels. Consider a dataset of social media posts for foodborne illness detection, which operates on two levels: determining if the post indicates a foodborne illness incident at the post level, and identifying relevant entities (e.g., food, symptom) at the word level. The two levels are interconnected: as the overall post context provides valuable insights to better identify relevant entities, while recognizing specific entities can offer clarity on the broader message of the post. However, the high costs of labeling often leave datasets largely

unlabeled. As a solution to label scarcity, LLMs offer a promising alternative. Using labeled examples as demonstrations, LLMs can efficiently annotate unlabeled data, but it may suffer from model hallucinations and generate low quality labels.

In this work, we introduce ICL2FID, an ICL-based labeling framework designed to annotate datasets of social media posts regarding two-level foodborne illness detection. This approach utilizes CoT method to guide the LLM to leverage insights from one level when it makes prediction at another. A critical verification step in between word and post level labeling steps eliminates incorrect entities extracted earlier, preventing them from influencing subsequent labeling outcomes. Employing varied example retrieval strategies at each stage, ICL2FID minimizes biases arising from repetitive exposure to identical posts and labels, thereby effectively mitigating the risk of model hallucination. This method offers a novel approach for labeling multi-level datasets in scenarios with limited resources .

Three tasks presented above mark a substantial stride in applying deep learning techniques to data with incomplete, noisy, and multi-level labels. Extensive evaluations on real-world datasets and comparison with state-of-the-art methods confirm their efficacy. This dissertation offers robust frameworks and sets a foundation for future research on real-world challenges associated with label completeness, quality, and structure in the data.

Acknowledgements

I extend my deepest gratitude to all who have supported me during my Ph.D. journey. Their kindness, support, and instruction have been pillars of my progress.

Foremost, my heartfelt thanks to my advisor, Professor Elke Rundensteiner, for her invaluable feedback, guidance, patience, and kindness. Her mentorship has been a beacon for my academic endeavors. Equally, I am thankful to my committee members, Professor Xiangnan Kong, Professor Nima Kordzadeh, and Dr. Liang Wang, for their contributions to my previous research and this dissertation. A special acknowledgment to Prof. Hao Feng for his assistance with the FACT project. The camaraderie and collaboration offered by my peers in the DAISY Lab, the FACT team, and the MQP students have been instrumental in my development. I owe a debt of gratitude to my close collaborators Ruofan Hu, Dandan Tao, Jidapa Thadajarassiri, Cansu Sen, and Thomas Hartvigsen for their friendship and guidance, which have greatly enriched my experience. I am privileged to work alongside such distinguished professors and colleagues.

Further appreciation goes to my parents and family, whose love, encouragement, and unwavering support have sustained me. Lastly, I thank WPI Data Science program and the FACT Project (Agriculture and Food Research Initiative award No. 2020-67021-32459) for their support of my research.

Contents

Chapter 1

Introduction

1.1 Motivation

Deep learning models, particularly neural networks, are renowned for their ability to decipher intricate patterns in vast amounts of data. Over the years, they have consistently demonstrated superior performance across a wide range of machine learning tasks from computer vision to natural language processing [1]. The traditional approach to training these networks requires a large amount of annotated data. Traditionally, it was explicitly or implicitly assumed that all annotations are correct [1]. However, this strict assumption of annotations being accurate in most cases clashes with the practical challenges of real-world data collection. Obtaining a large number of accurate labels is a daunting endeavor and it is in fact challenging to assert if and when this holds true. This challenge is largely attributed to high costs, limited labeling resources, and the necessity for domain-specific expertise [2]. The ImageNet dataset, extensively employed in the Computer Vision domain, traditionally treated its test set labels as "correct." However, a previous study uncovered a substantial number of labeling inaccuracies [3], highlighting the intricacies and potential for error in manual annotation processes within highly utilized datasets in artificial intelligence

research.

We note that different types of imperfections may arise in labels assigned to data sets. Some datasets are characterized by *incomplete labels*, where a significant portion pf the data items in the data set remains unlabeled, creating informational voids. Others suffer from *noisy labels*, where the provided annotations deviate from the ground truth. Such deviations pose risks of misdirection during model training, potentially leading to degraded performance [1]. The label noise ratio in real-world datasets is reported to range from 8.0% to 38.5% [1]. As for incomplete labels, they in fact may be more prevalent than one may expect, because even many fully-labeled real-world datasets are in fact just subset samples of a raw and otherwise much much larger unlabeled data set. Put differently, the vast majority of real-world data in most domains of interest remain unlabeled. In more challenging scenarios, datasets might even suffer from the existence of both incomplete and noisy labels, further complicating the training process.

Figure 1.1 illustrates diverse labeling scenarios in the context of annotating data for foodborne illness case detection [4, 5], where the objective is to ascertain whether the given text mentions a foodborne illness case. Models trained on such data play a pivotal role in the early detection of foodborne illnesses. This early detection is crucial for mitigating risks, controlling outbreaks, and safeguarding public health. The top-right quadrant represents the ideal situation where data are fully labeled and all labels are accurate (complete & clean). The top-left quadrant illustrates a scenario where data are fully labeled, yet some labels are inaccurate (complete & noisy). The bottom-right quadrant shows a case with partial labeling, where it holds that the specified labels are all accurate (incomplete & clean). The bottom-left quadrant represents the most challenging situation, where only some of the data items in the data set are labeled and some of these labels are in fact inaccurate (incomplete & noisy).

The concept of multi-level labeled data arises from the multifaceted nature of data and

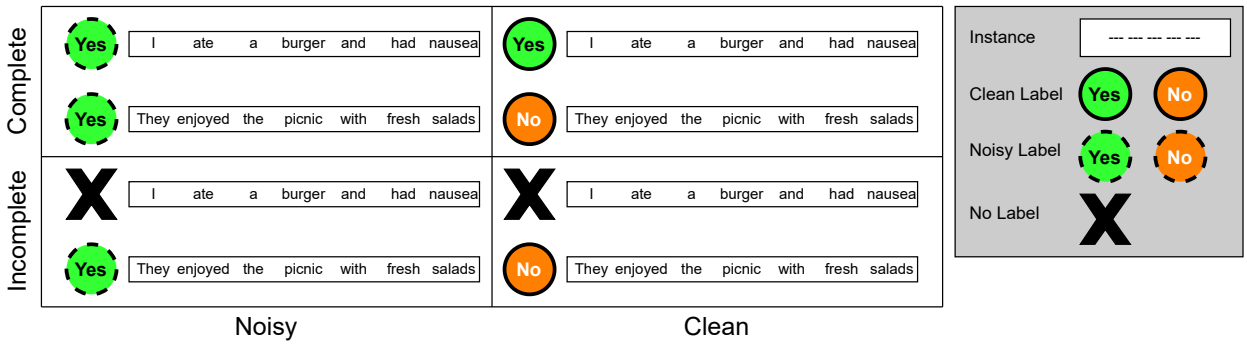


Figure 1.1: Label completeness and accuracy scenario for foodborne illness case detection task.

the different depths at which insights can be extracted. Figure 1.2 illustrates the foodborne illness detection task from two-level labeled posts. Here, a post could be classified for its overall relevance to foodborne illness (a broader post-level label) and for the specific entities it mentions (fine-grained word-level labels) such as symptoms or food items. A post might read "I ate a burger and had nausea." which could be broadly labeled as a potential foodborne illness case, with fine-grained word-level labels identifying "burger" as a food and "nausea" as a symptom. Both of the later finer grained labels are related to the foodborne illness case. Depending on the objective, both broad and fine-grained labels might be necessary, leading to datasets annotated at multiple levels.

However, the challenges in acquiring such different types of labels can vary significantly in terms of complexity and costs. For example, post-level labels are generally more straightforward to obtain because they require only a broad assessment of the post's content, which is less time-consuming and requires less specialized knowledge than identifying specific entities within the text. In contrast, word-level annotations demand a detailed examination of the text to accurately identify and classify each word in the text. As such, it is common to find datasets where overall post-level labels are abundant and mostly accurate, yet word-level slot annotations are sparse or fraught with errors [6]. Such disparities often stem from the need for specific domain knowledge for accurate word-level labeling, compounded by budgetary constraints that limit the extent of detailed annotation work that can be undertaken. Conse-

quently, both levels might have incomplete or noisy labels, but one level might be relatively more reliable than the other level. This variation in label quality and completeness across levels emphasizes the need for adaptive strategies in model design and training to achieve the best outcomes.

The emergence of Large Language Models (LLMs), such as GPT-3 in 2020, [7], ChatGPT in 2022, [8], Llama 2 in 2023[9], has captured widespread attention in recent years These models, provided with just a natural language instruction or a few demonstration examples, have demonstrated exceptional proficiency across a broad spectrum of tasks. This proficiency stems from their ability to perform tasks without the need for additional training on task-specific data, a stark contrast to traditional supervised learning models which require extensive training on labeled datasets for each new task. This capability of LLMs, often referred to as "in-context learning" [10, 11, 12], allows LLMs to generate predictions or complete tasks based on the context provided within the prompt, without undergoing further parameter updates or learning processes. In-context learning is distinct because it leverages the extensive knowledge and patterns LLMs have acquired during their initial, comprehensive training phase on vast datasets. Therefore, unlike supervised learning that necessitates task-specific model training, in-context learning exploits the pre-trained model's existing capabilities to understand and respond to new instructions, making it a highly versatile and efficient approach to tackling a wide array of tasks with minimal additional input.

Remarkably, ICL operates without the need for additional training or updates to the model's parameters, positioning it as an effective strategy in situations where computation resources are limited. This means, that leveraging a few labeled examples, LLMs could be instructed to annotate unlabeled data. This is exciting as it would help to bridge the huge gap identified above between label scarcity and the need for labeled data. However, it's important to note that human annotators, too, are prone to errors and inconsistencies in labeling, as evidenced by various studies [13, 14]. As LLM technology continues to evolve and become more sophisticated, it is expected that their accuracy and reliability will improve,

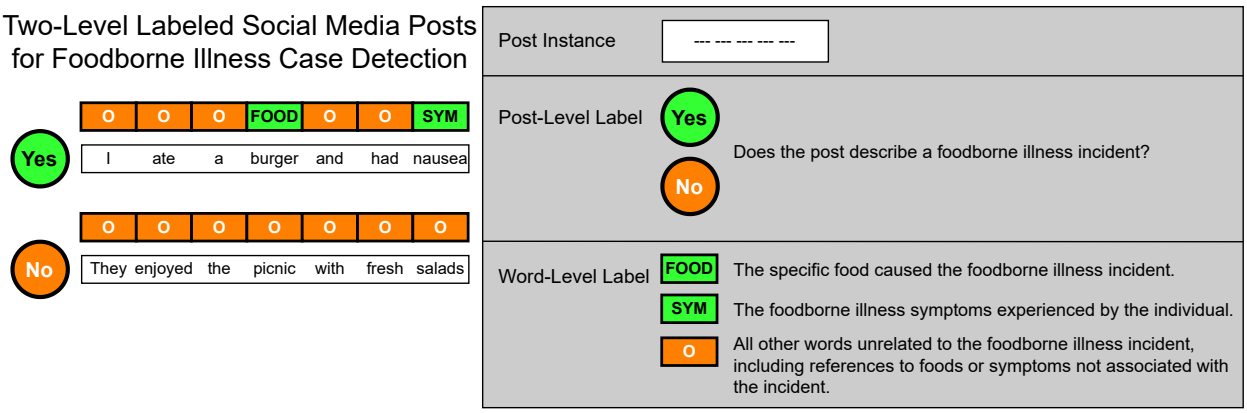


Figure 1.2: Two-level labeled social media posts for foodborne illness case detection

mitigating the issue of hallucinations. Moreover, as LLMs become more mainstream and their deployment costs decrease, their application in extensive and labor-intensive tasks such as label generation is likely to expand. This trend is promising for overcoming the challenges associated with acquiring large volumes of high-quality labeled data, especially in domains where resources are scarce. Thus, despite the current limitations, the use of LLMs in label generation holds significant promise for the future, potentially revolutionizing the way we approach and manage data annotation tasks as LLM technology matures.

1.1.1 Motivating Example: FACT Project

A practical example of real-world data exhibiting incomplete, noisy, and multi-level labels is the data collected in the FACT project[1]. The initiative within the FACT project aims to harness big data analytics technologies to enhance the safety of fresh produce, exploring social media analysis for early warnings about food safety concerns.

The objective is to train a deep learning model to detect mentions of foodborne illness incidents in social media posts, with tweets being collected to build a dataset. A successful method would not only identify if a tweet indicates a possible foodborne illness incident but also autonomously extract crucial entities from the tweet for aggregation into actionable

[1]Project link: https://www.nal.usda.gov/research-tools/food-safety-research-projects/fact-innovative-big-data-analytics-technology-microbiological-risk-mitigation-assuring-fresh-produce

trends. To accelerate this inspection process, it's divided into three automatable tasks: (1) determining if the tweet hints at a foodborne illness incident; (2) locating and extracting mentions of food, symptoms, locations, and foodborne illness-related keywords from the tweet; and (3) recognizing mentions of slots, i.e., the attributes of an incident like *What*, *Where*, and their values pertaining to any foodborne illness incident described in the tweet [4]. These tasks are classified as Text Relevance Classification (TRC), Entity Mention Detection (EMD), and Slot Filling (SF), respectively.

During the data collection phase, due to limited resources, most tweets remain unlabeled, with only a small portion labeled by crowdsourced workers and an even smaller fraction annotated by experts, resulting in an incompletely labeled dataset. During the label collection phase, annotations from 5 workers were collected for each tweet. Low-quality labels were discarded, retaining only those tweets with at least 5 high-quality labels. The quality of crowdsourced labels was evaluated using various aggregation methods, comparing them with expert labels, which served as the ground truth. The evaluation disclosed a quality gap between crowdsourced and expert labels, with the gap widening for fine-grained tasks like Entity Mention Detection and Slot Filling. These tasks, necessitating a word-by-word analysis, were found to be more demanding and challenging compared to broader-level tasks [4]. The aforementioned challenges hint at a broader implication: classifiers trained on datasets marred by incomplete and noisy labels are likely to underperform, showcasing diminished generalization capabilities. This underscores the necessity for developing more robust models that can adeptly navigate the complexities of real-world data, particularly those requiring nuanced understanding at both the broad and fine-grained levels.

1.2 Overall Objectives

This dissertation focuses on three directions of investigation to address the challenges posed by incomplete, noisy, and multi-level labeled datasets. These directions are 1) learning

from two-level labeled datasets with one level having complete labels and the other having incomplete labels, 2) learning from datasets with noisy labels, and 3) in-context learning from two-level labeled datasets with incomplete labels.

Referring back to the four quadrants illustrated in Figure 1.1, directions 1 and 3 specifically address the scenarios depicted by the bottom-right quadrant (incomplete & clean). Direction 1 targets the case where one level of labeling is complete and the other is incomplete, while direction 3 delves into situations where both labeling levels are incomplete, showcasing a nuanced approach to handling incompleteness across dataset levels. The direction 2 directly tackles the challenges represented by the top-left quadrant (complete & noisy), focusing on improving label accuracy in fully labeled datasets. However, it's noteworthy that the bottom-left quadrant (incomplete & noisy), representing datasets with both incomplete and noisy labels, is not directly tackled within this dissertation.

This focus reveals a gap in addressing datasets that simultaneously suffer from incompleteness and noise, particularly in scenarios where this affects both levels of labeling. The dissertation's scope, thus, highlights areas ripe for future investigation, aiming to bridge these gaps and enhance methodologies for dealing with the full spectrum of labeling challenges in datasets.

1.2.1 Learning from Two-Level Labeled Datasets with One Level Having Complete and the Other Incomplete Labels

For multi-level labeled datasets, acquiring labels at a more fine-grained level is often more resource-intensive and challenging than at broader levels. This leads to the exploration of learning from two-level labeled datasets, with one level having complete labels and the other having incomplete labels. The core idea is to deal with datasets where fine-grained labels, like Human Attention Maps (HAMs) in text classification, coexist with broader sentence-level labels. HAMs provide word-level labels that explain the influence

each word has on human predictions. However, collecting such detailed labels is much more demanding compared to obtaining broader classification labels for the entire sentence, as it requires a thorough examination of every word within a large dataset. The need for such detailed analysis often collides with the practical challenges of label collection. This may result in a dataset where each instance is associated with a classification label, and a small proportion of these training instances also have fine-grained word-level labels. In this case, a classifier trained on such a dataset may overemphasize the fully-labeled classification task and perform worse on the human-like attention generation task. We term this problem *explainable text classification with limited human attention supervision*, and it is the particular focus of this dissertation's first research direction.

1.2.1.1 State-of-the-Art: Learning from Two-Level Labeled Datasets with One Level Having Complete Labels and the Other Having Incomplete Labels

In the domain of learning with incomplete labels, two predominant methods emerge: semi-supervised learning [15] and active learning [16]. Active learning presupposes the availability of a human expert who can be queried to acquire ground-truth labels for selected unlabeled instances, unlike semi-supervised learning, which operates without such an assumption. In this dissertation work, the focus is primarily cast on methods based on semi-supervised learning.

In the realm of semi-supervised learning within multi-level labeled scenarios, a range of methods, including pseudo-labeling and self-training, have been developed to leverage unlabeled instances [17, 18]. Typically, these methods address scenarios with extensive unlabeled data, indicating incomplete labeling across all levels. Our case, however, presents the distinctive challenge of having one fully-labeled level alongside another level with limited labels, creating an imbalance in supervised information that has yet to be addressed by the literature.

In text classification, Barrett et al. [19] and Zhang et al. [20, 21] have utilized human attention maps (HAMs) for attention generation, relying on extensive word-level annotations or eye-tracking data. These methods, while insightful, require substantial annotation efforts and are not optimized for scenarios with limited HAMs. An efficient method capable of training with sparse HAMs remains an underexplored area warranting further investigation.

Recent Advances and Additional References: To ensure the citations are current and relevant, recent studies should be reviewed to update the references in this section to include works up to 2024 that discuss advancements in semi-supervised learning, active learning, and their applications in multi-level labeled datasets. This will not only reinforce the dissertation's credibility but also provide the most current insights into the challenges and solutions in this area of research.

1.2.1.2 Challenges: Learning from Two-Level Labeled Datasets with One Level Having Complete Labels and the Other Having Incomplete Labels

Tackling two-level labeled datasets, where one level boasts complete labels while the other grapples with incomplete labels, poses an inherently intricate challenge. At the core of this challenge lie two distinct tasks: the broader level task and the fine-grained level task. Each task presents a disparate amount of labeled data — while all data entries are accompanied by broader level labels, only a subset is furnished with fine-grained level labels. This disparity in label availability necessitates a nuanced approach to learning from such datasets.

A proficient solution must adeptly balance the feedback derived from each task. It is crucial to prevent an overemphasis on the fully-labeled task, which could overshadow the learning from the incompletely labeled task, potentially leading to a bias in the learning process. Furthermore, the solution should judiciously leverage the available fine-grained labels to enhance the learning process, while also effectively utilizing the broader labels to

compensate for the lack of fine-grained labels. The inherent hierarchical relationship between the broader and fine-grained tasks adds another layer of complexity. It requires a method that can seamlessly integrate information across these levels, promoting a harmonious learning process that optimally benefits from the unique information provided by each level of labeling.

1.2.2 Learning from Datasets with Noisy Labels

Obtaining high-quality labels for large volumes of data is both expensive and resource-intensive, often necessitating domain-specific knowledge [22]. Given these challenges, many researchers or practitioners have sought alternative labeling sources such as crowdsourcing [4] or automatic label annotation using pre-trained models [23]. While these methods enhance efficiency and curtail costs, they frequently compromise label quality [24]. Labels procured through these approaches are referred to as *noisy labels*, as they may deviate from the true ground-truth labels. Recent literature [1, 25] underscores that Deep Neural Networks (DNNs), notwithstanding their resilience across myriad AI applications, remain vulnerable to label noise. Such noisy labels can impede network performance, emphasizing the necessity to attain commendable generalization capacity [1].

1.2.2.1 State-of-the-Art: Learning from Datasets with Noisy Labels

The prevalence of noisy labels in datasets has spurred a substantial body of research aimed at enhancing the robustness of the learning method. Previous works addresss this by developing noise-adaptive architectures [26, 27], introducing regularization techniques [28, 29], and formulating improved loss functions [30, 31]. Nevertheless, these methods often struggle with high noise ratios and intricate noise patterns [1]. Recent studies have primarily spotlighted two techniques for training DNNs with noisy labels: sample selection and label correction. Sample selection methods endeavor to identify potentially mislabeled samples, minimizing their impact during training. Such samples might be discarded [32, 33], assigned

reduced weights in the loss function [34], or considered as unlabeled, with semi-supervised learning techniques employed [35, 36]. Conversely, label correction strategies seek to improve the training set by identifying and amending mislabeled instances. Both soft and hard correction approaches have been proposed [34, 23, 37]. While these methods have markedly enhanced noise robustness, the introduction of hyperparameters in these methods can make DNNs more vulnerable to variances in data and noise types [1].

1.2.2.2 Challenges: Learning from Datasets with Noisy Labels

Several challenges emerge when learning from noisy labels. including confirmation bias, which arises when the model makes incorrect selection or correction decisions, thereby becoming biased and progressively adapting to the noise. Additionally, some methods assume knowledge of the noise label ratio and pattern, utilizing this information to inform their hyper-parameter settings [32, 36, 35]. However, in real-world scenarios, this information is typically absent, making these methods less viable in practice. It is challenging to develop a universal method that can collect sufficient clean labels to train a strong model.

1.2.3 In-Context Learning from Two-Level Labeled Datasets with Incomplete Labels

Acquiring labels for large datasets is a difficult and expensive process. This difficulty increases when labeling tasks need annotation on multiple levels. Take, for instance, a dataset of social media posts aimed at detecting foodborne illness incidents, operating on two levels. On the post level, the task entails predicting whether a post indicates a foodborne illness incident. In contrast, on the word level, the objective is to identify mentions of slots, aka entities (e.g., What, Where) related to a mentioned foodborne illness incident. The overall context of a post can provide valuable insights to better identify specific relevant entities, while recognizing specific entities can offer clarity on the broader message of the post.

Given the high costs associated with label collection, datasets often remain largely unlabeled. Faced with label scarcity, particularly in settings with limited computation resources, Large Language Models (LLMs) present a viable solution. By using labeled instances in the dataset as demonstration examples, LLMs can annotate unlabeled data, narrowing the gap between the lack of labels and the need for comprehensive labeling. Yet, LLMs are prone to hallucination, resulting in potentially noisy labels. Additionally, the strategy to exploit the relationship between the two levels of labels remains unexplored.

1.2.3.1 State-of-the-Art: In-context Learning from Two-Level Labeled Datasets with Incomplete Labels

Numerous methods have been proposed for learning from two-level labeled datasets [38]. These methods achieve remarkable performance by leveraging the relationship between the two levels [39, 40, 41]. However, most of these works assume that both two level are fully labeled and these methods usually require training and model parameters updates, which do not address the issues of label completeness in resource-limited scenario. To our knowledge, no existing studies have tackled the specific problem setting we propose.

Regarding in-context learning, some research has investigated using LLMs to label various NLP tasks [42, 43] – though not the particular task that we are focussing on. With carefully constructed task descriptions and labeled examples, LLMs can label vast datasets without needing training or model updates. However, of course, this does require sustained effort on prompt engineering [44]. Further, the Chain-of-Thought (CoT) approach, which encourages LLMs to outline their thought process before the final answer, significantly boosts their ability to solve complex tasks [45]. Nevertheless, the specific application of LLMs for multi-level labeling within real-world datasets, especially leveraging the interconnections of labels across levels, remains largely uncharted territory.

1.2.3.2 Challenges: In-context Learning from Two-Level Labeled Datasets with Incomplete Labels

The task of learning from two-level labeled datasets is inherently challenging, requiring methods that effectively utilize the relationship between the two tasks [38]. This complexity is magnified in in-context learning scenarios using LLMs, which can suffer from hallucination issues, potentially leading to noisy label generation. Utilizing incorrect labels from one level to inform the prediction on the other level can result in compounded errors. Moreover, learning from datasets with few labeled examples is already challenging due to the risk of overfitting to the small labeled set. This challenge is exacerbated in ICL scenarios, where LLMs may become biased by the demonstration examples provided. Thus, the development of a robust strategy for selecting demonstration examples is crucial for ensuring reliable learning outcomes.

1.3 Dissertation Tasks

In this dissertation, I tackle the following three tasks described below. A detailed description of each task and its solution is presented in the subsequent sections. We note that tasks 1, 2, and 3 align with the previously mentioned research directions 1, 2, and 3, respectively.

Task 1: Explainable Text Classification with Limited Human Attention Supervision

In this task, we delve into a specific scenario entailing learning from two-level labeled datasets, wherein one task is endowed with complete labels while the other possesses incomplete labels. We are presented with a set of training documents, each tagged with a corresponding classification label. A small portion of these training documents also bear

fine-grained word-level Human Attention Map (HAM) labels, denoting the words a human annotator deemed most pertinent while assigning the class label. Our objective is to engineer a model capable of adeptly tackling the text classification task, whilst concurrently generating human-like attention weights that mirror those a human would produce for the given document.

For this task, we propose a deep learning framework named HELAS: Human-like Explanation with Limited Attention Supervision, designed to adaptively learn attention weights focusing on words in a manner analogous to human attention with very limited supervision. HELAS comprises two key components: the first is an innovative attention method called the human-like attention learner, which successfully learns attention weights that mimic human attention, adapting to different contexts; the second is a custom contextualized representation that considers the impact of all words when making its final prediction. HELAS effectively unifies joint learning, improving both text classification and human-like explanation tasks, even with only minimal supervision labels for the latter. Our evaluation studies on three real-world datasets demonstrate that HELAS outperforms state-of-the-art alternatives in learning an accurate text classifier and generating human-like attention, even when as little as 2% of the data contain HAM labels. **This work is published at IEEE Big Data 2021 [46].**

Task 2: Classification with Noisy Labels

This task pivots on devising a method to navigate the classification task amidst the quagmire of noisy labeled training data. We are given a set of training data, each item accompanied by a noisy classification label, with no insight into the accuracy of each label. Our aim is to cultivate a classification model that stands resilient to label noise and adeptly executes the classification task.

For this task, we introduce CoLafier, a cutting-edge framework tailored for learning

with noisy labels. It is built around two pivotal modules: the LID-based noisy label discriminator (LID-dis) and the LID-guided label generator (LID-gen). The LID-dis module processes the features and label of a training sample to create a refined representation. This process unveils that the Local Intrinsic Dimensionality (LID) score is adept at distinguishing between correct and incorrect labels. Capitalizing on this finding, CoLafier employs the LID scores from LID-dis to assign weights in our specialized loss function, guiding both LID-dis and LID-gen during training. These modules work in tandem to calibrate label updates, mitigating error propagation. We incorporate dual augmented views for each instance, with their respective LID scores steering the weighting and label correction strategies. Upon completion of training, LID-gen stands equipped for deployment as the classifier. Evaluations across multiple noise settings confirm that CoLafier delivers a significant boost in prediction accuracy, on average outperforming SOTA techniques. **This work is published at SDM 2024 [47].**

Task 3: LLM-based Two-Level Foodborne Illness Detection Label Annotation with Limited Labeled Samples

In this task, we are provided with a dataset of social media posts [4] aimed at detecting foodborne illness incidents, which operates on two levels. At the post level, the goal is to predict whether a post indicates a foodborne illness incident. In contrast, at the word level, the objective is to identify mentions of slots, aka entities (e.g., Food, Symptom) related to the foodborne illness incident. Most of the posts in the dataset are unlabeled on both levels, with a small portion of posts annotated with noisy labels for the two levels. Our goal is to develop an in-context learning method that can assign two-level labels for unlabeled posts in this dataset in particular. It is expected that the proposed methods show some promise, and thus this could contribute to tools equally applicable to alternate domains of interest.

For this task, we propose leveraging Large Language Models (LLMs) as annotators

by designing an LLM-based framework that produces both word-level and post-level labels for each unlabeled post. This solution is composed of three stages. Initially, an LLM is prompted to generate word-level labels, utilizing the CoT method to guide the model to first assess whether the overall context suggests a foodborne illness incident and subsequently to identify which words constitute relevant elements. In the second stage, the LLM evaluates the accuracy of the relevant entities extracted in the first stage, preventing spurious relevant entities from compromising the post-level prediction in the final stage. Lastly, the LLM uses the results from the word-level labeling as a reasoning basis for determining the post-level label. To minimize potential biases introduced by demonstration examples, different sets of examples are employed at each labeling stage. Our method's efficacy, particularly in resource-constrained environments, has been rigorously evaluated, revealing a significant improvement in labeling accuracy over existing state-of-the-art (SOTA) supervised learning approaches that typically rely on extensive parameter updates and computational resources. Moreover, our method achieves labeling quality comparable to that obtained through crowdsourcing but at a fraction of the cost. This comparison elucidates the practical advantages of our LLM-based approach, particularly its efficiency and cost-effectiveness relative to both traditional supervised learning models and conventional crowdsourced annotation methods. **This work will be submitted to CIKM 2024.**

1.4 Organization of this Dissertation

The rest of this dissertation document is structured as follows:

- Chapter 2: Learning from Two-Level Labeled Datasets with One Level Having Complete Labels and the Other Having Incomplete Labels.

  This chapter covers Task 1.

- Chapter 3: Learning from Datasets with Noisy Labels.

This chapter covers Task 2.

- Chapter 4: In-context Learning from Two-Level Labeled Datasets with Incomplete Labels.

  This Chapter covers Task 3.

- Chapter 5: Conclusion.

  This chapter covers a summary of the key contributions of this dissertation is provided, along with promising directions for future work.

- Chapter 6: List of Publications.

  This chapter covers a list of publications authored and co-authored during my PhD studies.

Chapter 2

Explainable Text Classification with Limited Human Attention Supervision

This research work was published at IEEE BigData 2021[46], with me serving as the principal author in collaboration with Cansu Sen, Jidapa Thadajarassiri, Thomas Hartvigsen, Professor Xiangnan Kong, and Professor Elke Rundensteiner. I was responsible for designing the model for the problem of explainable text classification under limited human attention supervision and conducing experiments. What follows is an abridged version of this work.

2.1 Motivation

Text classification is a crucial text mining task with broad applications including fake news detection [48], clinical diagnosis [49], and sentiment analysis [50]. With the availability of massive training corpora, several modern approaches [51, 52, 53] achieve impressive performance. Yet they are remain largely inapplicable in settings where explanations are required to support a decision. For example, a doctor must know on what information a diagnostic model relies before trusting its predictions. Attention-based models [54, 55, 56] can be used

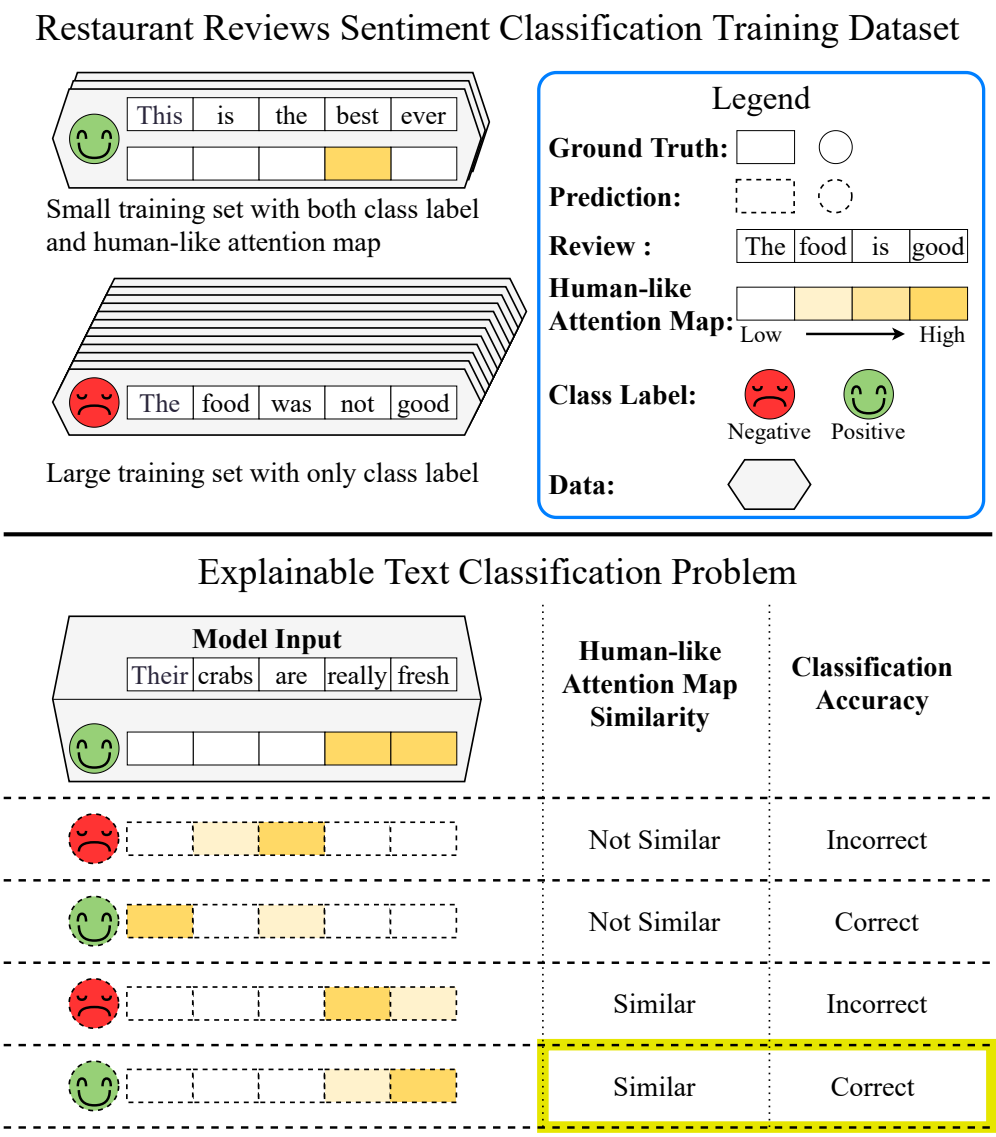


Figure 2.1: Explainable text classification with limited human attention supervision. Given a corpus of documents, each with a document-level label for classification task while only a few with word-level labels (human attention maps) for supervising attention, the dual goal is to learn a model that classifies text documents accurately and generates human-like word attention maps.

to acquire such explanations by learning to assign heavy weights to words that have a high impact on a model's prediction. Recently, there is growing evidence that attention weights that look as if they were generated by humans lead to both better explanations and sometimes even improved classification [57, 58]. However, attention generated by conventional attention approaches are dissimilar to human rationales [59, 58]. Classic attention contradicts

the ultimate goal of producing *explainable* models that allow human users to understand a model's rationale for a given prediction. Recent works [60, 61, 62, 63, 20, 19] have begun to overcome this hurdle, enhancing *explanations* by encouraging them to be *human-like*, or resemble rationales provided by humans. This has been achieved by collecting additional attention labels and explicitly *supervising* the attention mechanism.

Problem Definition. In this work, we are the first to study the problem of *explainable text classification with limited human attention supervision*. This addresses the real-world case where access to HAMs is severely limited. As illustrated in Figure 2.1, assume we are given a set of training documents, each with one associated classification label. A very small proportion of these training documents also have fine-grained word-level labels (HAMs), indicating which words a human annotator found to be most relevant as they assigned the class label. Our goal is to train a model that simultaneously solves the text classification task accurately while predicting *human-like attention weights* that are similar to those that would be generated by a human for the given document.

Challenges. Text classification with limited human attention supervision is challenging for the following reasons.

• *Sensitivity of Attention to Changing Contexts.* A word with high human attention in one document does not necessarily have high human attention in the other document. This implies that the attention weight for a word relies heavily on the context in which it appears. A successful attention method must effectively capture this reliance between context and human-like attention.

• *Conflict Between Human-Like Attention Generation and Text Classification.* Our problem requires a model to assign specialized weights to individual words. However, every word contributes to the classification task. Therefore, unsupervised attention weights are often more distributed across a sentence than a HAM. A successful model must balance between the two contradictory objectives of human-likeness and classification accuracy.

• *Varying Levels of Supervision.* This problem has two tasks: classification and human-like attention generation. However, each of the two tasks has a different amount of labeled data — all data have classification labels, only some have human attention maps. A good solution must balance the feedback given from each task without overemphasizing the fully-labeled task.

Proposed Method. To handle these challenges, we propose the deep learning architecture, HELAS: Human-like Explanation with Limited Attention Supervision, which produces human-like attention values during text classification, even when very few human attention labels are available. HELAS processes input text in three phases : (1) HELAS encodes input text through a *text representation learner* into both dense vectors for each word and one vector for the whole document. This text representation learner is highly modular and can learn representations using many recent text models such as RNNs [64, 65] or BERT [52]. (2) The *human-like attention learner* in HELAS learns human-like attention weights for each word by both considering its individual impact on the classification task *and* by carefully incorporating its contextual information. This allows the learned attention mechanism to be adaptive to context, similar to a human annotator. (3) The *contextualized representation* collates the contextualized information learned according to the human-like attention learner with the overall text representation to consider *both* sources or information and perform the final classification. Thus, our approach succeeds to capture the unique contribution of each word in a given document and produce both human-like attention and accurate classifications.

HELAS is optimized using a joint loss function for the classification and human-like attention-learning tasks. We introduce a hyper-parameter into the loss function for striking a balance between classification and attention supervision, resulting in one unified training objective. This newly defined loss handles the varying levels of supervision for both classification and attention supervision and thus allows HELAS to deliver accurate classifications and human-like attention weights simultaneously.

Contributions. Our contributions are as follows:

• We define the open problem of explainable text classification with limited human attention supervision, which is to develop a human-like explainable classifier when few HAMs are available.

• We propose the first solution to this problem, HELAS, which contains two key components: (1) a novel attention method, called human-like attention learner, that successfully learns human-like attention weights, adapting to different contexts, and (2) a custom contextualized representation that considers the impact of all words to make its final prediction.

• We propose a joint loss function for HELAS that balances the limited attention supervision and fully-supervised classification supervision, encouraging the model to generate more human-like attention values – even with very few HAMs.

• We demonstrate that even when HAMs are available for as little as 2% of the training data, HELAS still succeeds to generate human-like attention, achieving up to 22% increase in similarity compared to four state-of-the-art methods. HELAS also gets better performance on the classification task achieving significant (up to 19%) gains in accuracy.

2.2 Related Works

Supervised Attention Models. Attention supervision is used for NLP problems. In [66, 67], conventional alignment models are used to guide the attention module for language translation. [62] apply supervised attention method for event detection, namely, their model focuses on event information on both the word- and sentence-level. [63] introduce supervised attention for improving the accuracy of the semantic event recognition; namely, by deploying semantic word lists and dependency parsing trees [68] to guide the attention components. [19] propose a method to use estimated human attention derived from eye-tracking corpora to regularize attention functions for sequence classification tasks. While these works show

supervised attention can improve accuracy, the forms of guidance adopted remain limited – none of the methods mentioned above get attention guidance via word-level human attention maps collected for the classification task.

[20] propose a model with an attention mechanism for text classification that jointly exploits document classification labels and sentence-level annotation labels. They assume that annotators explicitly mark sentences that support their overall document categorization for each document in the corpus. However, collecting fine-grained sentence-level or word-level annotation labels for all instances in a dataset can be costly and time-consuming. Moreover, in [20], training with each level of labels is split into two steps. It is time-consuming and sophisticated to train their model. Hence, it is worthy of exploring a method that can be trained efficiently with limited access to HAMs.

Model Explainability. Deep-learning models suffer from a lack of explainability, despite the need for explainable models in many domain settings. Thus, several studies in recent years attempt to make neural network models more explainable. Rationale-based methods are examples of this for NLP [69, 70]. In these works, the goal is to train a classification model and produce binary “rationales" to serve as human-like explanations of model predictions. However, while their direction is promising, their classification performance remains a drawback compared to recent attention-based approaches [58]. Also, these rationale-based architectures make classifications based on the selected “rationales”, not the full text [69, 70]. So the information in these non-rationale words is missing during prediction.

Recent work in deep learning instead has begun to use attention mechanisms to attempt to bring interpretability to model predictions [54, 56, 55]. However, these works assess the produced attention maps solely qualitatively by visualizing a few hand-selected instances. [58] approaches attention explainability from a human-centered perspective. They investigate the similarity between human attention and machine attention and interpret such similarity as a measurement of the model explainability. It indicates that it is intuitive to humans as

it matches which words humans would rely on when making decisions. [58] makes a novel human attention map resource available to the community. Inspired by their approach, we now leverage human attention to explicitly train a model to concurrently produce the overall task prediction as well as the *human-like explanations* with the power of modern attention mechanisms.

2.3 Methodology

2.3.1 Problem Definition

In this paper, we study the problem of *explainable text classification with limited human attention supervision*. Given a set of N documents $\mathcal{I} = \{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$, each document $\mathcal{D}^i$ consists of T words $\mathcal{D}^i = [w_1^i, \ldots, w_T^i]$, and a set of class labels $y^i = [y_1^i, \ldots, y_K^i]$, where K is the cardinality of y^i, $y_k^i \in \{0, 1\}$ and $\sum_{k=1}^{K} y_k^i = 1$. The *document classification* task is to parameterize a function $f_\theta(\cdot)$ that maps $\mathcal{D}^i \rightarrow y^i$, generalizing to unseen instances.

A *Human Attention Map* (HAM) is a vector of length T, $[\alpha_1, \ldots, \alpha_T]$, where each entry α_t indicates the degree of attention that a human pays to a corresponding word w_t in a document. HAM is a binary map collected from humans, *i.e.*, $\alpha_t = 1$ indicates that the corresponding word receives high attention while 0 shows low attention. A *Machine Attention Map* (MAM $= [\hat{\alpha}_1, \ldots, \hat{\alpha}_T]$) is a human-like attention map *predicted* by a neural network model, where $\hat{\alpha}_i \in [0, 1]$ indicates the probability of the corresponding word that would receive high attention from humans.

For each document $\mathcal{D}^i$, we are given a class label y^i. However, only a limited amount of documents have HAMs. One component of $f_\theta(\cdot)$ is an attention mechanism that aims to output MAMs that are similar to HAMs. Our task is to jointly learn the function $f(\theta)$ while minimizing the difference between HAM^i and MAM^i for all documents $\mathcal{D}^i$, the latter task

is named *human-like attention generation*. If conditioned perfectly, $f_\theta(\mathcal{D}^i) \rightarrow (\hat{y}^i, MAM^i)$ such that $\hat{y}^i = y^i$ and $\text{MAM}^i = \text{HAM}^i$.

For readability, we henceforth describe our method for a single document D^i, dropping i when it is unambiguous.

2.3.2 Proposed Method: HELAS

Our proposed deep learning architecture, HELAS: Human-like Explanation with Limited Attention Supervision, is depicted in Figure 2.2. HELAS consists of three major components: (1) The *text representation learner* encodes raw text to their numerical representations. This component can be any sequential deep learning architecture, such as RNNs [64, 65] or BERT [52]. The purpose of this layer is to encode the input document into a document representation and a sequence of word representations. (2) The *human-like attention learner* generates a MAM aimed to be similar to the given HAM. The attention mechanism determines the human-like attention weight for each word by the interrelation between word and sentence representations. (3) The *contextualized representation* utilizes the MAM from the human-like attention learner to enhance the context vector to estimate the class label, y, of a document.

Text Representations Learning. We focus our study on the two popular sequence modeling including RNNs [64, 65] and BERT [52] while HELAS can be, in practice, paired with any sequence-representation learning architectures.

• **HELAS-RNN.** One common and powerful architecture for document classification is an RNN combined with an attention mechanism [54, 71]. Following this architecture, the HELAS-RNN model first utilizes an encoding layer to map words into real-valued vector representations where semantically-similar words are mapped close to one another. We use a pre-trained word embedding set ϕ for this mapping: $x_t = \phi w_t$. HELAS-RNN then employs a recurrent layer to embed vector representations of words into hidden states, processing words

once at a time. In our experiments, we use both LSTM [64] and GRU [65] memory cells. Assuming that Γ is the recurrence function (*e.g.*, LSTM or GRU) and x_t is the embedded t-th word from the document $\mathcal{D}$, HELAS-RNN is modeled as:

$$e_t = \Gamma(x_t, e_{t-1}) \tag{2.3.1}$$

where e_t is the hidden state. The final hidden state e_T is used as the document representation, defined as $r = e_T$.

• **HELAS-BERT.** HELAS-BERT first employs a transformer architecture [51] to encode words, initialized with a pre-trained BERT model [52]. Following the standard practice in BERT-based architectures, the first word of the input is the special token '[CLS]'. '[SEP]' token is added to the end of the input sequence to denote the end. '[PAD]' token is used to pad the sequence in case the input sequence is shorter than the maximum input length supported by the BERT model. HELAS-BERT generates two outputs. First is a sequence of learned word representations $[e_1, ...e_T]$ for each input word. Second is a vector representation r for the whole input document. This vector r corresponds to the output of the '[CLS]' token further processed by a linear layer and a tanh activation function.

$$[e_1, ...e_T], r = \text{BERT}([w_1, ...w_T]) \tag{2.3.2}$$

Human-Like Attention Generation. The goal is to generate attention scores to be as close as possible to human attention. This way, attention scores can be interpreted as human-like explanations for the final classification decision.

We hypothesize that the importance of each word relies heavily on its belonging document. Therefore, HELAS is designed to learn specialized attention function that is adaptable

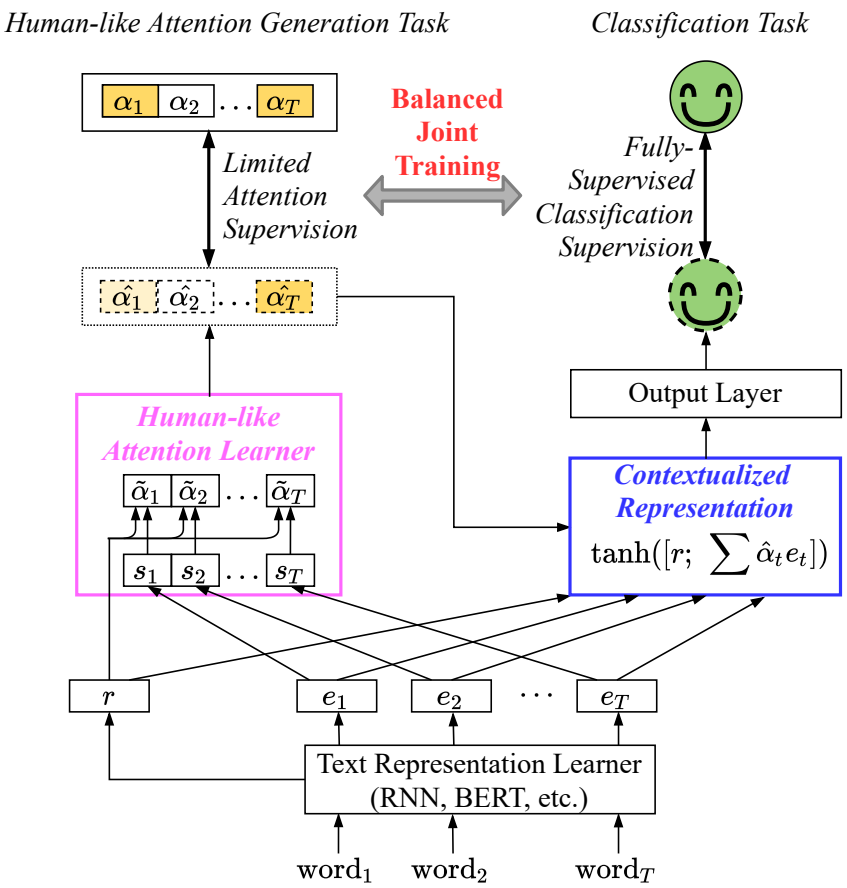


Figure 2.2: Overall architecture of HELAS.

for different training corpus. The human-like attention learner learns MAMs as follows:

$$s_t = \tanh(W_e e_t + b_e) \tag{2.3.3}$$

$$\tilde{\alpha}_t = s_t^\top r \tag{2.3.4}$$

$$\hat{\alpha}_t = \text{sigmoid}(\tilde{\alpha}_t) \tag{2.3.5}$$

where W_e and b_e are weight matrix and bias term in the linear layer, which can be optimized during the training time. Here, we first encode new word representations s_t from e_t. s_t serves as a specialized representation of the importance of w_t, while e_t is still a general representation of the information contained in w_t. Then the raw attention score $\tilde{\alpha}$ is determined by such s_t and the document representation r in order to induce MAMs to capture more flexible relation between e_t and r. MAM $= [\hat{\alpha}_1, \ldots, \hat{\alpha}_T]$ is then utilized by the subsequent layers of the HELAS. To this end, we take the binary cross entropy as the general loss of the attention

at the word level.

$$J_a(\text{HAM}, \text{MAM}) = -(\sum_{t=1}^{T}(\alpha_t \log \hat{\alpha}_t + (1-\alpha_t)\log(1-\hat{\alpha}_t))). \tag{2.3.6}$$

This objective optimizes the model to assign human-like attention scores to every word. By providing word-level supervision to the document classification model, we are able to teach it to focus on the most relevant areas selected by humans and thereby improve the quality of document representations along with the overall performance.

It is worth noting that special tokens, such as '[CLS]', '[SEP]' and '[PAD]', are invisible to human annotators (if the text representation learner is BERT). Thus, their corresponding human attention weights are always set to zero. Also, the tokenizer used by the BERT model is WordPiece [72], which sometimes splits a word into several words. These generated words are then assigned with the same human attention score as the original word.

Document Classification. Using the MAMs, the learned word representations e_t and the document representation r generated by the text representation learner, the contextualized representation c is computed as follows:

$$c = \tanh([r; \sum_t \hat{\alpha}_t e_t]). \tag{2.3.7}$$

Unlike the previous works [19, 20] which use $\sum_t \hat{\alpha}_t e_t$ as the final text representation for classification, we concatenate document representation r and weighed sum of word representations to model a dense embedding for the document. For a given HAM, when $\alpha = 0$, it does not indicate that the corresponding word was completely ignored by humans. During training, the values of some $\hat{\alpha}$s could be very close to 0. Since r contains basic information of the whole document, the contextualized representation will consider every word when performing classification, even if some words' associated $\hat{\alpha} \approx 0$. Those words with higher $\hat{\alpha}$

values remain a higher impact on final prediction results.

The output layer uses c as follows:

$$d = \text{Dropout}(W_c c + b_c) \tag{2.3.8}$$

$$\hat{p}(y_k = 1|D) = \frac{\exp(W_d^{(k)} d + b_d)}{\sum_{k=1}^{K} \exp(W_d^{(k)} d + b_d)} \tag{2.3.9}$$

To further fuse r and $\sum_t \hat{\alpha}_t e_t$ together and reduce the risk of overfitting, we apply a linear transformation followed by a dropout layer in Equation 2.3.8. Here, W_c, b_c are weight matrix and bias term in the linear layer. After dropout, Equation 2.3.9 assigns a probability to each possible class, where W_d, b_d are weight matrix and bias term in the softmax function. We use the cross-entropy loss as the document classification objective function where $\hat{p}(y_k = 1|D)$ is the prediction and y the ground truth label.

$$J_c(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log\left(\hat{p}(y_k = 1|D)\right). \tag{2.3.10}$$

Joint Training of HELAS. In HELAS, the human-like attention generation task and classification task are jointly trained. Thus, we define a joint loss function in the training process upon the losses specified for different subtasks as follows:

$$J(\theta) = \sum (J_c(y, \hat{y}) + \lambda J_a(\text{HAM}, \text{MAM})). \tag{2.3.11}$$

where θ denotes, as a whole, the parameters used in our model, and λ is the hyper-parameter for striking a balance between document classification supervision and attention supervision. When only a few documents contain HAMs, the tunable parameter λ can be optimized to emphasize the small corresponding supervision signals, then both the classification and the human-like explanation goals can be achieved evenly.

During the training process, if there is no HAM for the input text, we only minimize

Equation 2.3.10. When both HAMs and classification labels are available, we minimize Equation 2.3.11.

2.4 Experiments

We evaluate our proposed method on four publicly available datasets that are compared against four state-of-the-art methods.

2.4.1 Datasets

The four datasets used in our experiments contain document labels for all instances while only a few of them have HAMs. All datasets contain roughly balanced examples between positive and negative classes. The proportions of positive examples are between 45% to 68%. It should be noted that our work can also be applied to multi-class datasets.

• **Yelp-HAT [58].** This dataset provides human attention maps for a collection of 1000 reviews from the Yelp dataset. Each review comes with a human attention map and a class label indicating whether the review is positive or negative. All characters are lowercase, punctuation is removed. Reviews are 50-75 words long. 70% of reviews are used for training with the remaining 30% for testing.

The dataset contains annotations from *multiple* humans for each of the reviews because each annotator may have different opinions on how indicative words are for review sentiments. To obtain reliable representations of human attention, we apply Consensus Attention Maps as being used in [58], by extracting HAMs from all annotators' agreement that are then used to evaluate a sentiment classification task.

• **N2C2.** N2C2 NLP Research datasets contain unstructured notes from the Research Patient Data Repository at Partners Healthcare[1]. From this clinical note repository, we use

[1] `https://n2c2.dbmi.hms.harvard.edu`

the 2014 challenge data, consisting of a set of medical documents that track the progression of heart disease in diabetic patients. Each clinical note is assigned to an expert in order to indicate the presence and progression of a disease (diabetes or heart disease), associated risk factors, and the time they were present in the patient's medical history.

In this dataset, we focus on predicting heart disease. For each patient in the dataset, if there is a clinical note with a heart disease annotation (indicated by CAD tag), we assign all notes belonging to this patient to the positive class. Patients with no heart disease mention are assigned to the negative class. Then we train a model that inputs every individual clinical note and predicts whether this note belongs to a heart-disease patient. N2C2 dataset contains 520 clinical notes in the training set and 511 clinical notes for the testing set. A series of notes from the same patient is assigned into either the training or testing set.

We use all heart disease-related words, as outlined by the annotation guidelines of 2014 Heart Disease Risk Factors Challenge of n2c2 NLP Research Data Sets[2], to create human attention maps. These include remarks of patients having heart disease (e.g., "coronary artery disease") or indirect mentions (e.g., "unstable angina," "PLAVIX" - a blood thinner used to prevent heart attack).

• **Movie Reviews [73].** Each review comes with a positive/ negative sentiment label and human annotation on word-level. Due to the length constraint of our model, we used the first 200 words as text input. Reviews in which the first 200 words are all labeled 0 are dropped. After preprocessing, there are 1,241 reviews in the training set and 320 reviews in the testing set.

• **Standard Sentiment Treebank (SST) [74].** This dataset contains 9,545 sentences in the training set and 2,310 sentences in the testing set. Each sentence comes with a binary classification label (positive or negative). The original data do not contain human attention annotation. We randomly selected 400 sentences from the dataset (200 from training

[2]`https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/`

split and 200 from testing split, positive/negative sentences ratio 1:1). Then we asked four researchers in our groups to annotate words that are indicative of the review sentiment in each sentence.

2.4.2 Metrics

The following two metrics are used for evaluation:

Behavioral Similarity [58]. To evaluate the explainable nature of each method, we use the *Behavioral Similarity* metric proposed in [58]. This metric measures the similarity between human and machine attention maps via the Area Under the ROC Curve:

$$B(\text{HAM}, \text{MAM}) = \frac{1}{|\mathcal{D}|} \sum_i \text{AUC}(\text{HAM}^i, \text{MAM}^i) \tag{2.4.1}$$

where $|\mathcal{D}|$ is the number of documents in dataset $\mathcal{D}$. Behavioral similarity ranges between 0 and 1.

Accuracy. We use standard classification accuracy to measure the sequence classification performance.

2.4.3 Implementation Details

We implement the text representation learners as LSTM/GRU with 128-dimensional hidden states and BERT [52]. The learning rates are 1e-3 and 2e-5 for LSTM/GRU and BERT, respectively. The LSTM/GRU model is trained for 40 epochs, while the BERT model is trained for 20 epochs. All three models are set the dropout rate at 0.2 and optimized using Adam [75]. We did a hyperparameter search for λ in the joint loss function. The best λ for LSTM model is 20, for GRU is 30, and for BERT is 4. All experiments are implemented on PyTorch [76] and run on a Tesla V100 GPU.

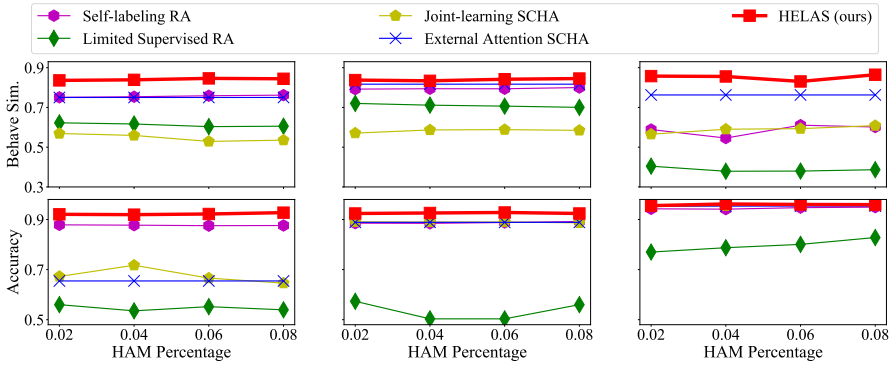
Table 2.1: Performance of three text representation learners LSTM, GRU, and BERT on three tasks of Yelp-HAT Sentiment Classification, N2C2 Heart Disease Prediction, and Movie Reviews Sentiment Classification, with 2% of training data having HAMs. Metrics: (1) Behavioral similarity for human-like attention generation task, and (2) Accuracy for classification task.

Dataset	Methods	LSTM		GRU		BERT	
		Behave Sim.	Accuracy	Behave Sim.	Accuracy	Behave Sim.	Accuracy
Yelp-HAT	Limited Supervised RA	.62 ± .03	.56 ± .04	.72 ± .01	.57 ± .03	.40 ± .01	.77 ± .03
	Self-labeling RA	.75 ± .01	.88 ± .02	.79 ± .01	.89 ± .01	.59 ± .06	.94 ± .01
	External Attention SCHA	.75 ± .07	.65 ± .05	.82 ± .00	.89 ± .01	.76 ± .03	.95 ± .00
	Joint-learning SCHA	.57 ± .01	.67 ± .06	.57 ± .02	.89 ± .02	.57 ± .03	.95 ± .01
	HELAS (ours)	**.84** ± .00	**.92** ± .01	**.84** ± .00	**.92** ± .00	**.86** ± .01	**.96** ± .00
N2C2	Limited Supervised RA	.90 ± .01	.62 ± .05	.91 ± .00	.72 ± .01	.48 ± .05	.69 ± .01
	Self-labeling RA	.92 ± .00	.76 ± .00	.91 ± .01	.76 ± .00	.68 ± .06	.77 ± .00
	External Attention SCHA	.56 ± .02	.76 ± .00	.62 ± .02	.76 ± .00	.46 ± .05	.76 ± .00
	Joint-learning SCHA	.52 ± .06	.68 ± .00	.70 ± .07	.76 ± .00	.49 ± .05	.76 ± .00
	HELAS (ours)	**.93** ± .00	**.78** ± .00	**.92** ± .00	**.77** ± .00	**.73** ± .05	**.78** ± .01
Movie Reviews	Limited Supervised RA	.54 ± .01	.54 ± .00	.56 ± .03	.54 ± .00	.42 ± .01	.58 ± .03
	Self-labeling RA	.53 ± .02	.58 ± .04	.54 ± .02	.63 ± .06	.56 ± .03	.83 ± .02
	External Attention SCHA	.61 ± .02	.54 ± .00	.61 ± .01	.54 ± .00	.58 ± .01	.86 ± .01
	Joint-learning SCHA	.50 ± .02	.54 ± .00	.46 ± .01	.54 ± .00	.58 ± .02	.86 ± .00
	HELAS (ours)	**.69** ± .01	**.77** ± .00	**.69** ± .02	**.76** ± .01	**.80** ± .01	**.87** ± .01
SST	Limited Supervised RA	.81 ± .00	.66 ± .00	.88 ± .02	.68 ± .01	.82 ± .06	.78 ± .01
	Self-labeling RA	.84 ± .02	.71 ± .01	.86 ± .01	.72 ± .00	.96 ± .00	**.87** ± .00
	External Attention SCHA	.89 ± .00	.54 ± .00	.89 ± .00	.54 ± .00	.84 ± .04	**.87** ± .00
	Joint-learning SCHA	.50 ± .00	.54 ± .00	.50 ± .00	.54 ± .00	.47 ± .06	**.87** ± .00
	HELAS (ours)	**.91** ± .00	**.77** ± .00	**.91** ± .00	**.77** ± .00	**.97** ± .00	**.87** ± .00

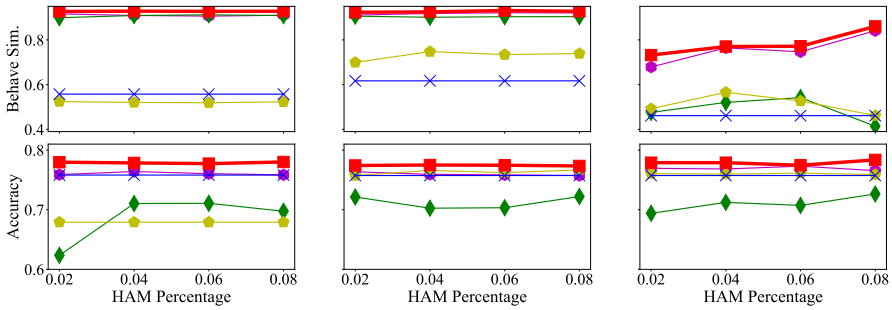
For Yelp-HAT dataset, we did a random train-test split every time. For N2C2, Movie Review, and SST datasets, we used the defined train-test splits every time. For Yelp-HAT, N2C2, and Movie Review datasets, the training data with and without HAMs are randomly assigned every time. We use the pre-trained BERT-base-uncased model from the "Transformers" library[3]. [77]

For each experiment, we save the model with the highest accuracy during training and report the average evaluation results of each model from 5 replications that are initialized randomly. When we train a model with LSTM or GRU as the text representation learner, words are embedded using 100-dimensional GloVe [78] for Yelp-HAT and Movie Reviews. For N2C2 dataset, we use the pre-trained embeddings from BioMed [79]. When the text

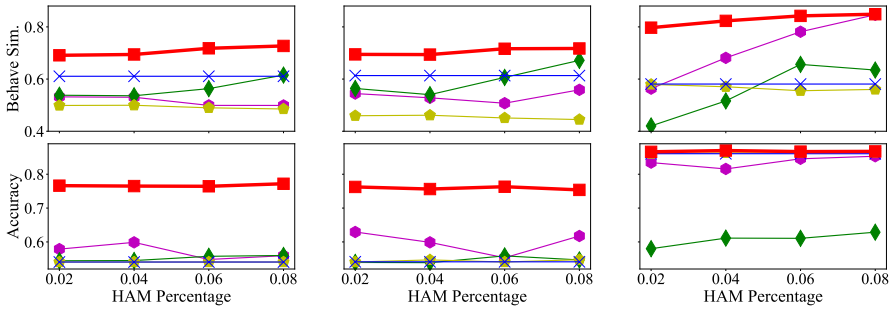
[3] `https://github.com/huggingface/transformers`



(a) Yelp-HAT dataset with text representation learners: LSTM, GRU, and BERT shown on columns 1, 2 and 3, respectively.



(b) N2C2 dataset with text representation learners: LSTM, GRU, and BERT shown on columns 1, 2 and 3, respectively.



(c) Movie Reviews dataset with text representation learners: LSTM, GRU, and BERT shown on columns 1, 2 and 3, respectively.

Figure 2.3: Compared performance on three datasets: Yelp, N2C2, and Movie Reviews. For each dataset, we experiment with three different text representation learner LSTM, GRU, and BERT. We vary the proportions of available HAMs in the training dataset as shown on the x-axis, ranging from 2%, 4%, 6%, and 8%. Metrics: (1) Behavioral similarity for human-like attention generation task, and (2) Accuracy for classification task are both plotted.

representation learner is BERT, we use the WordPiece embedding [72] provided by BERT model for all three datasets. All code and further training settings are publicly available[4].

[4] `https://github.com/zdy93/HELAS`

2.4.4 Experimental Results

We first evaluate HELAS's capacity to learn from very limited levels of HAMs, specifically focusing on the case where only 2% of training data have HAMs. As always, the entire training dataset still has classification labels. In this experiment, we measure the behavioral similarity and the accuracy of all compared methods on the Yelp-HAT, N2C2, and Movie Reviews dataset, randomly down-sampling the HAM annotations to 2%.

Our results are shown in Table 2.1, where we first observe that our HELAS models achieve superior behavioral similarity and accuracy compared to all baseline models.

For the *Yelp-HAT sentiment classification task*, all HELAS models achieve significant gains in behavioral similarity (up to 9%) compared to the baseline models. HELAS with LSTM achieves the most substantial improvement in accuracy by 4%. Great gains in behavioral similarity indicate that the human-like attention learner in HELAS models can better mimic the relation between context and human-like attention even with a limited amount of word-level labels more. The HELAS models show improvement in the classification accuracy for all three core sequence algorithms, with HELAS-BERT achieving the least gain. This is likely because the HELAS-BERT model is pre-trained on a large text corpus, whereas HELAS-LSTM and HELAS-GRU models are being trained from scratch on a small dataset. This causes these baseline BERT models to achieve an already high accuracy, which is challenging to improve upon.

For the *N2C2 heart disease prediction task* and *movie reviews sentiment classification task*, we observe similar trends as for the Yelp-HAT sentiment classification task. We observe the largest gains in the classification accuracy for HELAS-LSTM and HELAS-GRU models compared to HELAS-BERT over the baseline methods. Improvement in behavioral similarity is significant (up to 22%) for all core algorithms.

For the *SST sentiment classification task*, the results show again that our method out-

performs the other alternative methods on the behavior similarity. Both HELAS-LSTM and HELAS-GRU methods shown improvement in behavioral similarity by 2% and accuracy by around 5-6%. Most methods benefit from rich discriminative signals on this task and reach comparable performance when pairing with the BERT model.

Further results on other percentages of HAM availability are shown in Figure 2.3. Because we only labeled 400 sentences in the SST dataset, we did not conduct experiments on other percentages of HAM availability for the *SST sentiment classification task*. We observe that our HELAS models keep outperforming state-of-the-arts baselines across three tasks as the HAM proportion increases from 2% to 8%. Note that External Attention SCHA. utilizes an external source of HAMs and has no access to HAMs in classification task datasets, so its performance remains unchanged as HAM proportion increases.

2.5 Conclusion

For this task, we define the open problem of explainable text classification with limited human attention supervision, with the aim to support the real-world setting in that human attention maps (HAMs) are often scarce. We propose the first solution to this problem, named HELAS: Human-like Explanation with Limited Attention Supervision. Our proposed method contains two key components: a human-like attention learner that successfully learns human-like attention weights conditioned on context information, and a carefully designed contextualized representation that considers the contribution from all words to classify the document into a final class. Our specially-designed joint loss function balances the supervision signals from both the *human-like attention generation* and *document classification* tasks simultaneously, despite them having drastically different numbers of labels across training instances.

Our evaluation studies on three real-world datasets demonstrate that HELAS outper-

forms state-of-the-art alternatives on both learning an accurate text classifier and generating human-like attention, even when as little as 2% of the data contain HAMs. This result is consistent across different text representation learners from LSTM, GRU, to BERT.

Chapter 3

Classification with Noisy Labels

This work is published in SDM 2024, with me serving as the lead author, alongside Ruofan Hu and Professor Elke Rundensteiner. My contributions include the development of the solution framework and the execution of experiments. Below is an abridged version of this work.

3.1 Motivation

Deep neural networks (DNNs) have achieved remarkable success in a wide range of machine learning tasks [49, 46, 80]. Their training typically requires extensive, accurately labeled data. However, acquiring such labels is both costly and labor-intensive [1, 81, 82]. To circumvent these challenges, researchers and practitioners increasingly turn to non-expert labeling sources, such as crowd-sourcing [4] or automated annotation by pre-trained models [23]. Although these methods enhance efficiency and reduce costs, they frequently compromise label accuracy [4]. The resultant 'noisy labels' may inaccurately reflect the true data labels. Studies show that despite their robustness in AI applications, DNNs are susceptible to the detrimental effects of such label noise, which risks impeding their performance and

also generalization ability [25, 1].

State-of-the-Art. Recent studies on learning with noisy labels (LNL) reveal that Deep Neural Networks (DNNs) exhibit interesting memorization behavior [25, 83]. Namely, DNNs tend to first learn simple and general patterns, and only gradually begin to learn more complex patterns, such as data with noisy labels. Many methods thus leverage signals from the early training stage [35], such as loss or confidence scores, to identify potentially incorrect labels. For label correction, the identified faulty labels are either dropped, assigned with a reduced importance score, or replaced with generated pseudo labels [32, 36, 84].

However, these methods can suffer from accumulated errors caused by incorrect selection or miscorrection - with the later further negatively affecting the representation learning and leading to potential overfitting to noisy patterns [1, 85]. Worse yet, most methods require prior knowledge about the noise label ratio or the specific pattern of the noisy labels [32, 35]. In real-world scenarios, this information is typically elusive, making it difficult to implement these methods.

Local Intrinsic Dimensionality (LID), a measure of the intrinsic dimensionality of data subspaces [86], can be leveraged for training DNNs on noisy labels. Initially, LID decreases as the DNN models the low-dimensional structures inherent in the data. Subsequently, LID increases, indicating the model's shift towards overfitting the noisy label. Another study [87] applied LID to identify adversarial examples in DNNs, which typically increase the local subspace's dimensionality. These findings suggest LID's sensitivity to noise either from input features or labels. Nonetheless, previous research has utilized LID as a general indicator for the training stages or for detecting feature noise. While a promising direction for research, applying LID for detecting mislabeled samples has not been explored before.

Problem Definition. In this study, we propose a method for solving classification with noisy labeled training data. Given a set of training set with each item labeled with one noisy classification label, our goal is to train a robust classification model that solves the

classification task accurately without any knowledge about the quality or correctness of the given labels.

Challenges. Classification with noisy labeled training data is challenging for the following reasons:

• *Lack of knowledge about noise ratio and noise pattern.* Without knowledge about the noise ratio and noise pattern of the given dataset, it is challenging to develop a universal method that can collect sufficient clean labels to train a strong model.

• *Compounding errors in the training procedure.* Incorrect selection or correction errors made early in the learning process can compound, leading to even larger errors as the model continues to be trained. This can result in a model that is far off from the desired outcome.

Proposed Method. In response to these challenges, we conduct an empirical study to evaluate the effectiveness of the Local Intrinsic Dimensionality (LID) score as a potential indicator for mislabeled samples. We design a specialized classifier, namely, LID-based noisy label discriminator (LID-dis). LID-dis processes both a sample's features and label to predict the label. Notably, its intermediate layer yields an enhanced representation encompassing both feature and label information. Our uniquely crafted training scheme for LID-dis reveals that the LID score of this representation can effectively differentiate between correctly and incorrectly labeled samples. This differentiation is consistent across various noise conditions.

To complement LID-dis, we introduce the LID-guided label generator (LID-gen), a regular classification model that operates solely on the data's features - not requiring access to the label. LID-dis and LID-gen together as two subnets form our proposed framework, CoLafier: Collaborative Noisy Label purifier with LID guidance. During training, we generate two augmented views of each instance's features, which are then processed by both LID-dis and LID-gen. CoLafier consider the consistency and discrepancy of the two views' LID scores as produced by LID-dis to determine weights for each instance in our adapted loss function. This reduces the risk of incorrect weight assignment. Both LID-dis and LID-

gen undergo training using their respective weighted loss. Concurrently, LID-gen suggests pseudo-labels from these two augmented views for each training instance. LID-dis processes these pseudo-labels along with two views, deriving LID scores for them. These LID scores and the difference between prediction from LID-dis and LID-gen guide the decision on the label update. Information from the two views and two subnets together helps mitigate the risk of label miscorrection. After training is complete, LID-gen is utilized as the classification model to be deployed.

Contributions. Our contributions are as follows:

• We craft a pioneering approach to harness the LID score in the context of noisy label learning, leading to the development of LID-dis subnet. LID-dis processes not only a sample's features but also its label as input. This yields an enhanced representation adept at distinguishing between correct and incorrect labels across varied noise ratios and patterns.

• Drawing insights from the LID score, we introduce the CoLafier framework, a novel solution that integrates two LID-dis and LID-gen subnets. This framework utilizes two augmented views per instance, applying LID scores from the two views to weight the loss function for both subnets. LID scores from two views and the discrepancies in prediction from the two subnets inform the label correction decisions. This dual-view and dual-subnet approach significantly reduces the risk of errors and enhances the overall effectiveness of the framework.

• We conduct evaluation studies across varied noise conditions. Our findings demonstrate that, even in the absence of explicit knowledge about noise characteristics, CoLafier still consistently yields improved performance compared to state-of-the-art LNL methods.

3.2 Related Works

Learning With Noisy Labels. In recent studies, two primary techniques have emerged for training DNNs with noisy labels: sample selection and label correction. Sample selection approaches focus on identifying potentially mislabeled samples and diminishing their influence during training. Such samples might be discarded [32, 33], given reduced weights in the loss function [88, 5], or treated as unlabeled, with semi-supervised learning techniques applied [35, 36]. On the other hand, label correction strategies aim to enhance the training set by identifying and rectifying mislabeled instances. Both soft and hard correction methods have been proposed [88, 23, 37]. However, a prevalent challenge with these approaches is the amplification of errors during training. If the model makes incorrect selection or correction decisions, it can become biased and increasingly adapt to the noise. Another challenge arises when certain methods presuppose knowledge of the noise label ratio and pattern, using this information to inform their hyper-parameter settings [32, 36, 35]. However, in real-world scenarios, this information is typically unavailable, rendering these methods less practical for implementation.

Supervised Learning and Local Intrinsic Dimensionality. The Local Intrinsic Dimensionality (LID) [86] has been employed to detect adversarial examples in DNNs, as showcased by [87]. Their research highlights that adversarial perturbations, a specific type of input feature noise, tend to elevate the dimensionality of the local subspace around a test sample. As a result, features rooted in LID can be instrumental in identifying such perturbations. Within the Learning with Noisy Labels (LNL) domain, LID has been employed as a global indicator to assess a DNN's learning behavior and to develop adaptive learning strategies to address noisy labels [89]. However, it has not been utilized to identify samples with label noise.

In contrast to these applications, our study introduces a framework that leverages LID

to detect and purify noisy labels at the sample level. Using LID, we can differentiate between samples with accurate and inaccurate labels, and its insights further guide the decision to replace noisy labels with more reliable ones.

3.3 Methodology

This section is organized as follows: we first introduce the problem definition, then we demonstrate the utilization of the LID score to differentiate between true-labeled and false-labeled instances. Finally, we present our proposed method, CoLafier: Collaborative Noisy Label purifier with LID guidance.

3.3.1 Problem Definition

In this study, we address the problem of training a classification model amidst noisy labels. Let's define the feature space as $\mathcal{X}$ and $\mathcal{Y} = \{1, ..., N_c\}$ to be the label space. Our training dataset is represented as $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^{N}$, where each $\tilde{y}_i = [\tilde{y}_{i,1}, \tilde{y}_{i,2}, ..., \tilde{y}_{i,N_c}]$ is a one-hot vector indicating the *noisy label* for the instance x_i. Here, N_c denotes the total number of classes. If c is the noisy label class for x_i, then $\tilde{y}_{i,j} = 1$ when $j = c$; otherwise, $\tilde{y}_{i,j} = 0$. It is crucial to note that a noisy label, $\tilde{y}_i$, might differ from the actual ground truth label, y_i. An instance is termed a *true-labeled instance* if $\tilde{y}_i = y_i$, and a *false-labeled instance* if $\tilde{y}_i \neq y_i$. The set of all features in $\tilde{D}$ is given by $X = \{x_i | (x_i, \tilde{y}_i) \in \tilde{D}\}$. Our primary goal is to devise a classification method, denoted as $f(x; \Theta) \rightarrow \hat{y}$, which can accurately predict the ground-truth label of an instance. In this context, $\hat{y}_i = [\hat{y}_{i,1}, \hat{y}_{i,2}, ..., \hat{y}_{i,N_c}]$ is a probability distribution over the classes, with $\sum_{j=1}^{N_c} \hat{y}_{i,j} = 1$.

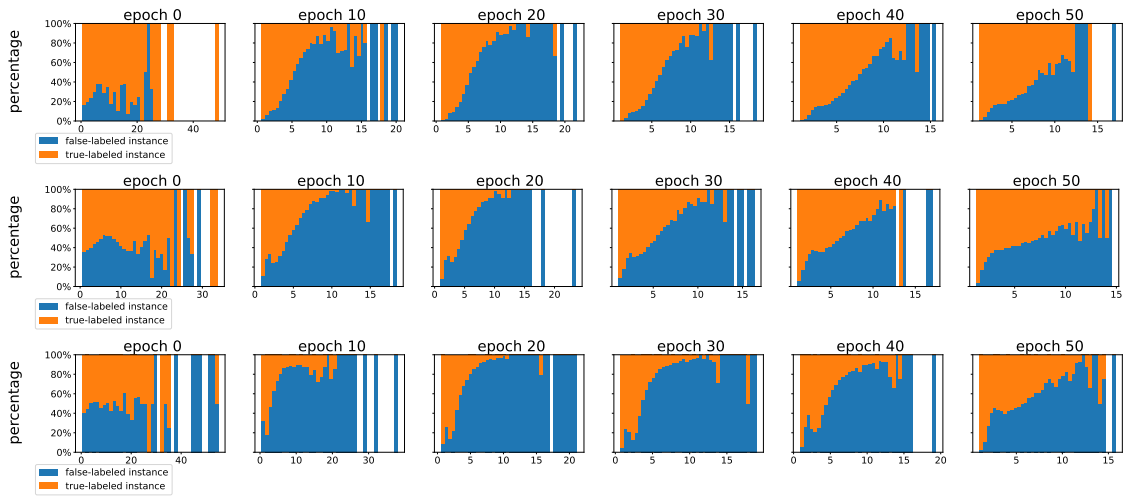


Figure 3.1: Distribution of LID scores for true-labeled versus false-labeled instances in three noise conditions. The heights of the orange and blue bars represent the proportions of true-labeled and false-labeled instances' LID scores within specific bins, respectively. LID scores are based on the enhanced representation of features and labels in the LID-dis. From top to bottom, the noise conditions for the three figures are: 20% instance-dependent noise, 40% instance-dependent noise, and 50% symmetric noise.

3.3.2 LID and Instance with Noisy Labels

In this section, we outline the use of a specially designed classifier: LID-based noisy label discriminator (LID-dis) f_{LD}, that employs LID as a feature to identify samples with incorrect labels. Prior research [87] has leveraged the LID scores from the final layer of a trained DNN classifier to characterize adversarial samples. However, in our context, the noise is present in the labels, not in the features. To ensure that f_{LD} can detect this noise, we input both the features and label into f_{LD}. f_{LD} consists of three components: a standard backbone model g_{LD} (which accepts x_i as input), a label embedding layer g_{LE} that processes the label $\tilde{y}_i$, and a classification head h_{LD} that takes the outputs of g_{LD} and g_{LE} to produce the final classification. The output from the backbone model $g_{\text{LD}}(x_i)$ and the label's embedding $g_{\text{LE}}(\tilde{y}_i)$ are merged as follows:

$$z(x_i, \tilde{y}_i) = \text{LayerNorm}\left(g_{\text{LD}}(x_i) + g_{\text{LE}}\left(\tilde{y}_i\right)\right) \tag{3.3.1}$$

The result, $z(x_i, \tilde{y}_i)$, the *enhanced representation* of $(x_i, \tilde{y}_i)$, is then passed to the classification head h_{LD} to predict $\tilde{y}_i$: $\hat{y}_i^D = h_{\text{LD}}(z(x_i, \tilde{y}_i))$. Here, $\hat{y}_i^D$ represents the predicted value of $\tilde{y}_i$. If we train f_{LD} directly using the cross-entropy loss $\mathcal{L}_{\text{CE}}(\tilde{y}_i, \hat{y}_i^D) = -\sum_{j=1}^{N_C} \tilde{y}_{i,j} \log(\hat{y}_{i,j}^D)$, the model will consistently predict $\tilde{y}_i$. Using the noisy label as the "ground truth" label for measuring prediction accuracy would yield a 100% accuracy rate. This is because the model's predictions are solely based on the input label $\tilde{y}_i$. To compel the model to consider both the input features and label, we randomly assign a new label $\tilde{y}_i^*$ for each x_i, ensuring that $\tilde{y}_i^* \neq \tilde{y}_i$. We input the pair $(x_i, \tilde{y}_i^*)$ into f_{LD} to obtain another prediction $\hat{y}_i^{*D}$. We then employ the sum of the cross-entropy losses $\mathcal{L}_{\text{CE}}(\tilde{y}_i, \hat{y}_i^D) + \mathcal{L}_{\text{CE}}(\tilde{y}_i, \hat{y}_i^{*D})$ to train the network. This approach ensures that f_{LD} doesn't rely solely on the input label for predictions.

For LID calculation, we follow the method described in [89] (see Equation 1.5 in supplementary materials). Note that the input to f_{LD} is $(x_i, \tilde{y}_i)$. Assume that $(x_i, \tilde{y}_i) \in \tilde{D}_B, \tilde{D}_B \subset \tilde{D}$. Here, $\tilde{D}_B$ is the mini-batch drawn from $\tilde{D}$. Let $z(\tilde{D}_B) = \{z(x_i, \tilde{y}_i) | (x_i, \tilde{y}_i) \in \tilde{D}_B\}$. The equation to calculate LID score for $(x_i, \tilde{y}_i)$ can be presented below:

$$\widehat{\text{LID}}((x_i, \tilde{y}_i), \tilde{D}_B) = -\left(\frac{1}{k} \sum_{j=1}^{k} \log \frac{r_j(z(x_i, \tilde{y}_i), z(\tilde{D}_B))}{r_{\max}(z(x_i, \tilde{y}_i), z(\tilde{D}_B))}\right)^{-1}. \tag{3.3.2}$$

Here, the term $r_j(z(x_i, \tilde{y}_i), z(\tilde{D}_B))$ represents the distance of $z(x_i, \tilde{y}_i)$ to its j-th nearest neighbor in the set $\tilde{D}_B$, and $r_{\max}$ is the neighborhood's radius. Following the training procedure described above, in order to explore the properties of the LID score of $z(x_i, y_i)$ in f_{LD}, we conducted an empirical study on the CIFAR-10 dataset with three types of noise conditions: 20% instance-dependent noise, 40% instance-dependent noise, and 50% symmetric noise. We used ResNet-34 [90] as the backbone network g_{LD}. During the training procedure, we recorded the estimation of the LID score (computed by Equation 3.3.2) for each instance $(x_i, \tilde{y}_i)$ at every epoch. We then split these LID scores into equal length bins and visualized the percentage distribution of false-labeled instances ($\tilde{y}_i \neq y_i$) and true-labeled instances ($\tilde{y}_i = y_i$) in each bin in Figure 3.1. As shown in this figure, for all three

types of noise conditions, false-labeled instances tend to have higher LID scores compared to true-labeled instances. This observation underscores that, across various noise conditions, LID scores from LID-dis serve as a robust metric to differentiate between true-labeled and false-labeled instances.

3.3.3 Proposed Method: CoLafier

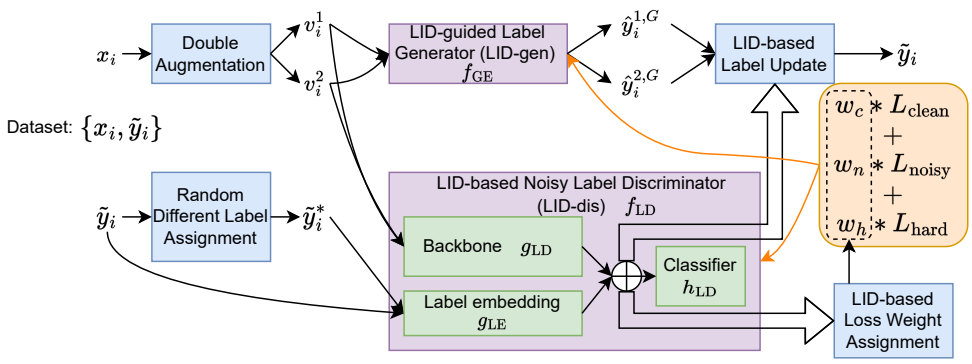


Figure 3.2: The overall framework of CoLafier.

Informed by these observations, we introduce a collaborative framework, CoLafier: Collaborative Noisy Label purifier with LID guidance, tailored for learning with noisy labels. The comprehensive structure of CoLafier is depicted in Figure 3.2, and the pseudo-code of CoLafier is presented in the supplementary materials. This framework is bifurcated into two primary subnets: LID-dis f_{LD} and LID-guided label generator (LID-gen) f_{GE}. Specifically, f_{GE} operates as a conventional classification model, predicting $\hat{y}_i$ based on x_i. The training regimen of CoLafier unfolds in four distinct phases:

1. **Pre-processing:** For an instance $(x_i, \tilde{y}_i)$ drawn from batch $\tilde{D}_B$, we employ double augmentation to generate two distinct views: v_i^1 and v_i^2. Subsequently, a new label $\tilde{y}_i^*$ is assigned, ensuring it differs from $\tilde{y}_i$.

2. **Prediction and LID Calculation:** Post inputting the features and (features, label) pairs into LID-dis and LID-gen, predictions are derived from both subnets. Let's denote

the predictions from f_{GE} as $\hat{y}_i^{1,G}$ and $\hat{y}_i^{2,G}$. Concurrently, CoLafier computes the LID scores for both $(v_i^1, \tilde{y}_i)$ and $(v_i^2, \tilde{y}_i)$.

3. **Loss Weight Assignment:** Utilizing the two LID scores from last step, CoLafier allocates weights to each instance. Every instance is endowed with three distinct weights: clean, noisy, hard weights, with each weight catering to a specific loss function.

4. **Label Update:** LID-dis processes $(v_i^1, \hat{y}_i^{1,G})$ and $(v_i^2, \hat{y}_i^{2,G})$, deriving LID scores for them. These scores and the difference between prediction from f_{LD} and f_{GE} subsequently guide the decision on whether to substitute $\tilde{y}_i$ with a combination of $\hat{y}_i^{1,G}$ and $\hat{y}_i^{2,G}$ for future epochs.

Pre-processing. Consider a mini-batch $\tilde{D}_B = \{(x_i, \tilde{y}_i)\}_{i=1}^{N_B}$ drawn from $\tilde{D}$. This batch can be partitioned into a feature set $X_B = \{x_i | (x_i, \tilde{y}_i) \in \tilde{D}_B\}$ and a label set $\tilde{Y}_B = \{\tilde{y}_i | (x_i, \tilde{y}_i) \in \tilde{D}_B\}$. For each $x_i \in X_B$, CoLafier generates two augmented views, v_i^1 and v_i^2. This two-augmentation design ensures that weight assignment and label update decisions in subsequent steps are not solely dependent on the original input. Such a design can lower the risk of error accumulation during the training procedure. The augmented views lead to: $V_B^1 = \{v_i^1 | v_i^1 = \text{augmentation1}(x_i), \forall x_i \in X_B\}, V_B^2 = \{v_i^2 | v_i^2 = \text{augmentation2}(x_i), \forall x_i \in X_B\}$. Same as Section 3.3.2, for each label $\tilde{y}_i$ in $\tilde{Y}_B$, a random new label $\tilde{y}_i^*$ is assigned, ensuring that $\tilde{y}_i^* \neq \tilde{y}_i$. The resulting set is given by: $\tilde{Y}_B^* = \{\tilde{y}_i^* | \tilde{y}_i^* = \text{assignNewLabel}(\tilde{y}_i), \tilde{y}_i \in \tilde{Y}_B\}$. Consequently, we can define four input pair sets: $\tilde{D}_B^k = \{(v_i^k, \tilde{y}_i) | v_i^k \in V_B^k, \tilde{y}_i \in \tilde{Y}_B\}, \tilde{D}_B^{k*} = \{(v_i^k, \tilde{y}_i^*) | v_i^k \in V_B^k, \tilde{y}_i^* \in \tilde{Y}_B^*\}$, where $k \in \{1, 2\}$. The LID-gen subnet f_{GE} processes V_B^1 and V_B^2, while the LID-dis subnet f_{LD} handles $\tilde{D}_B^1, \tilde{D}_B^2, \tilde{D}_B^{1*}$, and $\tilde{D}_B^{2*}$.

Prediction and LID Calculation. The subnet f_{GE} takes V_B^1 and V_B^2 as inputs to predict: $\hat{Y}_B^{k,G} = \{\hat{y}_i^{k,G} | \hat{y}_i^{k,G} = f_{\text{GE}}(v_i^k), v_i^k \in V_B^k\}$, where $k \in \{1, 2\}$. The subnet f_{LD} processes $\tilde{D}_B^1, \tilde{D}_B^2, \tilde{D}_B^{1*}$, and $\tilde{D}_B^{2*}$ to predict: $\hat{Y}_B^{k,D} = \{\hat{y}_i^{k,D} | \hat{y}_i^{k,D} = f_{\text{LD}}(v_i^k, \tilde{y}_i), (v_i^k, \tilde{y}_i) \in \tilde{D}_B^k\}$, $\hat{Y}_B^{k*,D} = \{\hat{y}_i^{k*,D} | \hat{y}_i^{k*,D} = f_{\text{LD}}(v_i^k, \tilde{y}_i^*), (v_i^k, \tilde{y}_i^*) \in \tilde{D}_B^{k*}\}$, where $k \in \{1, 2\}$. In f_{LD}, each input

pair result in an enhanced representations, we use Equation 3.3.2 to calculate LID scores for instances in $\tilde{D}_B^1$ and $\tilde{D}_B^2$:

$$\widehat{\mathrm{LID}}^W(v_i^k, \tilde{y}_i) = \widehat{\mathrm{LID}}((v_i^k, \tilde{y}_i), \tilde{D}_B^k), \tag{3.3.3}$$

$$\widehat{\mathrm{LID}}^W(\tilde{D}_B^k) = \{\widehat{\mathrm{LID}}^W\left(v_i^k, \tilde{y}_i\right) \mid \left(v_i^k, \tilde{y}_i\right) \in \tilde{D}_B^k\}, \tag{3.3.4}$$

where $k \in \{1, 2\}$. These LID scores are for the weight assignment use only. After we obtain prediction $\hat{Y}_B^{1,G}$ and $\hat{Y}_B^{2,G}$ from f_{GE}, we create another two input pair sets: $\hat{D}_B^k = \{(v_i^k, \hat{y}_i^{k,G}) | v_i^k \in V_B^k, \hat{y}_i^{k,G} \in \hat{Y}_B^{k,G}\}$, where $k \in \{1, 2\}$. Both $\hat{D}_B^1$ and $\hat{D}_B^2$ are fed into the f_{LD} to obtain enhanced representations. Because we want to compare the LID scores from current noisy label and f_{GE}'s prediction to determine if we want to update the label, we create two union sets, then calculate LID scores within the two sets as follows:

$$U_B^k = \tilde{D}_B^k \cup \hat{D}_B^k, \tag{3.3.5}$$

$$\widehat{\mathrm{LID}}^U(v_i^k, \tilde{y}_i^k) = \widehat{\mathrm{LID}}((v_i^k, \tilde{y}_i^k), U_B^k), \tag{3.3.6}$$

$$\widehat{\mathrm{LID}}^U(v_i^k, \hat{y}_i^{k,G}) = \widehat{\mathrm{LID}}((v_i^k, \hat{y}_i^{k,G}), U_B^k), \tag{3.3.7}$$

$$\begin{aligned}\widehat{\mathrm{LID}}^U(U_B^k) = \{&\widehat{\mathrm{LID}}^U(v_i^k, \tilde{y}_i^k) | (v_i^k, \tilde{y}_i^k) \in \tilde{D}_B^k\} \cup \\ &\{\widehat{\mathrm{LID}}^U(v_i^k, \hat{y}_i^{k,G}) | (v_i^k, \hat{y}_i^{k,G}) \in \hat{D}_B^k\},\end{aligned} \tag{3.3.8}$$

where $k \in \{1, 2\}$. We also collect the output from f_{LD}: $\hat{Y}_B^{k,G,D} = \{\hat{y}_i^{k,G,D} | \hat{y}_i^{k,G,D} = f_{\mathrm{LD}}(v_i^k, \hat{y}_i^{k,G}), (v_i^k, \hat{y}_i^{k,G}) \in \hat{D}_B^k\}$, where $k \in \{1, 2\}$. Note that $\hat{Y}_B^{1,G,D}$ and $\hat{Y}_B^{2,G,D}$ are only used in label update step and do not participate in loss calculation.

Loss Weight Assignment. After estimating the LID scores, we compute weights for each instance. We introduce three types of weights: clean, hard, and noisy. Each type of weight is associated with specific designed loss function. A higher clean weight indicates that the instance is more likely to be a true-labeled instance, while a higher noisy weight suggests the

opposite. A high hard weight indicates uncertainty in labeling. As observed in Section 3.3.2, instances with smaller LID scores tend to be correctly labeled. Thus, we assign higher clean weights to instances with lower LID scores, higher noisy weights to instances with higher LID scores, and higher hard weights to instances with significant discrepancies in LID scores from two views. To mitigate potential biases in weight assignment, we prefer using $\widehat{\text{LID}}^W$ over $\widehat{\text{LID}}^U$. This preference is due to the observation that the prediction from f_{GE} that are identical to the label could lower the label's $\widehat{\text{LID}}^U$ score. Such a decrease does not necessarily indicate the correctness of a label and hence could skew the weight assignment. The weights are defined as:

$$q_{\text{low}}^{k,W} = \text{quantile}(\widehat{\text{LID}}^W(\tilde{D}_B^k), \epsilon_{\text{low}}^W), \tag{3.3.9}$$

$$q_{\text{high}}^{k,W} = \text{quantile}(\widehat{\text{LID}}^W(\tilde{D}_B^k), \epsilon_{\text{high}}^W), \tag{3.3.10}$$

$$q_i^{k,W} = \frac{q_{\text{high}}^{k,W} - \widehat{\text{LID}}^W(v_i^k, \tilde{y}_i)}{q_{\text{high}}^{k,W} - q_{\text{low}}^{k,W}}, \tag{3.3.11}$$

$$w_{i,k} = \min\left(\max\left(q_i^{k,W}, 0\right), 1\right), \tag{3.3.12}$$

$$w_{i,c} = \min\left(w_{i,1}, w_{i,2}\right), \tag{3.3.13}$$

$$w_{i,h} = |w_{i,1} - w_{i,2}|, \tag{3.3.14}$$

$$w_{i,n} = \min\left(1 - w_{i,1}, 1 - w_{i,2}\right), \tag{3.3.15}$$

where $k \in \{1, 2\}$. Here, $w_{i,c}$, $w_{i,h}$, and $w_{i,n}$ represent clean, hard, and noisy weights, respectively. It's ensured that the sum of $w_{i,c}$, $w_{i,h}$, and $w_{i,n}$ equals 1, which is proved in the supplementary materials. The thresholds ϵ_{low}^W and ϵ_{high}^W are predefined, satisfying $0 \leq \epsilon_{\text{low}}^W \leq \epsilon_{\text{high}}^W \leq 1$. Initially, the value of ϵ_{high}^W is set low and is then linearly increased over τ epochs. This approach ensures that the model does not prematurely allocate a large number of high clean weights, given that the majority of labels have not been refined in the early stages. Only instances with low LID scores are predominantly correctly labeled. For instances with a high clean weight, we employ the cross-entropy loss for optimization. The

clean loss is defined as:

$$\mathcal{L}_{\text{clean,GE}} = w_{i,c} \sum_{k=1}^{2} \mathcal{L}_{\text{CE}} \left(\tilde{y}_i, \hat{y}_i^{k,G} \right), \tag{3.3.16}$$

$$\mathcal{L}_{\text{clean,LD}} = w_{i,c} \sum_{k=1}^{2} \left(\mathcal{L}_{\text{CE}} \left(\tilde{y}_i, \hat{y}_i^{k,D} \right) + \lambda^* \mathcal{L}_{\text{CE}} \left(\tilde{y}_i, \hat{y}_i^{k*,D} \right) \right). \tag{3.3.17}$$

Instances with a high $w_{i,h}$ indicate a significant discrepancy between $\widehat{\text{LID}}(v_i^1, \tilde{y}_i)$ and $\widehat{\text{LID}}(v_i^2, \tilde{y}_i)$. This suggests that these instances might be near the decision boundary. While we aim to utilize these instances, the cross-entropy loss is sensitive to label noise. Therefore, we adopt a more robust loss function, the generalized cross entropy (GCE) [91], defined as:

$$\mathcal{L}_{\text{GCE}} \left(\tilde{y}_i, \hat{y}_i \right) = \textstyle\sum_{j=1}^{N_C} \tilde{y}_{i,j} \left(1 - (\hat{y}_{i,j})^q \right) / q, \tag{3.3.18}$$

where $q \in (0, 1]$. As shown in [91], this loss function approaches the cross-entropy loss as $q \to 0$ and becomes the MAE loss when $q = 1$. We set $q = 0.7$ as recommended by [91]. The hard loss is then:

$$\mathcal{L}_{\text{hard,GE}} = w_{i,h} \sum_{k=1}^{2} \mathcal{L}_{\text{GCE}} \left(\tilde{y}_i, \hat{y}_i^{k,G} \right), \tag{3.3.19}$$

$$\mathcal{L}_{\text{hard,LD}} = w_{i,h} \sum_{k=1}^{2} \left(\mathcal{L}_{\text{GCE}} \left(\tilde{y}_i, \hat{y}_i^{k,D} \right) + \lambda^* \mathcal{L}_{\text{GCE}} \left(\tilde{y}_i, \hat{y}_i^{k*,D} \right) \right). \tag{3.3.20}$$

Instances with a high $w_{i,n}$ are likely to be mislabeled. To leverage these instances without being influenced by label noise, we adopt the CutMix augmentation strategy [92]. In essence, CutMix combines two training samples by cutting out a rectangle from one and placing it onto the other [1]. For a detailed explanation and methodology of CutMix, readers are referred to [92]. We apply CutMix twice for each sample within $\tilde{D}_B^1$ and $\tilde{D}_B^2$. The

[1] In this work, we use images as input. While CutMix was designed for images, it hasn't been widely applied to other types of input. For non-image data, other augmentation methods like Mixup [93] can be considered.

augmented views are defined as:

$$\check{v}_i^k = \mathbf{M}^k v_i^k + (1 - \mathbf{M}^k) v_{r_k(i)}^k, \quad k \in \{1, 2\}. \tag{3.3.21}$$

$$\check{y}_i^k = \lambda_k \tilde{y}_i + (1 - \lambda_k) \tilde{y}_{r_k(i)} \quad k \in \{1, 2\}. \tag{3.3.22}$$

Here, $r_1(i)$ and $r_2(i)$ are random indices for the two views, and $\mathbf{M}^k$ is a binary mask indicating the regions to combine. The factors λ_1 and λ_2 are sampled from the beta distribution $Beta(\alpha, \alpha)$, with $\alpha = 1$ as suggested by [92]. The proportion of the combination is determined by the λ term. Specifically, λ represents the ratio of the original view retained, while $1 - \lambda$ denotes the proportion of the other view that's patched in. The CutMix views $\check{v}_i^1$ and $\check{v}_i^2$ are then fed into f_{GE} to obtain predictions $\hat{y}_i^{1,G}$ and $\hat{y}_i^{2,G}$. Similarly, $(\check{v}_i^1, \check{y}_i^1)$ and $(\check{v}_i^2, \check{y}_i^2)$ are processed by f_{LD} to get $\hat{y}_i^{1,D}$ and $\hat{y}_i^{2,D}$.

The loss for CutMix instances is defined as:

$$\mathcal{L}'_{\text{noisy,GE}} = \sum_{k=1}^{2} \left[\lambda_k \mathcal{L}_{\text{CE}} \left(\tilde{y}_i, \hat{y}_i^{k,G} \right) + (1 - \lambda_k) \mathcal{L}_{\text{CE}} \left(\tilde{y}_{r_k(i)}, \hat{y}^{k,G} \right) \right], \tag{3.3.23}$$

$$\mathcal{L}'_{\text{noisy,LD}} = \sum_{k=1}^{2} \left[\lambda_k \mathcal{L}_{\text{CE}} \left(\tilde{y}_i, \hat{y}_i^{k,D} \right) + (1 - \lambda_k) \mathcal{L}_{\text{CE}} \left(\tilde{y}_{r_k(i)}, \hat{y}^{k,D} \right) \right]. \tag{3.3.24}$$

To enhance the learning from instances with high noise weights in f_{LD}, we also employ a consistency loss. This loss, based on cosine similarity, ensures consistent predictions between $\tilde{y}_i$ and $\tilde{y}_i^*$ without relying on label guidance. The consistency loss for f_{LD} is:

$$\mathcal{L}_{\text{cons,LD}} = \sum_{k=1}^{2} \left(1 - \cos \left(\hat{y}_i^{k,D}, \hat{y}_i^{k*,D} \right) \right). \tag{3.3.25}$$

We combine the consistency loss and CutMix loss to get the noisy loss as:

$$\mathcal{L}_{\text{noisy,GE}} = w_{i,n} \mathcal{L}'_{\text{noisy,GE}}, \tag{3.3.26}$$

$$\mathcal{L}_{\text{noisy,LD}} = w_{i,n} (\mathcal{L}'_{\text{noisy,LD}} + \lambda_{\text{cons}} \mathcal{L}_{\text{cons,LD}}). \tag{3.3.27}$$

The overall training objectives for f_{GE} and f_{LD} combine the clean, hard, and noisy losses:

$$\mathcal{L}_{\text{GE}} = \mathcal{L}_{\text{clean,GE}} + \mathcal{L}_{\text{hard,GE}} + \mathcal{L}_{\text{noisy,GE}}, \tag{3.3.28}$$

$$\mathcal{L}_{\text{LD}} = \mathcal{L}_{\text{clean,LD}} + \mathcal{L}_{\text{hard,LD}} + \mathcal{L}_{\text{noisy,LD}}. \tag{3.3.29}$$

Both f_{GE} and f_{LD} are optimized separately using their respective loss functions.

Label Update. For determining whether to update the label based on the prediction from f_{GE}, we consider both the LID scores and the prediction difference between f_{GE} and f_{LD}. As discussed in Section 3.3.2, instances with smaller LID scores are more likely to be correctly labeled. If the LID scores associated with f_{GE}'s prediction are smaller than the current label's scores, then the prediction is more likely to be accurate. The prediction difference is defined as:

$$\Delta\tilde{y}_i^k = \textstyle\sum_{j=1}^{N_c} |\hat{y}_{i,j}^{k,G} - \hat{y}_{i,j}^{k,D}|, \quad k \in \{1,2\}. \tag{3.3.30}$$

$$\Delta\hat{y}_i^k = \textstyle\sum_{j=1}^{N_c} |\hat{y}_{i,j}^{k,G} - \hat{y}_{i,j}^{k,G,D}|, \quad k \in \{1,2\}. \tag{3.3.31}$$

The principle of agreement maximization suggests that different models are less likely to agree on incorrect labels [33]. The Δ value measures the level of disagreement between f_{GE} and f_{LD}. A larger Δ value indicates that the corresponding prediction or label is less likely to be correct. Generally, if a prediction has a smaller LID score and a smaller Δ compared to the current label, it's a candidate for label replacement. Using the LID scores computed in Section 3.3.3, CoLafier make decision on label updating as follows:

$$q_{\text{low}}^{k,U} = \text{quantile}(\widehat{\text{LID}}^U(U_B^k), \epsilon_{\text{low}}^U), \tag{3.3.32}$$

$$q_{\text{high}}^{k,U} = \text{quantile}(\widehat{\text{LID}}^U(U_B^k), \epsilon_{\text{high}}^U), \tag{3.3.33}$$

$$\tilde{q}_i^{k,U} = (q_{\text{high}}^{k,U} - \widehat{\text{LID}}^U(v_i^k, \tilde{y}_i))/(q_{\text{high}}^{k,U} - q_{\text{low}}^{k,U}), \tag{3.3.34}$$

$$\hat{q}_i^{k,U} = (q_{\text{high}}^{k,U} - \widehat{\text{LID}}^U(v_i^k, \hat{y}_i^{k,G}))/(q_{\text{high}}^{k,U} - q_{\text{low}}^{k,U}), \tag{3.3.35}$$

$$\tilde{t}_i^k = \min\left(1, \max\left(0, \tilde{q}_i^{k,U} * (2 - \Delta\tilde{y}_i^k)/2\right)\right), \tag{3.3.36}$$

$$\hat{t}_i^k = \min\left(1, \max\left(0, \hat{q}_i^{k,U} * (2 - \Delta\hat{y}_i^k)/2\right)\right), \tag{3.3.37}$$

where $k \in \{1, 2\}$, ϵ_{low}^U and ϵ_{high}^U are thresholds. Mirroring the approach of ϵ_{high}^W, ϵ_{high}^U starts low and linearly rises over τ epochs, enabling the model to judiciously assess the reliability of labels and predictions. The Δ values are normalized to the [0,1] range using $(2 - \Delta)/2$, which is elaborated in the supplementary materials. In these equations, predictions or labels with smaller LID values and smaller cross-subnet differences have larger t values, and vice versa. The process of converting both $\hat{y}_i^{1,G}$ and $\hat{y}_i^{2,G}$ to one-hot label vectors is as follows:

$$\acute{y}_i = [\acute{y}_{i,0}, \acute{y}_{i,1}, ...\acute{y}_{i,N_c}] \tag{3.3.38}$$

$$\acute{y}_{i,l} = \begin{cases} 1, & \text{if } l = \underset{j}{\text{argmax}}(\hat{y}_{i,j}) \\ 0, & \text{otherwise} \end{cases}$$

We determine whether to update the label as follows:

$$\begin{aligned} \text{cond} = \hat{t}_i^1 > \tilde{t}_i^1 \ \ \& \ \ \hat{t}_i^2 > \tilde{t}_i^2 \\ \& \ \ \hat{t}_i^1 > \epsilon_k \ \ \& \ \ \hat{t}_i^2 > \epsilon_k \ \ \& \ \ \acute{y}_i^{1,G} = \acute{y}_i^{2,G} \end{aligned} \tag{3.3.39}$$

$$\tilde{y}_i = \begin{cases} \acute{y}_i^{1,G}, & \text{if cond} \\ \tilde{y}_i, & \text{otherwise} \end{cases} \tag{3.3.40}$$

In this decision-making process, a label is only updated when the prediction's t value from both views surpasses a predefined threshold ϵ_k and is higher than higher than the t value of the corresponding label. Both predictions $\acute{y}_i^{1,G}$ and $\acute{y}_i^{2,G}$ must also be equal, minimizing the chance of assigning an incorrect label to instance x_i. Note that the new $\tilde{y}_i$ is used for

the next epoch, ensuring that in current epoch, the loss calculation in Section 3.3.3 is not affected by the label update.

3.4 Experiments

In this section, we assess the performance of CoLafier across various noisy label settings. We also present ablation studies to validate the contribution of each component. All experiments are executed using A100 GPUs and PyTorch 1.13.1.

3.4.1 Experiment Setup

CoLafier is evaluated on CIFAR-10 [94] with three type of noise: symmetric (sym.), asymmetric (asym.) and instance-dependent (inst.) noise. Sym. noise involves uniformly flipping labels at random, while asym. noise flips labels between neighboring classes at a fixed probability, following methods in [32, 36]. Inst. noise is generated per instance using a truncated Gaussian distribution as per [35, 95]. Noise ratios are set at {20%, 50%, 80%} for sym. noise, 40% for asym. noise, and {20%, 40%, 60%} for inst. noise, aligning with settings in [96, 35]. Additionally, experiments are conducted on CIFAR-10N [97], a real-world noisy dataset with re-annotated CIFAR-10 images. CIFAR-10N provides three submitted labels (i.e., Random 1, 2, 3) per image, aggregated to create an Aggregate and a Worst label. The Aggregate, Random 1, and Worst label are used in experiment. ResNet-18 [90] serves as the backbone for CIFAR-10 with sym. and asym. noise, while ResNet-34 is used for CIFAR-10 with inst. noise and CIFAR-10N. Additional details are provided in the supplementary materials.

Table 3.1: Performance comparison against SOTA methods on the CIFAR-10 dataset under symmetric and asymmetric noise. Our implementations are marked with *; others are from [35]. **Bold** scores are the highest, and underlined scores the second highest in each setting.

Methods	Sym. 20%	Sym. 50%	Sym. 80%	Asym. 40%	Average
Cross Entropy	83.31 ± 0.09	56.41 ± 0.32	18.52 ± 0.16	77.06 ± 0.26	58.83
Mixup [93]	90.17 ± 0.12	70.94 ± 0.26	47.15 ± 0.37	82.68 ± 0.38	72.74
Decoupling [98]	85.40 ± 0.12	68.57 ± 0.34	41.08 ± 0.24	78.67 ± 0.81	68.43
Co-teaching [32]	87.95 ± 0.07	48.60 ± 0.19	17.48 ± 0.11	71.14 ± 0.32	56.29
JointOptim [99]	91.34 ± 0.40	89.28 ± 0.74	59.67 ± 0.27	90.63 ± 0.39	82.73
Co-teaching+[100]	87.20 ± 0.08	54.24 ± 0.23	22.26 ± 0.55	79.91 ± 0.46	60.90
GCE [91]	90.05 ± 0.10	79.40 ± 0.20	20.67 ± 0.11	74.73 ± 0.39	66.21
PENCIL [101]	88.02 ± 0.90	70.44 ± 1.09	23.20 ± 0.81	76.91 ± 0.26	64.64
JoCoR [33]	89.46 ± 0.04	54.33 ± 0.12	18.31 ± 0.11	70.98 ± 0.21	58.27
DivideMix* [36]	92.87 ± 0.46	94.75 ± 0.14	81.25 ± 0.26	91.88 ± 0.12	90.19
ELR [102]	90.35 ± 0.04	87.40 ± 3.86	55.69 ± 1.00	89.77 ± 0.12	80.80
ELR+ [102]	95.27 ± 0.11	94.41 ± 0.11	81.86 ± 0.23	91.38 ± 0.50	90.73
Co-learning [103]	92.14 ± 0.09	77.99 ± 0.65	43.80 ± 0.76	82.70 ± 0.40	74.16
GJS* [104]	83.57 ± 1.24	50.26 ± 2.54	15.49 ± 0.18	85.64 ± 1.37	58.74
DISC* [35]	**95.99 ± 0.15**	**95.03 ± 0.12**	81.84 ± 0.21	94.20 ± 0.07	91.69
CoLafier (ours)	95.32 ± 0.08	93.64 ± 0.11	**84.42 ± 0.20**	**94.67 ± 0.11**	**92.01**

3.4.2 Experiment Results

Table 3.1 shows that CoLafier consistently ranks among the top three in prediction accuracy on the CIFAR-10 dataset under both sym. and asym. noise conditions, achieving the highest average accuracy across four scenarios. Its robustness to various noise ratios and types stands out. In Table 3.2, CoLafier achieves the highest average accuracy under inst. noise, notably excelling at an 80% noise ratio. Table 3.3 further demonstrates CoLafier's superior performance and robustness under real-world noise settings, particularly under high noise conditions (Worst, 40% noise). These findings underscore the robustness and superior generalization capability of CoLafier.

Table 3.2: Performance comparison against SOTA methods on the CIFAR-10 dataset under instance dependent noise. Our implementations are marked with *; others are from [35]. **Bold** scores are the highest, and underlined scores the second highest in each setting.

Methods	Inst. 20%	Inst. 40%	Inst. 60%	Average
Cross Entropy	83.93 ± 0.15	67.64 ± 0.26	43.83 ± 0.33	65.13
Forward T [105]	87.22 ± 1.60	79.37 ± 2.72	66.56 ± 4.90	77.72
DMI [106]	88.57 ± 0.60	82.82 ± 1.49	69.94 ± 1.34	80.44
Mixup [93]	87.71 ± 0.66	82.65 ± 0.38	58.59 ± 0.58	76.32
GCE [91]	89.80 ± 0.12	78.95 ± 0.15	60.76 ± 3.08	76.50
Co-teaching [32]	88.87 ± 0.24	73.00 ± 1.24	62.51 ± 1.98	74.79
Co-teaching+ [100]	89.80 ± 0.28	73.78 ± 1.39	59.22 ± 6.34	74.27
JoCoR [33]	88.78 ± 0.15	71.64 ± 3.09	63.46 ± 1.58	74.63
Reweight-R [107]	90.04 ± 0.46	84.11 ± 2.47	72.18 ± 2.47	82.11
Peer Loss [108]	89.12 ± 0.76	83.26 ± 0.42	74.53 ± 1.22	82.30
DivideMix* [36]	92.95 ± 0.29	94.99 ± 0.14	89.30 ± 1.32	92.41
CORSES^2 [109]	91.14 ± 0.46	83.67 ± 1.29	77.68 ± 2.24	84.16
CAL [96]	92.01 ± 0.12	84.96 ± 1.25	79.82 ± 2.56	85.60
DISC* [35]	**96.34 ± 0.13**	**95.27 ± 0.21**	91.15 ± 2.20	94.25
CoLafier (ours)	95.73 ± 0.10	94.66 ± 0.11	**92.45 ± 1.25**	**94.28**

Table 3.3: Performance comparison on CIFAR-10N. Our implementations are marked with *; others are from [110]. **Bold** scores are the highest, and underlined scores the second highest in each setting.

Methods	Aggregate	Random	Worst	Average
Cross Entropy	87.77 ± 0.38	85.02 ± 0.65	77.69 ± 1.55	83.49
Forward T [105]	88.24 ± 0.22	86.88 ± 0.50	79.79 ± 0.46	84.97
Co-teaching [32]	91.20 ± 0.13	90.33 ± 0.13	83.83 ± 0.13	88.45
ELR+ [102]	94.83 ± 0.10	94.43 ± 0.41	91.09 ± 1.60	93.45
CORES^2 [109]	95.25 ± 0.09	94.45 ± 0.14	91.66 ± 0.09	93.79
DISC* [35]	**95.96 ± 0.04**	**95.33 ± 0.12**	90.20 ± 0.24	93.83
CoLafier (ours)	95.74 ± 0.14	95.21 ± 0.27	**92.65 ± 0.10**	**94.53**

3.4.3 Ablation Study

In our ablation study, we examine four CoLafier variants to assess the impact of the dual-view design and the three loss types. Table 3.4 presents these variants: the first uses only

the original features for weight assignment and label updates; the second to fourth exclude both noise & hard loss, noise loss only, or, hard loss only, respectively. All these variants still use LID scores to assign weight and determine label updates. Results show that the complete CoLafier model outperforms its variants. The performance gap between the single-view variant and CoLafier highlights the dual-view approach's role in reducing errors and enhancing model robustness. The lesser performance of the latter three variants compared to CoLafier confirms that combining all three loss types effectively utilizes information from both correctly and incorrectly labeled instances.

Table 3.4: Ablation study for two views and loss types.

Variations	CIFAR-10, 40% Inst. Noise
CoLafier w/o two views	84.31 ± 0.59
CoLafier w/o noise and hard loss	91.06 ± 0.22
CoLafier w/o noise loss	91.36 ± 0.34
CoLafier w/o hard loss	93.56 ± 0.25
CoLafier	**94.66 ± 0.11**

3.5 Conclusion

In this study, we present CoLafier, a novel framework designed for learning with noisy labels. It is composed of two key subnets: LID-based noisy label discriminator (LID-dis) and LID-guided label generator (LID-gen). Both two subnets leverage two augmented views of features for each instance. The LID-dis assimilates features and labels of training samples to create enhanced representations. CoLafier employs LID scores from enhanced representations to weight the loss function for both subnets. LID-gen suggests pseudo-labels, and LID-dis process pseudo-labels along with two views to derive LID scores. These LID scores and the discrepancies in prediction from the two subnets inform the label correction decisions. This dual-view and dual-subnet approach significantly reduces the risk of errors and enhances the

overall effectiveness of the framework. After training, LID-gen is ready to be deployed as the classifier. Extensive evaluations demonstrate CoLafier's superiority over existing state of the arts in various noise settings, notably improving prediction accuracy.

Chapter 4

LLM-based Two-Level Foodborne Illness Detection Label Annotation with Limited Labeled Samples

This work will be submitted to top quality conference, with me serving as the lead author, alongside Ruofan Hu and Professor Elke Rundensteiner. My contributions include the development of the labeling framework, and the execution of the experiments. Below is an abridged version of this work.

4.1 Motivation

Foodborne illnesses constitute a significant public health threat, impacting millions of Americans annually. They lead to productivity loss, elevated medical expenses, and in some cases, fatalities [111, 112, 113]. Early detection of foodborne illnesses is crucial for risk mitigation, outbreak control, and ensuring public health, while conventional approaches of collecting data from official sources such as hospitals or CDC reporting systems, while more

controlled, can take precious time [114]. Consumer-generated data, spanning social media to internet searches, has emerged as a valuable resource for surveillance, paving the way for the development of surveillance tools grounded in conventional supervised machine learning. Local health agencies have tested these tools utilizing Twitter (current name is X) data in cities like New York [115], Chicago [116], and Las Vegas [117], Yelp reviews in San Francisco [118] and New York [119], as well as Google search queries in Las Vegas and Chicago [120].

Machine Learning or Deep Learning models are typically employed to detect foodborne illness incidents within social media posts [114, 117], in the aforementioned surveillance systems. An effective method should not only ascertain if a given post signifies a potential foodborne illness incident but also automatically extract relevant attributes from the post for aggregation into actionable insights. This post examination task bifurcates into two levels: at the post level, the objective is to predict whether the post indicates a foodborne illness incident, whereas at the word level, the aim is to identify mentions of slots, aka entities (e.g., food, symptom, location, time) related to the noted foodborne illness incident. These entities can be critical for uncovering for instance where the treat may have originated in terms of food type or location, as well as its likely spread.

Nonetheless, supervised models necessitate high-quality labeled training data for assuring accurate results. This in turn requires access to experts with domain knowledge to provide these labels. However, this approach is exceedingly resource-intensive and often prohibitively expensive to gather [5]. In our problem setting, a post has to be labeled on both the post and word level, requiring word-by-word analysis, which is more demanding and time-intensive compare to the traditional annotation of a single label for a possibly complex object (such as our social media posts). Budget constraints often hinder the collection of an adequate number of labels, leaving substantial data unlabeled.

Other than collecting label from expert, crowdsourcing platform could be a alternative way. In our prior study [4], we created a tweet dataset, TWEET-FID, for foodborne illness

incident detection. We collected labels from crowdsourced workers, to investigate the crowdsourced labeling quality, we compared these with labels from domain experts, with the latter serving as the ground truth, as detailed in Chapter 1.1.1. Throughout the data collection process, we gathered annotations from five workers per tweet and employed aggregation algorithms to synthesize a single annotation (including both tweet and word-level label) for each tweet. Despite deploying algorithms to filter out low-quality annotations, a notable quality gap between crowdsourced and expert labels remained, as shown in Table 4.2. The cost of crowdsourcing was significant, with each annotation costing $0.1, leading to at least $0.5 spent per tweet, underscoring the economic and quality challenges of this approach.

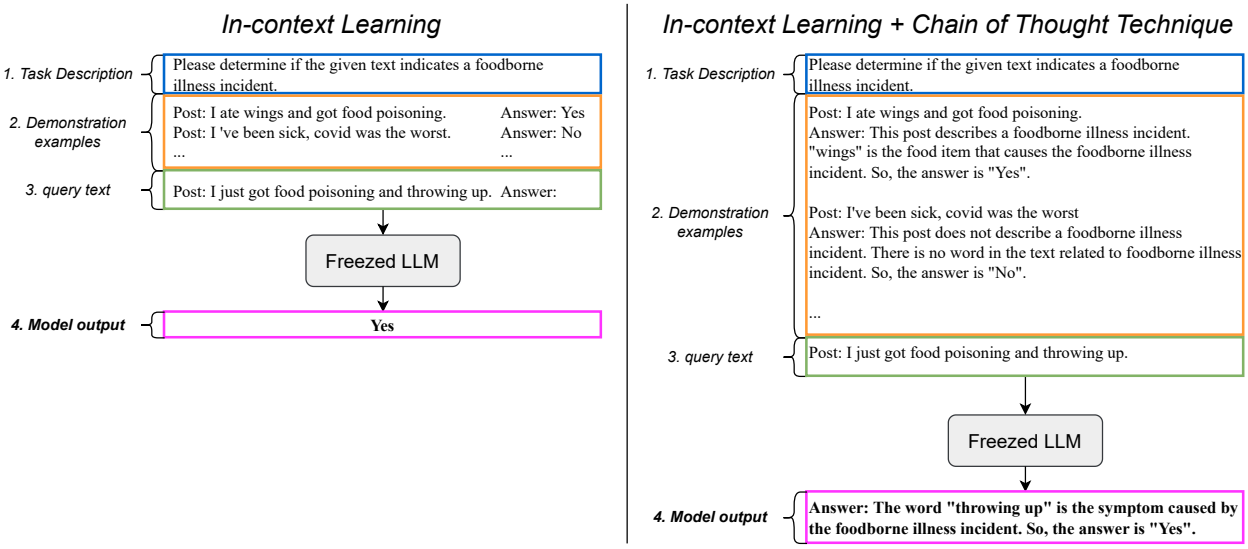


Figure 4.1: Illustration of In-Context Learning (ICL) and the Chain of Thought (CoT) technique. ICL utilizes a demonstration context with a task description and a few examples. Taking this demonstration context and a query text as the input, Large Language Models (LLMs) make predictions without updating parameters. The CoT technique introduces intermediate reasoning steps between inputs and outputs during the demonstration phase, guiding LLMs to reveal not only the final answer but also the underlying reasoning process.

Recently, LLMs have demonstrated a remarkable in-context learning (ICL) ability, enabling them to make predictions based on few examples within a specific demonstration context [11]. The left part of Figure 4.1 illustrates how ICL works. The demonstration context contains a task description and few demonstration examples. By feeding this demonstration context and a query text into LLMs, LLMs make predictions for the query text. This abil-

ity has been showcased across a variety of complex tasks, including solving math problems [45]. ICL operates without updating model parameters, directly leveraging the pretrained models for predictions. This ability is particularly advantageous in scenarios with limited resources. Further enhancing LLMs' efficacy, some studies [45, 121] have introduced the concept of chain-of-thoughts processes (CoT). As shown in the right part of Figure 4.1, this CoT approach involves adding intermediate reasoning steps between inputs and outputs in the demonstration phase, thereby guiding LLMs to predict not just the final answer but the underlying reasoning process as well. It is interesting in that reasoning steps to be exposed would allow us to check for the correctness of the logical processes underlying a response. Beyond this increase in interpretability, this has also been shown that the CoT method enhances LLM performance significantly.

Recently, prior work has utilized LLMs as annotators to generate labels for text [43, 42]. These studies have demonstrated the potential for LLMs in achieving text annotation and other related NLP tasks. Compared to human annotation, LLM generates good quality label at a lower cost [11]. However, we note that most of these works focus on the annotation task on a single level. To the best of our knowledge, there has yet to be research exploring the application of LLMs for annotating data at two or multiple levels of social media posts. The later is now the focus of our study.

Problem Definition. In this study, we propose to address the problem of annotating two-level labels for foodborne illness detection in social media posts utilizing a limited number of labeled samples.

Figure 4.2 illustrates a dataset of posts collected for the task of detecting foodborne illness incidents. The post level annotation determines whether the post signals a foodborne illness incident, while the word level annotations identify the entity type of each word that relates to the incident. Only a small subset of posts have been labeled by human annotators at both levels. Our objective is to explore the potential of LLMs for this annotation problem,

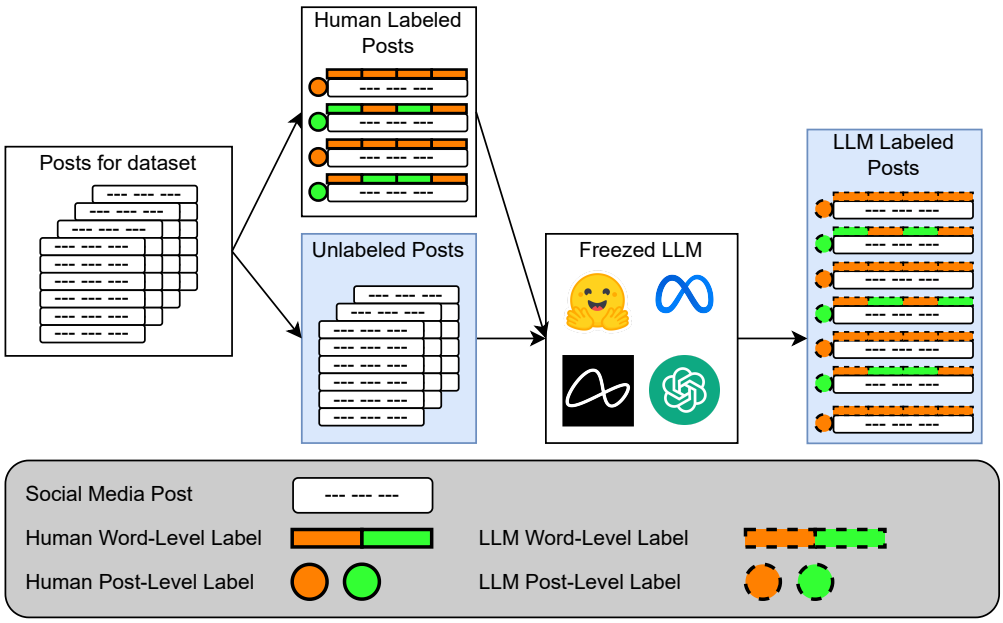


Figure 4.2: LLM-based Two-level Foodborne Illness Detection Label Annotation with limited human annotated samples. Given a social media post dataset for foodborne illness detection task, where limited number of posts have human labels for both two levels. Our goal is to develop a in context learning framework to assign labels for unlabeled posts with LLMs.

in comparison to traditional methods such as supervised learning models or crowdsourcing annotations. The focus is on developing a method based on in-context learning that leverages an LLM [1] to annotate unlabeled posts across both post and word levels.

Challenges. • *Interdependency between two levels.* This task requires annotations at both the post and word levels. It involves not only identifying whether a post suggests a foodborne illness incident but also extracting related entities. Importantly, an entity must be directly linked to a foodborne illness incident to be considered relevant. For example, consider a sentence like "Just watched a documentary on food safety, which really opened my eyes to the importance of avoiding food poisoning. #awareness." Although it mentions "food poisoning" and "food safety", these references are not connected to a specific incident. This necessitates that the LLM accurately assesses the relevance of such entities to an actual

[1]Throughout this study, we employ GPT-3.5-turbo [122] as our backbone LLM, and unless otherwise specified, references to LLMs pertain specifically to GPT-3.5-turbo. However, it is important to note that our labeling framework is designed to be adaptable and could be seamlessly applied to other advanced LLMs, including GPT-4 [123], Gemini [124], and Llama 2 [9].

foodborne illness incident. Additionally, for post-level labeling, the LLM must determine if any entities in the text suggest a foodborne illness incident. This complex task requires precise guidance for the LLM to comprehend and accurately perform the annotations.

• *Model hallucination in labeling procedure.* A well-known challenge with LLMs is their tendency to "hallucinate", often leading to discrepancies between the content generated by the LLMs and the ground truth [125, 12]. In our situation, most posts do not indicate foodborne illness incidents, and relevant entities are rare within the dataset [4]. However, the LLM might incorrectly label many irrelevant posts or words as related, resulting in numerous false positives. Thus, addressing model hallucination is essential in the labeling process, highlighting the importance of developing strategies to minimize these discrepancies and ensure the accuracy of LLM annotations.

• *Budget and token constraint.* Access to advanced, closed-source LLMs typically involves API usage that incurs a cost, calculated per token for both inputs and outputs[126]. Additionally, these APIs enforce a maximum token limit [123]. This limitation constrains the number of examples we can include in the demonstration context and restricts the length of outputs the model can generate. Therefore, it's imperative to design the labeling framework judiciously to secure high-quality annotations while managing token usage efficiently and maintaining costs at a viable level.

Proposed Method. To overcome these challenges, we propose a novel labeling framework, ICL2FID: In-context Learning based Annotation for Two-level Foodborne Illness Detection. ICL2FID consists of three steps. Initially, in the word-level labeling step, we leverage the CoT method to guide the LLM to first access the post's overall relevance to foodborne illness incident before identifying relevant entities within. Subsequently, in the word-level label verification step, the model is instructed to first evaluates the identified entity's relevance to the foodborne illness incident, then determine its validity. Irrelevant entity are discarded. The final step involves presenting the model with word-level labeling outcomes

and instructing it to verify these results before concludes whether the post indicates a foodborne illness incident. Labeled posts in the dataset compose the demonstration example set. Distinct retrieval strategy is employed at each step to ensure a diverse range of demonstration posts and labels. This approach helps avoid repetitive exposure to the same posts and labels throughout the labeling process, thus reducing potential biases.

Contributions. Our key contributions are as follows:

• We propose ICL2FID, the first labeling framework that employs LLMs to annotate posts with two-level labels for foodborne illness detection. ICL2FID generates word and post level labels in a sequence of steps. To better utilize the interconnection between post and word levels, ICL2FID instructs the LLM to leverage information from one level when it makes a prediction on the other level. As we will demonstrate in Section 4.3.2, this yields improved labeling results on both levels.

• To mitigate model hallucination, we incorporate a verification step between word and post level labeling. This verification step eliminates incorrect entities extracted in the former step, preventing them from influencing subsequent labeling outcomes. In this verification step, we introduce *Existence Diversity Similarly*, a demonstration example retrieval method to provide the model with both positive (extracted entity is correct) and negative (extracted entities are not correct) examples, encouraging a thorough analysis of whether previously extracted entities are genuinely related to foodborne illness incidents. At the post level labeling step, we propose *augmented diversity similarity*, another demonstration example retrieval method. This method prepares the model to access if extracted entities from previous step indicate a foodborne illness incident, It ensures the model is exposed to examples where word-level labeling results may or may not be correct, fostering a more nuanced evaluation.

• Through evaluations with varying the size of the demonstration example set (training set for supervised learning method), we demonstrate that ICL2FID not only outperforms traditional supervised learning approaches but also advances beyond ICL-based methods

utilizing the same LLM model, even with a very limited number of labeled posts. It is interesting to observe that its performance is in fact very close to aggregated crowd-sourced human annotation, albeit at a significantly reduced cost, as discussed in Section 4.4. Furthermore, our investigation into the optimal number of examples for demonstration contexts has yielded insights into balancing quality label generation against economic efficiency. These findings highlight ICL2FID's potential as a viable alternative for label collection in scenarios with limited resources.

4.2 Related Works

Large Language Model and In-context Learning for The Annotation Creation Task. Large language models (LLMs) have revolutionized natural language processing tasks by achieving significant performance improvements [12]. One simple yet effective application of pretrained LLMs is in-context learning (ICL), a technique where LLMs, using a *demonstration* of a few examples, generate text that aligns with the given context. Here, demonstration refers to the sample inputs and outputs provided to the model, serving as a guide for the expected task performance [7, 11]. ICL is a training-free learning framework. This could substantially lower computational costs associated with adapting models to new tasks [11]. Additionally, ICL capability can be further improved through a continual training stage, model warmup, between pretraining and ICL inference [127, 128]. Warmup is an optional procedure for ICL, which is not a focus in our study.

A notable advancement in ICL, Chain-of-Thought (CoT), introduces an intermediate reasoning step into the demonstrations to enhance LLMs' performance on complex tasks by predicting both the reasoning process and the final answer [45]. This approach mirrors the principles of multi-task learning (MTL), where models trained on multiple related tasks simultaneously can often outperform those trained on individual tasks. The underlying concept is that learning related tasks together allows the model to generalize better by leveraging

commonalities and differences across tasks [129]. While ICL and CoT methods have been applied to tasks like Named Entity Recognition [43], document information extraction [130], machine translation [131], and Relation Extraction [42] recently, the exploration of leveraging these techniques as part of a solution strategy for tackling the multi-level labeling task, specifically leveraging the interconnections between labels across different levels, remains an open problem.

Foodborne Illness Detection Dataset Labeling. Social media data has been identified as a great source of information for public health due to its timeliness and scalability. Previously, most research in this domain focused on collecting a single class label per tweet, specifically determining its relevance to foodborne illness events. These studies utilized machine learning models to identify relevant Yelp reviews or tweets within specific regions [115, 132, 133, 118, 117, 119]. However, unfortunately, most of the more detailed information had to be retrieved manually during the inspection process. In our prior research, we introduced TWEET-FID [4], the first publicly available annotated dataset for detecting foodborne illness incidents at two distinct levels. TWEET-FID, curated from Twitter [2] , is annotated at both the tweet and word levels, with labels provided by both experts and crowdsourced workers. By being a publically released resource, this dataset paves the way for future research in foodborne outbreak detection.

However, the reliance on human annotators to label data presents a significant cost barrier, especially for datasets of substantial size. Given that Large Language Models (LLMs) have shown exceptional performance across numerous NLP tasks, and considering the lower and in some cases even negligible expenses compared to hiring human annotators, investigating the capabilities of LLMs for labeling tasks in foodborne illness detection presents a promising avenue for exploration.

[2]Twitter has been renamed as X. The data collection were carried out when the Twitter API was accessible for academic research.

4.3 Our Proposed Methodology

4.3.1 Problem Definition

Let $D = \{\mathbf{x}_i\}_{i=1}^{N}$ denote a dataset of social media posts collected via the Twitter API, using keywords associated with foodborne illnesses, where N represents the total number of posts in D. Each post $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,M^i}]$ is a sequence of M^i words. The dataset is divided into a labeled set $D^l = \{(\mathbf{x}_i^l)\}$ and an unlabeled set $D^u = \{\mathbf{x}_i^u\}_{i=1}^{N^u}$, with D being the union of both D^u and $\{\mathbf{x}_i^l\}_{i=1}^{N^l}$. For all $\mathbf{x}_i^l$ in D^l, have a mapping to a triplet $(\mathbf{x}_i^l, y_i^l, \mathbf{s}_i^l)$. The post-level label $y_i^l \in \{0, 1\}$ indicates whether $\mathbf{x}_i^l$ describes a foodborne illness incident (1 for yes, 0 for no), while $\mathbf{s}_i^l = [s_{i,1}^l, s_{i,2}^l, \ldots, s_{i,M^i}^l]$ denotes the sequence of word-level labels for all words inside of $\mathbf{x}_i^l$, categorizing each word into one of five classes as detailed in Table 4.1. Due to the high cost of obtaining human annotations, most posts remain unlabeled, leading to a ratio of $\frac{N^u}{N^l} \gg 1$.

This study aims to identify four types of relevant entities (food, location, symptom, and keyword) within a post, with all other words classified as "outside" these entities of interest. Here, only entities directly associated with a foodborne illness incident are considered relevant. For instance, in the sentence “I ate an apple and it tastes great!”, the food entity “apple” is not implicated in a potential foodborne illness incident and thus should be classified as outside relevant entities.

In our study, we explore the feasibility of deploying a pretrained Large Language Model (LLM), denoted as $\Phi(\theta)$, to annotate the unlabeled dataset D^u on both word and post levels through in-context learning. Drawing on the concept of in-context learning—defined in prior research [11] as the capability of language models to adapt to specific tasks through exposure few examples—we pose the following problem: Given the labeled set D^l as the pool of demonstration examples, can we design an ICL-based strategy that accurately annotates

Label	Definition
Food	The food item that caused the potential foodborne illness incident.
Location	The location where the affected person purchased or acquired the food associated with the potential foodborne illness.
Symptom	The symptom experienced by the affected person as a result of the suspected foodborne illness.
Keyword	Other relevant keyword or term associated with a foodborne illnesses incident, *e.g.*, "food poisoning".
Out of relevant entity	Words that does not belong to any classes described above. Note that mentions of entities that are not related to a foodborne illness incident should be seen as "out of relevant entity"

Table 4.1: Definition of word level label classes.

D^u?

4.3.2 Proposed Approach: ICL2FID

Next, we design ICL2FID, a framework leveraging Large Language Models (LLMs) to annotate social media posts for foodborne illness detection on two levels. Given the complexity of assigning labels at both the post and word levels—where we target the extraction of four specific types at the word level—it becomes necessary to craft a comprehensive input context for the LLM. This input must encapsulate the task description, definitions of the four entity types, and possibly include examples to illustrate the desired output format. Many Large Language Models (LLMs) are constrained by input and output length limits. For instance, GPT-3.5-turbo imposes a maximum of 16,385 tokens for input and 4,096 tokens for output, while GPT-4 permits even fewer tokens, capping input at 8,192 tokens [123]. This limitation becomes significant when considering the length of social media posts; Facebook posts can extend up to 63,206 characters, X Message (formerly Twitter Message) allows 10,000 characters, and Instagram posts can be up to 2,200 characters [134]. Given that one English text token is approximately equivalent to four characters [135], many social media posts fall well within these limits. However, the challenge arises in including both a

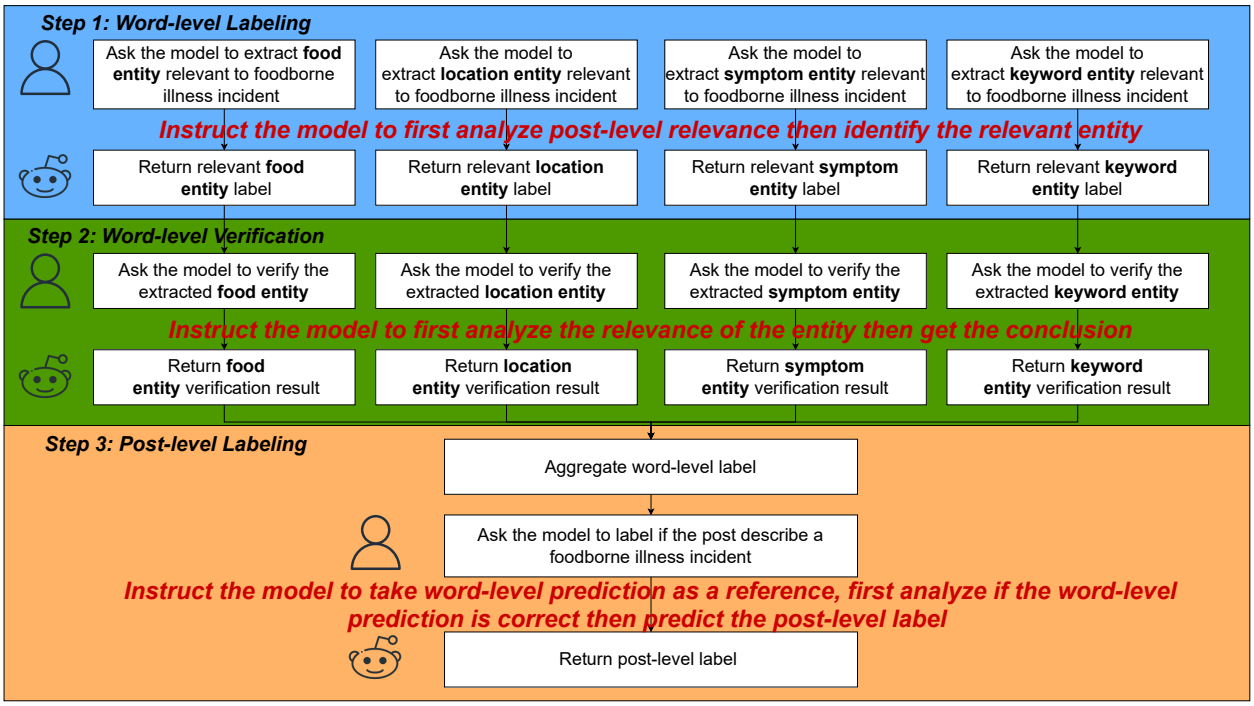


Figure 4.3: Pipeline of ICL2FID. The labeling framework is composed of three steps. Step 1 Word-level Labeling: the LLM is asked to extract relevant entities from the given post. Step 2 Word-level Verification: the LLM verifies the prediction from the previous step and filters out spurious entities. Step 3 Post-level Labeling: taking word-level label information for reference, the LLM generates the post-level label for the post.

descriptive context and multiple examples for each relevant entity and post-level label within a single input due to these token restrictions. In our experiment, we have tested to construct the instruction to ask the model to return labeled sentences (see description in next subsection) for four four types of relevant entities and the post-level class. Unfortunately, we sometimes encountered failures where the model could not return an answer due to the token limits. To overcome this limitation, we design the annotation process as a sequence of three distinct phases:

1. **Word Level Labeling:** We instruct the model $\Phi(\theta)$ to identify the four types of relevant entities for each post $\mathbf{x}_i^u$ in the unlabeled dataset D^u, conducting separate iterations for each entity type. In this step, we leverage the Chain-of-Thought (CoT) method [45] to guide the LLM to first consider the post's overall relevance to foodborne

illness incident before identifying relevant entities within;

2. **Word Level Label Verification:** LLM significantly suffers from the hallucination or overprediction issue [125]. In our case, LLM has a strong inclination to overconfidently label irrelevant words as relevant entities [43]. After extracting the relevant entities for post $\mathbf{x}_i^u$, to alleviate the model hallucination issue, $\Phi(\theta)$ is utilized to verify the correctness of each extracted entity. In this verification step, the LLM is instructed to evaluate the identified entity's relevance to foodborne illness incident in the reasoning. Entities which are verified as irrelevant are discarded at this step;

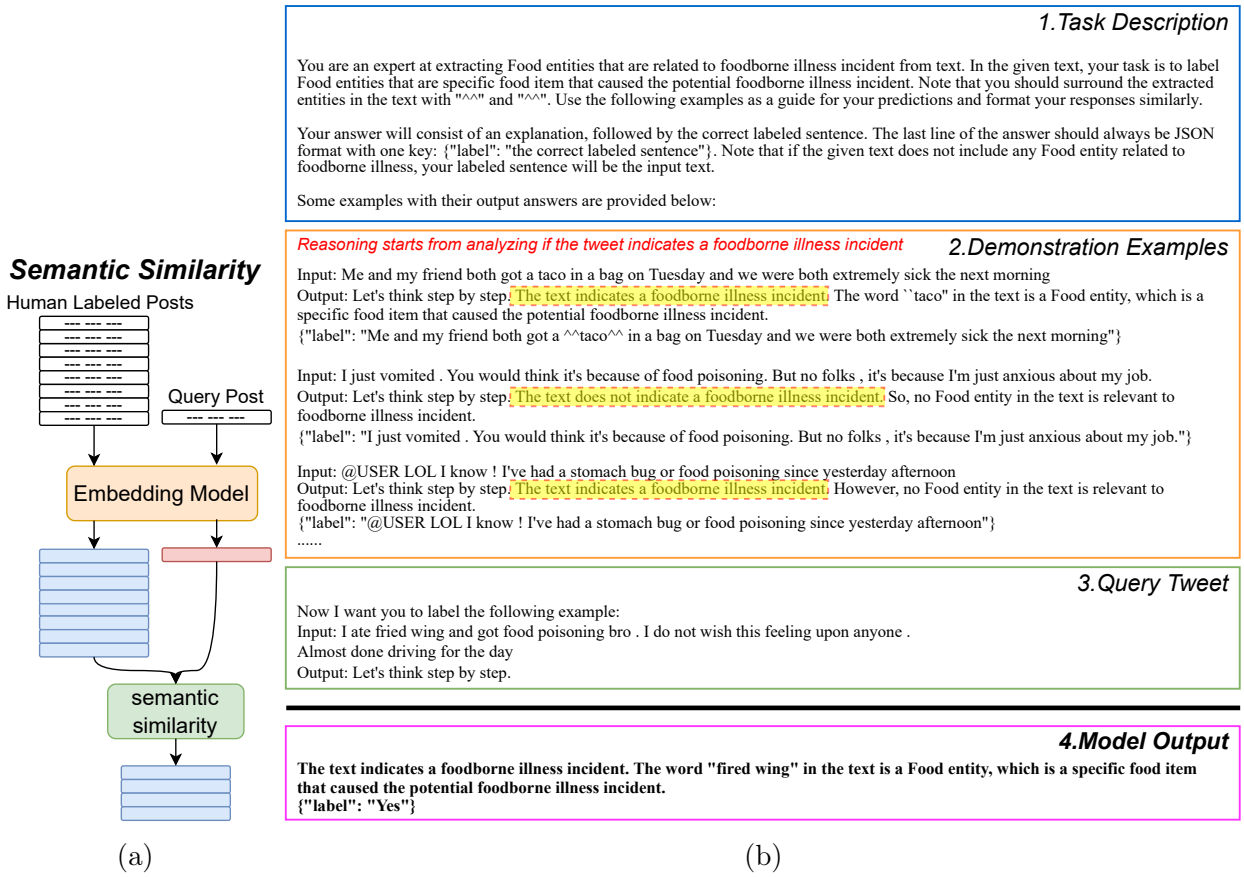
3. **Post Level Labeling:** With the post $\mathbf{x}_i^u$ and its verified entities as context, we instruct $\Phi(\theta)$ to determine the overall post-level label. In this step, the LLM is instructed to first analyze the word-level labeling result's correctness then get the conclusion for the final output.

The pipeline of ICL2FID is shown in Figure 4.3. Note that, the whole procedure is "training free", which means the model does not update any parameters. Since we are using the GPT-3.5-turbo as the backbone LLM, we don't deploy the LLM (and the embedding model, which is detailed in the following subsection) in our local machine, thus the labeling procedure can be done with limited computation resource.

In this design, the model is instructed to give word-level prediction at the first step. And in this step, the model also analyze post-level relevance in the reasoning step. However, due to the model hallucination issue, we cannot simply trust the results. In the second step, we introduce the model to verify the word-level results, filter out irrelevant entities. At the last step, we instruct the model to reconsider if entities from the second step does signal a foodborne illness incident and then give the conclusion for the post-level label. Through our design, the model is instructed to generate and verify the results from distinct point of views - search for evidence by itself and analyzes the correctness of evidence. Which can efficiently utilizes the interconnection between two levels of labels and mitigate hallucination issue.

In the following sections, we delve into the methodologies for constructing prompts and designing demonstration examples using a custom-retrieval strategy for each phase in detail.

4.3.2.1 Step 1: Word Level Labeling



(a) (b)

Figure 4.4: An overview of word-level labeling step. Left (4.4a): Semantic similarity example retrieval strategy. Right (4.4b): an example of word-level labeling prompt. Content above the bold black line is the input to the LLM, and the content below the bold black line is the LLM's output. The input composes of three parts: task description, demonstration examples and query post. Note that there could be more than three posts as demonstration examples.

In the word-level labeling step, the LLM is instructed to identify four types of **relevant** entities from the given post. As we discussed in last subsection, due to input and output length restrictions, it is impractical to include descriptions and sufficient demonstrations for all four entity types in a single prompt. Consequently, for each query post, we create four separate prompts, each tailored to one of the entity types.

Figure 4.4b presents an example of word-level labeling. The prompt is divided into three sections. The initial section provides a task overview, instructing the model to identify food entities associated with foodborne illness incidents. The opening sentences outline the task, followed by a description of the expected answer format. Drawing inspiration from GPT-NER [43], we suggest that the LLM's output adhere to a specific format. Namely, if the query post lacks any relevant food entities, $\Phi(\theta)$ simply replicates the query post $\mathbf{x}_i^u$; while for any relevant food entity or entities within the post, we encase them in special tokens "^^" to highlight their presence. This approach, as discussed in [43], effectively narrows the gap between traditional sequence labeling tasks and generative modeling. For a given sentence "I ate chicken and got diarrhea". The intuitive format of word-level label sequence is: "O O FOOD O O SYMPTOM", where "O" denotes "outside" relevant entity, "FOOD", "SYMPTOM" denote relevant food and symptom entity respectively. This intuitive format requires the model to learn the alignment between word and label, which add up the difficulty for the model to generate the label sequence. But this new output format design from [43] simplifies the model's task to merely marking the locations of the extracted entities while replicating the rest of the text within the same context of its sentence structure.

In the original work in [43], "@@" and "##" were used to denote extracted entities, but given the frequent use of "@" and "#" in social media posts, in our work, we instead select "^" as label notation to minimize confusion.

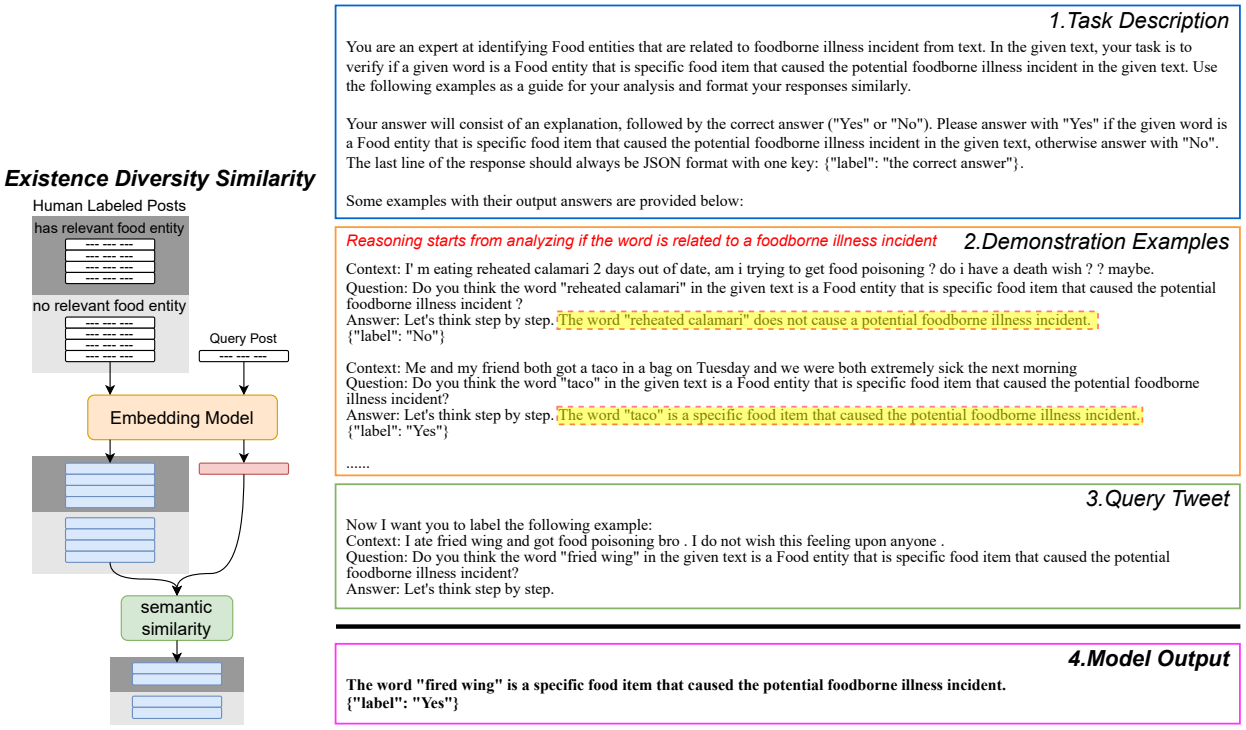
In the task description's third sentence, we specify the expected output format. However, drawing inspiration from the Chain of Thought (CoT) technique [45], we encourage the model to engage in a reasoning process before providing an answer. As illustrated in the demonstration examples in 4.4b. The model first is asked to assesses the overall relevance of the given post to a foodborne illness incident and then identifies the relevant entities within the post. Upon reaching a conclusion, it formats the output according to our specification. The concluding sentence of the task description signals that next we will provide a few-shot demonstration to guide the model's response via some illustrative examples.

Research has demonstrated that examples that are semantically closely related to the query post can notably enhance model performance [136, 137, 43]. For this step, we employ the semantic similarity selection (SSS) strategy [136] to carefully select the most suitable examples from our labeled subset D^l. Figure 4.4a details this process, where a embedding model processes all examples in the labeled set D^l to obtain their corresponding embedding vectors, which represent the post's semantic content. All these embedding can be collected and stored in memory before the labeling procedure. Then the embedding model process the query post to obtain its embedding vector. We then identify and retrieve the examples most semantically aligned with the query post, enriching the model's context for more accurate labeling. More implementation details for this similarity search procedure are referred to [138].

4.3.2.2 Word Level Label Verification

LLMs often grapple with hallucination or overprediction issues as illustrated in [125]. This problem can be particularly pronounced in our context of post-level labeling, where the accuracy of entity extraction as an intermediate step in the overall prediction process is crucial. Our task demands that the model identifies entities specifically **related** to foodborne illness incidents. However, a post might mention similar entities unrelated to such incidents. For instance, in the previously mentioned example, "I ate an apple and it tastes great!", the word "apple" might be mistakenly tagged as a relevant food entity despite its irrelevance to a foodborne illness. This may occur simply because an apple is indeed a food entity; so in that sense matches the desired outcome to some degree.

To mitigate this issue of model hallucination in our FID context and to ensure spurious entities do not compromise post-level predictions, we adopt a word-level label verification step inspired by GPT-NER [43] to filter out irrelevant entities extracted in the first step. This verification step can give the model a chance to consider from a distinct perspectives



(a) (b)

Figure 4.5: An overview of word-level verification step. Left (4.5a): Existence diversity similarity example retrieval strategy. This method provides the model with both positive (extracted relevant entity result is correct) and negative (extracted relevant entity result is not correct) examples. Right (4.5b): an example of word-level verification prompt. Content above the bold black line is the input to the LLM, and the content below the bold black line is the LLM's output. The input composes of three parts: task description, demonstration examples and query post. Note that in the prompt there could be more than two posts as demonstration examples.

whether the previously extracted entity is truly related to a foodborne illness incident. This intermediate step requires the model to assess the relevance of entities identified in the preceding step. Figure 4.5b illustrates the prompt used for this verification. Similar to the previous stage, the prompt is divided into three sections. In this step, we apply the Chain of Thought (CoT) method, prompting the model to analyze whether the identified word is indeed a type of entity related to a foodborne illness, followed by a simple "yes" or "no" response. Entities verified as "yes" are retained, while those receiving a "no" are excluded. If the first labeling step is akin to asking a student to solve for the unknown in an equation, this verification step is comparable to having the student substitute the value of the unknown

back into the equation to verify if the equality holds true. This helps ensure that relevant entities are more likely returned and thus will influence the final post-level label.

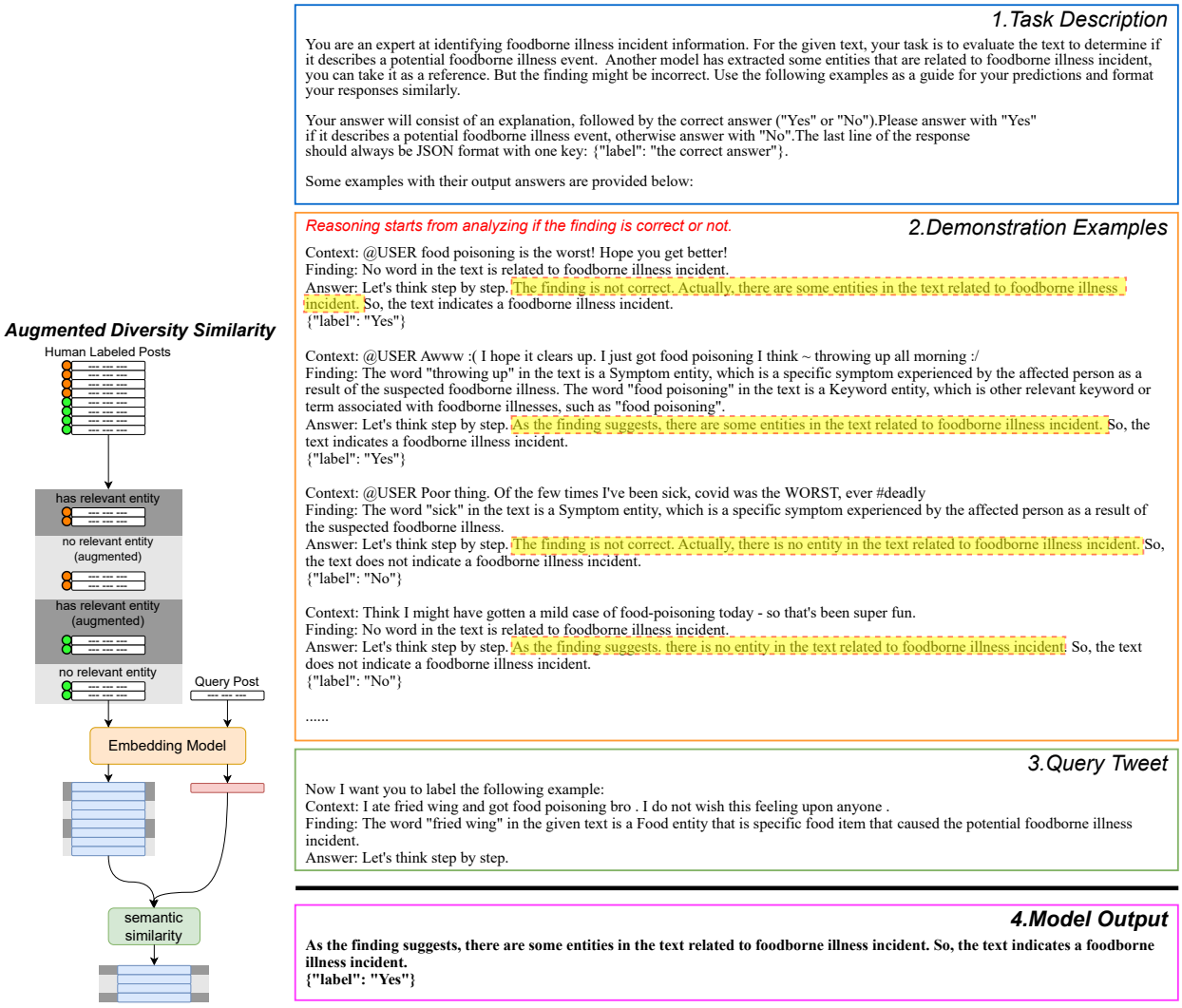
Figure 4.5b displays a series of demonstration examples (in the orange box) for the food entity verification task. All these demonstration examples are retrieved from the labeled set D^l. The questions for each example are constructed based on their word-level labels. We design a standardized template that then is utilized to compose all these questions, namely: "Do you think the word ENTITY_WORD in the given text is a Food entity that is the specific food item that caused the potential foodborne illness incident?" with ENTITY_WORD, the placeholder. That is, if the post contains a relevant food entity, this entity is inserted as the ENTITY_WORD in the question template, and the corresponding response is "Yes". Conversely, if the post lacks a relevant food entity, a random text span from the post is used as ENTITY_WORD, and the answer is "No".

Recall that for the word-level labeling step, we employed a semantic similarity selection strategy to select promising examples. However, utilizing this same method at this verification step could lead to selecting an identical set of posts as demonstration examples. Worse yet, if all demonstration examples mention food entities, the model might be biased towards responding with "Yes", even if the food entity extracted in the previous step doesn't pertain to a foodborne illness.

To address this, we now design an example retrieval strategy for this verification step, termed *Existence Diversity Similarity* that overcomes this challenge. First, for this food entity verification task, we initially divide all labeled instances in D^l into two subsets: $D^l_f = \{\mathbf{x}^l_i | \exists s^l_{i,j} \in \mathbf{s}^l_i, s^l_{i,j} = \text{Food}\}$, comprising posts with relevant food entities, and $D^l_{uf} = \{\mathbf{x}^l_i | \forall s^l_{i,j} \in \mathbf{s}^l_i, s^l_{i,j} \neq \text{Food}\}$, containing posts without relevant food entities. We then apply the semantic similarity method to retrieve an equal number of examples from both D^l_f and from D^l_{uf}. This approach ensures the prompt includes both positive and negative examples and an equal number of both, guiding the model to recognize that it can either confirm or

refute the entity identifications from the preceding step.

4.3.2.3 Post Level Labeling



(a) (b)

Figure 4.6: An overview of post-level labeling step. Left (4.6a): Augmented diversity similarity example retrieval strategy. This method filps some examples' word level label to provide the model with both positive (word-level label result is correct) and negative (word-level label result is incorrect) examples. Right (4.6b): An example of post-level labeling prompt. The prompt composes of three parts: task description, demonstration examples and query post. Content above the bold black line is the input to the LLM, and the content below the bold black line is the LLM's output. The input composes of three parts: task description, demonstration examples and query post. Note that in the real prompt there could be more than four posts as demonstration examples.

After the verification step, we now aggregate the confirmed relevant entities across the four categories into one single set per post. This aggregated data then is leveraged for the construction of the prompt for the subsequent post-level labeling step. Here, the model is tasked with determining whether the post indicates a foodborne illness. Figure 4.6b presents a prompt example for realizing this step. Utilizing the Chain of Thought (CoT) method, the model is directed to first evaluate related entities information (referred to as "findings" in the prompt) from earlier steps, analyze if these entities truly signals a foodborne illness incident, and then conclude whether the post describes a foodborne illness. This approach allows the model to reference the word-level labeling outcomes as a basis for generating the post-level label without blindly follow these earlier results. The model retains the power to revise prior conclusions based on its subsequent analysis.

In this labeling step, we want to present the model two scenarios: correct (positive) and word-level label results is not correct (negative). However, in our labeled set D^l, all instances' word-level result is correct. To address this issue, we introduce an additional example retrieval strategy named *Augmented Diversity Similarity*. As depicted in Figure 4.6a, we categorize instances in D^l based on their post-level labels into two groups: $D_1^l = (\mathbf{x}_i^l, y_i^l, \mathbf{s}_i^l)|y_i^l = 1$ and $D_0^l = (\mathbf{x}_i^l, y_i^l, \mathbf{s}_i^l)|y_i^l = 0$. We then randomly augment 50% of posts from both D_1^l and D_0^l with new word-level labels. For the selected post in D_0^l, some text spans in the post are randomly selected and assigned with arbitrary relevant entity labels. Whereas for the chosen posts in D_1^l, we will mark every word's corresponding word-level label as belonging to the "out of entity" type. Through this way, we create negative examples which can be used in the demonstration. After the augmentation process, demonstration examples are again selected based on semantic similarity. Since we flipped 50% of word-level labels, for each query post, there would be roughly 50% of examples are negative cases.

In Figure 4.6b, both the first and third demonstration examples stem from this augmentation process. The first post says: "@USER food poisoning is the worst! Hope you get better!" In this example, "food poisoning" should be identified as a relevant keyword entity.

In contrast, the "finding" mistakenly claims "No word in the text is related to foodborne illness incident", which the example answer corrects by first emphasizing the presence of relevant keywords, then drawing the conclusion that this text indicates a foodborne illness incident. In the third example, the post "@USER Poor thing. Of the few times I've been sick, covid was the WORST, ever #deadly" does not indicate a foodborne illness incident. Further, no word in the text is related to a foodborne illness incident. The word "sick" is wrongly labeled as a symptom entity that is related to foodborne illness incident. For these negative examples, the corresponding answers in demonstration start from refuting the word-level label results then get the conclusion on post-level label. These augmented examples serve to remind the model that initial findings are not infallibly accurate, urging a thorough analysis of the post to reach an accurate conclusion. This design strategically reduces the likelihood of the model being misled by incorrect word-level labeling results.

4.4 Experimental Study

This section assesses the effectiveness of our proposed method using the TWEET-FID dataset [4] and compares it against a variety of baseline approaches. Additionally, we conduct an ablation study evaluating our method alongside several of its variants to highlight the significance of each component within our framework.

Social Media Dataset. In our previous research, we have developed and publically released the TWEET-FID dataset [4], which includes 1,362 (33%) relevant and 2,760 (67%) irrelevant tweets related to foodborne illness. Each tweet has been labeled by both experts and through a crowdsourcing process, creating a richly annotated resource. As detailed in Chapter 1.1.1 and in Section 4.1, during crowdsourcing label collection procedure, we have rejected some low-quality crowdsourced annotations, and aggregate remained annotation per tweet to improve its quality compared to a single crowdsourced annotation. The dataset was segmented into training, validation, and testing sets with the aim to make all three

sets have the same balance of positive (tweet indicates foodborne illness incident) versus negative (tweet does not indicate foodborne illness incident) determined based on expert labels. That is, 1,088 relevant and 2,210 irrelevant tweets are designated for training, and both the validation and test sets comprise 137 relevant and 275 irrelevant tweets each.

For our current LLM-based study, the validation set serves as the demonstration example set D^l, while the training and test sets are merged to form the unlabeled set D^u. We only use the training and test sets label for performance evaluation. To ensure the model isn't overwhelmed by excessively long tweets and reduce labeling cost, we excluded tweets exceeding 42 words in length. This cutoff represents the third quartile of tweet lengths within our demonstration example set, leaving 311 tweets for use as demonstration examples. As discussed in Section 4.4.2, we can reduce the labeling cost without affect the labeling quality. Note that we do not exclude any tweet in the unlabeled set D^u since our goal to annotate all of them.

Experimental Setup. Our experiments leverage the gpt-3.5-turbo [122] as the primary LLM, chosen for its balance between performance and cost-efficiency, making it preferable to the more expensive GPT-4 for our purposes. Additionally, we utilize the Text-embedding-ada-002-v2 [139] model, recognized as the most advanced second-generation embedding model available. We configured gpt-3.5-turbo with a temperature setting of 0.1 as recommended in [140] to ensure precision in responses. Our methodology includes the use of 8 demonstration examples in each prompt, with a detailed discussion on selecting the optimal number of examples provided in Section 4.4.4. The entire labeling framework is developed using Python 3.9.12, incorporating the AutoLabel module [141] for creating demonstration contexts and facilitating interactions with the LLM and the embedding model via the OpenAI API [142]. Tasks such as label format transformation and aggregation, saving labeling results, evaluation, and visualization are executed using Pandas [143], Sklearn [144], SeqEval [145], Matplotlib [146], and other Python modules. The code and additional implementation details will be made available on GitHub following the publication of our work.

For both word and post level analysis, we employ F1 score and *balanced accuracy* (B.Acc) [147] to measure each method's performance. B.Acc s particularly effective for imbalanced datasets, where traditional accuracy metrics may provide a skewed view of a method's effectiveness. These metrics collectively offer a multifaceted perspective on the performance of each method, accommodating for the challenges posed by imbalanced class distributions. Given budgetary constraints, we conducted a single implementation of all methods based on gpt-3.5-turbo and employed the bootstrap resampling technique to estimate the mean and standard deviation of performance scores across the unlabeled set.

Baselines. Our study incorporates a range of widely-used supervised learning models as baseline comparisons. These supervised learning model are either trained or fine-tuned with demonstration example set D^u. That is ensure that, for this dataset, these methods' knowledge scope of these methods are within the demonstration example set. Since our ICL-based solution can only select demonstration examples from D^u. Supervised learning methods and ICL-based methods are therefore comparable. To evaluate if our proposed method could be an viable alternative for labeling collection. We also compare our method performance and cost against aggregated crowdsourced annotations. Additionally, we conduct an ablation study and examine key variants of ICL2FID to assess the impact of cross-level information prompting, the verification step, example retrieval strategies, exclusion of long demonstration example, and the order of labeling steps.

Supervised Learning Method. As described above, these methods are either trained or fine-tuned on the entire demonstration example set:

1. **RoBERTa.** Proposed by Liu et al. [53], RoBERTa refines the BERT [52] pre-training procedure by eliminating the next-sentence prediction task and optimizing training with larger mini-batches and learning rates. Outperforming BERT and other state-of-the-art model across multiple benchmarks, RoBERTa is implemented in both independent and joint versions. The independent version predicts at one of the levels, while

the joint version simultaneously predicts at both the word and post levels. Word-level predictions utilize a linear classification layer over each token's hidden state to predict its corresponding relevant entity class label. For post-level prediction, RoBERTa employs a classification head atop the [CLS] token's hidden state to predict its post-level relevance label. The independent variant only features with one of the classification heads described above. For the joint variant, it features two classification heads, each dedicated to a specific task.

2. **BERTweet.** Introduced by [148], BERTweet is the inaugural large-scale pre-trained language model for English tweets, adhering to the BERT-base architecture [52] and RoBERTa's pre-training methodology [53]. Outperforming previous models on tweet dataset benchmarks [148], The architecture of BERTweet for two-level in single level are the same with the RoBERTa.

3. **BiLSTM**. Bidirectional LSTM processes sequences using pre-trained GloVe embeddings[3]. The BiLSTM's hidden states feed into a classifier for prediction. Like RoBERTa, BiLSTM is adapted for both independent and joint version. For post-level predictions, it uses the concatenated final hidden states from both directions, while word-level predictions employ the hidden states from each word. The joint version assigns two classification heads to address each task independently.

Aggregated Human Annotation. In our previous research [4], we collected labels from crowdsource workers for each tweet on a crowdsourcing platform, with each tweet receiving five labels per level. We filtered out low-quality labels, retaining three per level for analysis. To aggregate these labels, we utilized two approaches: majority voting (MV) and the Bayesian sequence combination (BSC) method as proposed in [149]. MV was applied at both the word and tweet levels, while BSC was specifically employed for word-level label aggregation due to its suitability for sequence label tasks. By evaluating the quality and

[3]We utilize GloVe embeddings from the Common Crawl dataset, comprising 840 billion tokens and 2.2 million vocabularies. https://nlp.stanford.edu/projects/glove/

cost of labels produced by ICL2FID against those aggregated from crowdsourcing, our goal is to ascertain if ICL2FID can equal or exceed human annotation quality, thereby presenting a cost-efficient and effective approach for generating labels.

ICL2FID variants. To validate the impact of individual components, we introduced several variants of ICL2FID for an ablation study:

1. **ICL2FID -Independent.** This variant isolates the word and post-level labeling steps as two independent tasks. That is, we do not utilize the word-level labeling information to inform the post-level labeling. For instance, in Figure 4.6b, it excludes the "finding" and the related reasoning step in demonstration examples and the query tweet, with adjustments made to the task description to reflect this change. It thus also omits the instruction for models to consider post-level relevance before making word-level predictions. In Figure 4.4b, it omits the reasoning step highlighted in yellow color in demonstration example, with the task description alternation to match the change. This variant still remains the step 2 verification.

2. **ICL2FID w/o Step 2.** Omitting the word-level label verification step, this variant directly uses word-level labeling results in the final step without prior verification.

3. **ICL2FID w/ Extra Verification Step.** This variant add one more word-level/post-level verification step after the step 3 in original ICL2FID.

4. **ICL2FID w/ Semantic Similarity (SS) only.** Here, the example retrieval strategy for all steps is limited to the semantic similarity strategy. The existence diversity similarity and augmented diversity similarity strategies are not employed in Steps 2 and 3. Consequently, for Step 2, there's no assurance that demonstration examples for a query will include both cases where the entity extracted from Step 1 is correctly identified and cases where it is incorrectly identified. In Step 3, without negative cases in the demonstration examples (where the word-level label result is incorrect), the

model is not prompted to assess the accuracy of word-level labels. Instead, it proceeds based on the assumption that the word-level labels are correct to generate the post-level label. The task description and demonstration examples (refer to Figure 4.6b) are adjusted accordingly to reflect these changes.

5. **ICL2FID w/ Random Retrieval (RR) only.** Here, the example retrieval strategy for all steps is limited to the random selection strategy. This approach does not ensure that the examples chosen for demonstration are similar to the query post. Although augmentation is employed in step 3, there is no assurance that a balanced mix of positive and negative cases will be present in the demonstration examples for steps 2 and 3.

6. **ICL2FID w/ All Labeled Data.** This variant employs the same framework as the original ICL2FID. The sole distinction is that it utilizes a labeled set comprising all labeled tweets, including those exceeding 42 words in length, without any exclusions.

7. **ICL2FID Reversed Order.** This variant reverses the order of operations, starting with post-level labeling and verification before proceeding to the word-level labeling. Mirroring the original ICL2FID design, it instructs the model to analyze word-level entity information for post-level labeling first. The post-level results are then used as a reference for word-level labeling, ensuring a thorough and informed analysis across levels.

4.4.1 Experimental Results

Comparison of ICL2FID with Baseline Methods. Table 4.2 demonstrates that our method, ICL2FID, surpasses all non-human baselines in both word and post-level predictions. Unlike supervised learning methods that require updates to model parameters to tune the pretrained language models, ICL2FID can produce high-quality labels without such a model

Learning	Model	Method	Word-level		Post-level	
			F1	B.Acc	F1	B.Acc
Supervised	BERTweet	Independent	0.1873 ± 0.0051	0.661 ± 0.009	0.6077 ± 0.0122	0.7113 ± 0.0076
Supervised	RoBERTa	Independent	0.2588 ± 0.0062	0.701 ± 0.008	0.6912 ± 0.0106	0.7715 ± 0.0075
Supervised	BiLSTM	Independent	0.3638 ± 0.0057	0.688 ± 0.007	0.6361 ± 0.0105	0.7280 ± 0.0069
Supervised	BERTweet	Joint	0.1808 ± 0.0002	0.612 ± 0.006	0.4742 ± 0.0102	0.6278 ± 0.0062
Supervised	RoBERTa	Joint	0.2172 ± 0.0055	0.625 ± 0.008	0.5252 ± 0.0118	0.6414 ± 0.0079
Supervised	BiLSTM	Joint	0.3336 ± 0.0068	0.675 ± 0.009	0.5911 ± 0.0112	0.6934 ± 0.0076
In Context	gpt-3.5-turbo	ICL2FID Independent	0.5609 ± 0.0092	0.6693 ± 0.0114	0.6819 ± 0.0097	0.7682 ± 0.0060
In Context	gpt-3.5-turbo	ICL2FID	**0.6010 ± 0.0088**	**0.6760 ± 0.0110**	*0.7171 ± 0.0093*	*0.8000 ± 0.0058*
Human	Crowdsourcing	MV	*0.5908 ± 0.0146*	*0.6701 ± 0.0160*	**0.7759 ± 0.0082**	**0.8515 ± 0.0051**
Human	Crowdsourcing	BSC	0.5414 ± 0.0141	0.6711 ± 0.0157	N/A	N/A

Table 4.2: Performance comparison against SOTA methods on the Tweet-FID dataset. **Bold** scores are the highest, and *italic* scores the second highest in each metric.

learning and/or fine-tuning step. This ICL2FID Independent also demonstrates superior performance compared to other supervised learning methods, showcasing the impressive capabilities of GPT-3.5-turbo. GPT-3.5-turbo benefits from extensive pretraining on a larger dataset, which endows it with robust in-context learning abilities not present in traditional language models.

Comparison of ICL2FID with Human Labelers. Remarkably, our model's performance closely rivals that of aggregated human labels and even exceeds word-level labels aggregated via both the BSC and the MV method. The cost-effectiveness of ICL2FID is also notable: while crowdsourcing labels costs approximately $0.50 per tweet, labeling with GPT-3.5-turbo costs about $0.0005 to $0.001 per tweet, factoring in both input and output tokens as detailed in [126]. Additionally, obtaining labels from human crowdsourcers can take several days to weeks, whereas ICL2FID, utilizing the OpenAI API, can process labels within a few hours. Given the significant time and financial costs associated with gathering human labels through crowdsourcing, this result thus indicates that ICL2FID may be offering a valuable alternative for label generation in resource-constrained scenarios.

Model	Method	Word-level		Post-level	
		F1	B.Acc	F1	B.Acc
gpt-3.5-turbo	ICL2FID Independent	0.5609 ± 0.0092	0.6693 ± 0.0114	0.6819 ± 0.0097	0.7682 ± 0.0060
gpt-3.5-turbo	ICL2FID w/o Step 2	0.5031 ± 0.0087	**0.6848 ± 0.0106**	0.7051 ± 0.0095	0.7858 ± 0.0066
gpt-3.5-turbo	ICL2FID w/o Extra Verification Step	0.5963 ± 0.0089	0.6551 ± 0.0114	0.6290 ± 0.0098	0.7147 ± 0.0075
gpt-3.5-turbo	ICL2FID w/ SS only	0.5913 ± 0.0088	0.6609 ± 0.0112	0.6935 ± 0.0100	0.7771 ± 0.0065
gpt-3.5-turbo	ICL2FID w/ Random Retrieval	0.4992 ± 0.0083	0.4398 ± 0.0097	*0.7101 ± 0.0092*	0.7853 ± 0.0065
gpt-3.5-turbo	ICL2FID w/ All Labeled Data	*0.5957 ± 0.0090*	0.6634 ± 0.0110	0.7058 ± 0.0095	*0.7897 ± 0.0062*
gpt-3.5-turbo	ICL2FID Reversed Order	0.5571 ± 0.0088	0.5699 ± 0.0116	0.6816 ± 0.0116	0.7628 ± 0.0079
gpt-3.5-turbo	ICL2FID	**0.6010 ± 0.0088**	*0.6760 ± 0.0110*	**0.7171 ± 0.0093**	**0.8000 ± 0.0058**

Table 4.3: Ablation study of ICL2FID on the Tweet-FID dataset. **Bold** scores are the highest, and *italic* scores the second highest in each metric.

4.4.2 Ablation Study of ICL2FID

As shown in Table 4.3, ICL2FID independent variant highlights the benefits of leveraging the connection between post and word-level labels to enhance model performance. The significant difference in word-level F1 scores with and without the verification step emphasizes the critical role of word-level verification in improving label accuracy and minimizing the impact of erroneous word-level predictions on post-level outcomes. Word-level B.Acc is the average of recall rate for all type entities, ICL2FID's word-level B.Acc is a little bit lower than the variant without verification step because ICL2FID filtered out few relevant entities in the verification step. Interestingly, the variant incorporating additional verification steps performs worse than the original ICL2FID, suggesting that further verification and rectification do not necessarily lead to enhanced labeling outcomes.

The variant relying solely on semantic similarity for example retrieval slightly performs below that with all similarity selection strategies at both levels, suggesting that our introduced Existence Diversity Similarity and Augmented Diversity Similarity methods effectively reduce example set bias and boost model performance. The variant relying on random retrieval performs worse on word-level labeling, indicating that demonstration example similar to the query text enhances model performance. This variant's post-level performance is close to original ICL2FID, since it still has the augmented negative examples in the post-level labeling step, which can guide the model to first carefully verify the word-level labeling results

then get the post-level conclusion.

By comparing ICL2FID with its variant using all labeled data, we can conclude that removing excessively long tweet can reduce labeling cost and improve the overall performance. Additionally, the disparity in word-level F1 and B.Acc between ICL2FID and its reversed-order variant underscores the effectiveness of leveraging word-level labeling for the subsequent post level inference. This may be so because in scenarios indicating a foodborne illness, the presence of multiple relevant entities within a post can provide strong clues for post-level labeling, even if some entities are overlooked. However, incorrect initial post-level predictions in the reversed-order variant may adversely affect subsequent word-level labeling, leading to the identification of incorrect entities in cases of false positives or missing entities in cases of false negatives.

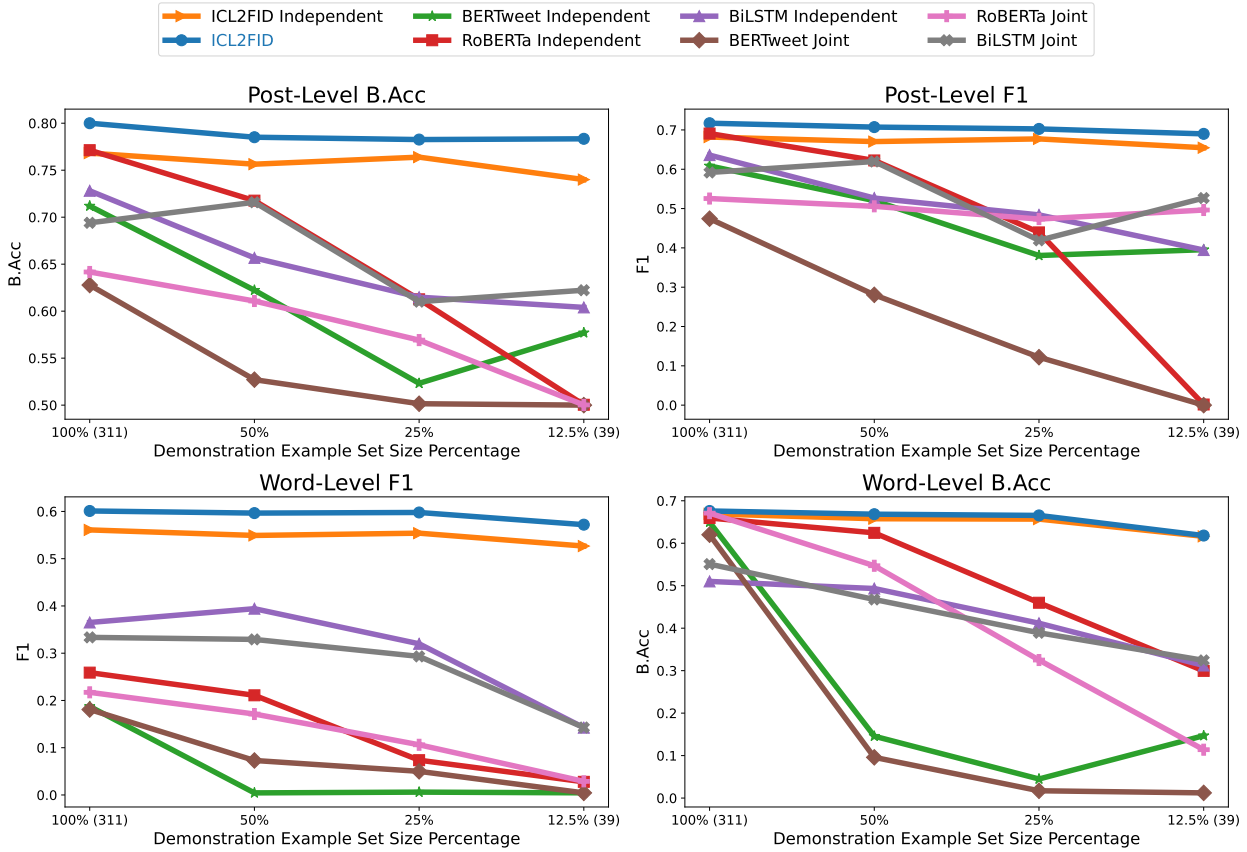


Figure 4.7: Performance comparison against SOTA methods under varying size of demonstration example set.

4.4.3 Effect of Size of Demonstration Example Set

To assess the resilience of our method ICL2FID in scenarios with an extremely limited number of labeled data, we conducted experiments varying the size of the demonstration example set. This was achieved by randomly selecting a certain sized subset of tweets from the overall example set. As depicted in Figure 4.7, the X-axis percentage values indicate the proportion of tweets retained for use as demonstration examples. Our findings reveal that ICL2FID maintains fairly consistent performance across various sizes of the demonstration example set as we look from left to right, i.e., we decrease the number of examples from 311 to 39. In contrast, the effectiveness of other supervised learning methods declines more significantly as the size of the demonstration example set decreases. The performance of ICL2FID independent variant is robust but it is worse than the original ICL2FID. This observation underscores ICL2FID's capability to produce high-quality labels even when confronted with a small number of labeled tweets, highlighting its adaptability and efficiency in label generation under constrained conditions.

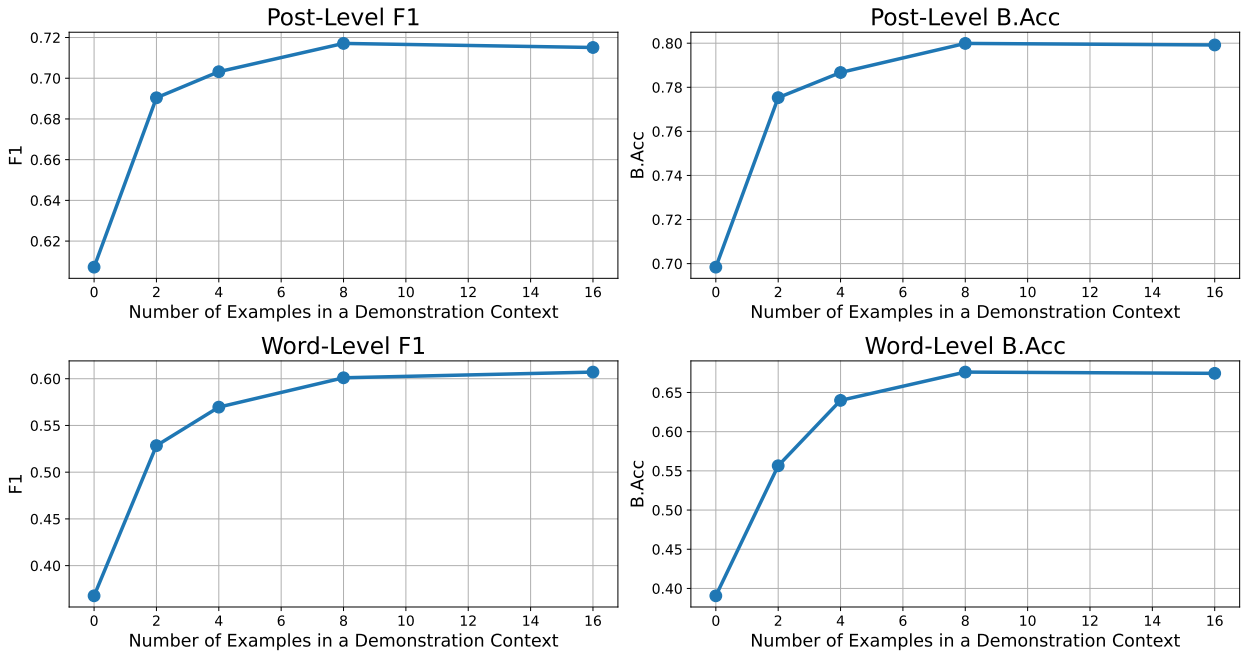


Figure 4.8: Performance of ICL2FID under varying number of examples included in a demonstration context.

4.4.4 Effect of Number of Demonstration Examples

Figure 4.8 showcases the performance of ICL2FID across varying quantities of demonstration examples. The data indicate a marked improvement in ICL2FID's effectiveness on both word-level and post-level tasks as the number of included examples is augmented, with performance plateauing beyond the inclusion of 8 examples. Given that these examples constitute the majority of the input context and that additional examples would significantly increase the input size, it is pragmatic to select 8 as the optimal number of examples for our methodology.

4.5 Conclusion

In this work, we introduce ICL2FID, a pioneering labeling framework that utilizes LLM to annotate posts with two-level labels aimed at detecting foodborne illnesses. ICL2FID incorporates several innovative strategies in order to leverage the semantic interrelationships between the two-leveled label structure. One key idea includes the utilization of the the CoT method for the initial word-level labeling task ICL2FID to guide the LLM to first access the post's overall relevance to foodborne illness incidents before identifying relevant entities within. Furthermore, to avoid the propagation and amplification of potential errors across these two levels, the strategy inserts the word-level label verification step that aims to verify the validity of each entity's relevance to the foodborne illness incident. Irrelevant entity are discarded, preventing them from influencing the subsequent labeling outcomes. Most importantly, equipped with the outcome of the word-level labeling task, supported by an instructing for the model to verify these results, the model is asked to infer whether the post indicates a foodborne illness incident. At each step, ICL2FID employs a small set of labeled posts as demonstration examples. Several example selection strategies are employed at each step to ensure a varied selection of posts and labels. This approach is designed to

prevent the model from being exposed to very similar or even the same data repeatedly, thereby minimizing potential biases.

ICL2FID capitalizes on the intricate relationship between the post and the word levels. In addition, it also effectively counters model hallucination, resulting in performance that surpasses existing methodologies. The labels generated by ICL2FID closely rival the quality of crowd-sourced human annotation, even with a markedly limited dataset. Note that the human crowd-source users also had undergone a training process where they were shown and explained examples of labeled posts. This demonstrates ICL2FID's capability as an efficient alternative for label collection, particularly in resource-constrained environments. It thus highlights its potential to advance the field of public health surveillance through social media.

Limitations. In this study, we experimented with various Large Language Models (LLMs) besides GPT-3.5-turbo as the foundational models for our framework, notably, Llama2 [9] and Refuel-LLM [150]. Although these models showed promise in information retrieval tasks, their performance was relatively inconsistent. Specifically, they sometimes struggled to comprehend our instructions, producing outputs that did not adhere to the desired format and were challenging to process further. Consequently, we opted for GPT-3.5-turbo as our primary model. A compelling avenue for future research would be devising strategies to effectively prompt LLMs to generate labels that strictly conform to the specified format, enhancing the utility and applicability of LLM-based frameworks in complex data annotation tasks.

Chapter 5

Conclusion

5.1 Summary of Contributions

In summary, my dissertation endeavors to navigate the challenges arising from incomplete, noisy, and multi-level labeled datasets. The dissertation is structured around three directions: 1) learning from two-level labeled datasets with one level having complete labels and the other having incomplete labels, 2) learning from datasets with noisy labels, and 3) in context learning of two-level labels when given a small number of labeled examples only.

For the first research direction, I explore the task of explainable Text Classification with Limited Human Attention Supervision. This task presents a set of training documents, each tagged with a classification label, with a smaller subset also bearing fine-grained word-level labels (HAMs). We introduce the open problem of explainable text classification with limited human attention supervision, given the scarcity of human attention maps (HAMs). Our proposed solution comprises two key components: a human-like attention learner and a contextualized representation, driven by a specially-designed joint loss function. This function harmonizes the supervision signals from both human-like attention generation and document classification tasks, despite their different numbers of labels across training instances.

Addressing the second research direction with task 2, we introduce CoLafier, a novel framework for learning with noisy labels. This framework consists of two central modules: the LID-based noisy label discriminator (LID-dis) and the LID-guided label generator (LID-gen). LID-dis ingests both the features and label of a training sample to generate a refined representation. CoLafier uses the LID scores from LID-dis to determine weights for each instance in our specialized loss function. Both LID-dis and LID-gen are trained using this weighted loss. They collaborate to determine label updates. To mitigate error accumulation, we employ two augmented perspectives for each instance using their corresponding LID scores to guide weight assignments and label update choices. Evaluations across multiple noise settings confirm that CoLafier significantly boosts prediction accuracy, outperforming state-of-the-art techniques.

For the third research direction, task 3 investigates the inference of label annotations for two-level foodborne illnesses detection task with limited labeled examples. This task provides us with a collection of tweets, the majority of which are unlabeled, with only a few having labels obtained from human annotators at both tweet and word levels.

For this task, we propose a novel labeling framework, ICL2FID, structured in three steps: word-level labeling, word-level label verification, and tweet-level labeling. At the labeling steps, ICL2FID utilizes the CoT method to guide the LLM to leverage insights from one level when it makes predictions at the other level. A critical verification step in between word and tweet level labeling steps eliminates incorrect entities extracted earlier, preventing them from influencing subsequent labeling outcomes. By employing example retrieval strategies at each stage, ICL2FID minimizes biase arising from repetitive exposure to identical tweets and labels, thereby effectively mitigating the risk of model hallucination.

The culmination of the three tasks delineated above represents a substantial stride towards the application of deep learning methodologies on incomplete, noisy, and multi-level labeled data. Each task embodies a contribution in a distinct subdomain of machine

learning research. Task 1 pioneers a solution for explainable text classification amid limited human attention supervision. Task 2 introduces a novel method leveraging LID scores of internal representations to discern correctly and incorrectly labeled data. Our proposed method on average outperforms SOTA techniques across a spectrum of noise settings. Task 3 offers a novel labeling framework that utilizes LLMs to annotate tweets with two-level labels aimed at detecting foodborne illnesses, addressing challenges of label incompleteness. Therefore, the insights and methodologies proposed in this dissertation are expected to benefit the broader machine learning community and its applications to important domain problems. It provides robust frameworks for tackling real-world challenges associated with label completeness, quality, and structure in the data.

5.1.1 Future Directions

In this dissertation, we have examined several problem settings related to learning from datasets characterized by incomplete, noisy, and multi-level labels. Below, we propose several avenues for future research that build upon the challenges and findings presented in this work:

1. Our research primarily addressed two-level labeled datasets with incomplete labels in the first and third tasks. However, the scenario where labels at both levels may be noisy was not explored. This scenario presents a unique challenge as the model must navigate the complexities of both incomplete and noisy labels. The difficulty of learning from a limited label set, compounded by the risk of overfitting on such a dataset, is further intensified by the introduction of label noise. This necessitates the development of a sophisticated strategy capable of mitigating label inaccuracies while efficiently leveraging the sparse labeled data across both levels.

2. In the second task, we introduced a novel framework designed to learn with noisy labels, wherein LID-dis utilizes both features and labels of training samples to refine their representations. For training LID-dis, labels are essential, yet the dataset may

exhibit issues of both label incompleteness and noise, often with only a handful of instances annotated with noisy labels. Extending our method to effectively make use of limited noisy labels poses a significant challenge and warrants further investigation.

3. The third task involved employing LLMs for labeling unlabeled data. Despite promising results, a performance disparity remains between LLM-generated labels and those provided by human experts. Bridging this gap presents an intriguing research opportunity. A potential solution could involve applying the noise-detection and label-purification techniques proposed in the second task to LLM-generated labels. Moreover, the intricate relationship between the two levels of labels studied in the first and third task offers additional prospects for improving the detection of inaccurate predictions.

4. While the primary focus of my dissertation has been on label-related issues, real-world datasets often encounter problems related to both feature completeness and noise [151], which are intricately linked to label quality [1]. The LID score, employed by our proposed method CoLafier for detecting noisy labels, was initially developed for identifying instances with noisy input. This approach has the potential to lay the groundwork for future research focused on concurrently improving both feature and label quality.

5. In Task 1, we adopt human-like attention maps as explanations for model predictions. Given that LLMs have been widely used across various domains, it would be intriguing to explore their potential to provide self-explanations for their behaviors. In Task 3, we employ the Chain of Thought (CoT) technique, instructing the model to rationalize its predictions in alignment with our demonstration examples. Here, we assume that the demonstration example set is labeled on both levels, allowing us to construct explanations for one level using information from the other. This approach could be adapted to situations where one level of labeling in the demonstration set is incom-

plete, presenting a greater challenge as the LLM would need to autonomously address missing information.

6. Due to budget constraints, in Task 3, we did not explore some advanced LLMs, including GPT-4-turbo [123] and Gemini [124]. These models are more expensive but offer potentially superior performance compared to GPT-3.5-turbo. Additionally, their capacity for longer inputs and outputs could make it feasible to instruct the model to generate labels for both levels in a single step.

7. Beyond the foodborne illness detection task, the framework developed in Task 3 could be applied to other domains. For instance, in the healthcare sector, the LLM could be used to derive diagnostic results from a patient's clinical notes. We could have clinicians annotate a few clinical notes, providing diagnostic conclusions and highlighting key supporting evidence. By using these annotations to construct demonstration examples, we could guide the LLM to not only return diagnostic results but also present the evidence supporting these conclusions.

These proposed directions underscore the complexity of learning from imperfect datasets and highlight the need for innovative solutions that can address the multifaceted challenges of data quality in machine learning.

Chapter 6

List of Publications

In Conference Proceedings:

1. **Dongyu Zhang**, Ruofan Hu, Elke Rundensteiner. *CoLafier: Collaborative Noisy Label Purifier With Local Intrinsic Dimensionality Guidance*, in 2024 SIAM International Conference on Data Mining (SDM), pp. 82-90, 2024.

2. Ruofan Hu, **Dongyu Zhang**, Dandan Tao, Huayi Zhang, Hao Feng, Elke Rundensteiner. *UCE-FID: Using Large Unlabeled, Medium Crowdsourced-Labeled, and Small Expert-Labeled Tweets for Foodborne Illness Detection*, in 2023 IEEE International Conference on Big Data (Big Data), pp. 5250-5259, 2023.

3. **Dongyu Zhang**, Liang Wang, Xin Dai, Shubham Jain, Junpeng Wang, Yujie Fan, Chin-Chia Michael Yeh, Yan Zheng, Zhongfang Zhuang, Wei Zhang. *FATA-Trans: Field and Time-Aware Transformer for Sequential Tabular Data*, conference paper in Proceedings of the 32nd ACM International Conference on Information & Knowledge Management (CIKM), 2023.

4. Ruofan Hu, **Dongyu Zhang**, Dandan Tao, Thomas Hartvigsen, Hao Feng, Elke Run-

densteiner. *TWEET-FID: An Annotated Dataset for Multiple Foodborne Illness Detection Tasks*, in Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC), pp. 6212-6222, 2022.

5. **Dongyu Zhang**, Cansu Sen, Jidapa Thadajarassiri, Thomas Hartvigsen, Xiangnan Kong, Elke Rundensteiner. *Human-like Explanation for Text Classification with Limited Attention Supervision*, in 2021 IEEE International Conference on Big Data (Big Data), pp. 957-967, 2021.

6. **Dongyu Zhang**, Jidapa Thadajarassiri, Cansu Sen, Elke Rundensteiner. *Time-Aware Transformer-based Network for Clinical Notes Series Prediction*, in Proceedings of the 5th Machine Learning for Healthcare Conference (MLHC), PMLR 126:566-588, 2020.

In Journals:

1. Dandan Tao, **Dongyu Zhang**, Ruofan Hu, Elke Rundensteiner, Hao Feng. *Epidemiological Data Mining for Assisting with Foodborne Outbreak Investigation*, journal article in Foods, vol. 12, no. 20, art. 3825, 2023.

2. Dandan Tao, Ruofan Hu, **Dongyu Zhang**, Jasmine Laber, Anne Lapsley, Timothy Kwan, Liam Rathke, Elke Rundensteiner, Hao Feng. *A Novel Foodborne Illness Detection and Web Application Tool Based on Social Media*, journal article in Foods, vol. 12, no. 14, art. 2769, 2023.

3. Dandan Tao, **Dongyu Zhang**, Ruofan Hu, Elke Rundensteiner, Hao Feng. *Crowdsourcing and machine learning approaches for extracting entities indicating potential foodborne outbreaks from social media*, journal article in Scientific Reports, vol. 11, Article no. 21678, 2021.

In Submission:

1. **Dongyu Zhang**, Ruofan Hu, Elke Rundensteiner. *LLM-based Two-Level Foodborne Illness Detection Label Annotation with Limited Labeled Samples*, submitted to the Proceedings of the 33rd ACM International Conference on Information & Knowledge Management (CIKM), 2024.

2. Harriet Sibitenda, Awa Diattara, Assitan Traore, Ruofan Hu, **Dongyu Zhang**, Elke Rundensteiner, Cheikh Ba. *Extracting Semantic Sentence Topics about Development from Social Network Comments: A case study on YouTube, Facebook, and Twitter* submitted to IEEE Access, 2024.

Bibliography

[1] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[2] B. Han, Q. Yao, T. Liu, G. Niu, I. W. Tsang, J. T. Kwok, and M. Sugiyama, "A survey of label-noise representation learning: Past, present and future," *arXiv preprint arXiv:2011.04406*, 2020.

[3] C. G. Northcutt, M. ChipBrain, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," *stat*, vol. 1050, p. 1, 2021.

[4] R. Hu, D. Zhang, D. Tao, T. Hartvigsen, H. Feng, and E. Rundensteiner, "TWEET-FID: An annotated dataset for multiple foodborne illness detection tasks," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, (Marseille, France), pp. 6212–6222, European Language Resources Association, June 2022.

[5] R. Hu, D. Zhang, D. Tao, H. Zhang, H. Feng, and E. Rundensteiner, "Uce-fid: Using large unlabeled, medium crowdsourced-labeled, and small expert-labeled tweets for foodborne illness detection," in *2023 IEEE International Conference on Big Data (BigData)*, pp. 5250–5259, IEEE, 2023.

[6] J. Liu, L. Wang, G. Dong, X. Song, Z. Wang, Z. Wang, S. Lei, J. Zhao, K. He, B. Xiao, *et al.*, "Towards robust and generalizable training: An empirical study of noisy slot filling for input perturbations," *arXiv preprint arXiv:2310.03518*, 2023.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[8] OpenAI, "Chatgpt." https://openai.com/blog/chatgpt, 2022. Accessed: 2024-03-29.

[9] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[10] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[11] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[12] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[13] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of the conference on empirical methods in natural language processing*, pp. 254–263, Association for Computational Linguistics, 2008.

[14] M. Lease, "On quality control and machine learning in crowdsourcing," *Human Factors*, vol. 11, no. 2, 2011.

[15] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[16] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[17] O. Lan, S. Zhu, and K. Yu, "Semi-supervised training using adversarial multi-task learning for spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6049–6053, IEEE, 2018.

[18] S. Zhu, R. Cao, and K. Yu, "Dual learning for semi-supervised natural language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1936–1947, 2020.

[19] M. Barrett, J. Bingel, N. Hollenstein, M. Rei, and A. Søgaard, "Sequence classification with human attention," in *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 302–312, 2018.

[20] Y. Zhang, I. Marshall, and B. 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*, vol. 2016, p. 795, NIH Public Access, 2016.

[21] J. Strout, Y. Zhang, and R. Mooney, "Do human rationales improve machine explanations?," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, (Florence, Italy), pp. 56–62, Association for Computational Linguistics, Aug. 2019.

[22] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, 2019.

[23] H. Song, M. Kim, and J.-G. Lee, "Selfie: Refurbishing unclean samples for robust deep learning," in *International Conference on Machine Learning*, pp. 5907–5915, PMLR, 2019.

[24] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on amazon mechanical turk," *Judgment and Decision making*, vol. 5, no. 5, pp. 411–419, 2010.

[25] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, *et al.*, "A closer look at memorization in deep networks," in *International conference on machine learning*, pp. 233–242, PMLR, 2017.

[26] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Nov 2015.

[27] A. J. Bekker and J. Goldberger, "Training deep neural-networks based on unreliable labels," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2682–2686, IEEE, 2016.

[28] S. Jenni and P. Favaro, "Deep bilevel learning," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 618–633, 2018.

[29] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. Alexander, and N. Silberman, "Learning from noisy labels by regularized estimation of annotator confusion," *Cornell University - arXiv,Cornell University - arXiv*, Feb 2019.

[30] N. Manwani and P. S. Sastry, "Noise tolerance under risk minimization," *IEEE Transactions on Cybernetics*, vol. 43, p. 1146–1151, May 2013.

[31] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31-1, 2017.

[32] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[33] H. Wei, L. Feng, X. Chen, and B. An, "Combating noisy labels by agreement: A joint training method with co-regularization," in *CVPR*, 2020.

[34] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *International conference on machine learning*, pp. 4334–4343, PMLR, 2018.

[35] Y. Li, H. Han, S. Shan, and X. Chen, "Disc: Learning from noisy labels via dynamic instance-specific selection and correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24070–24079, 2023.

[36] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *International Conference on Learning Representations*, 2019.

[37] Y. Wu, J. Shu, Q. Xie, Q. Zhao, and D. Meng, "Learning to purify noisy labels via meta soft label corrector," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35(12), pp. 10388–10396, 2021.

[38] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot filling models in natural language understanding," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, 2022.

[39] H. E, P. Niu, Z. Chen, and M. Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 5467–5471, Association for Computational Linguistics, July 2019.

[40] L. Qin, T. Liu, W. Che, B. Kang, S. Zhao, and T. Liu, "A co-interactive transformer for joint slot filling and intent detection," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8193–8197, 2021.

[41] J. Mei, Y. Wang, X. Tu, M. Dong, and T. He, "Incorporating bert with probability-aware gate for spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 826–834, 2023.

[42] Z. Wan, F. Cheng, Z. Mao, Q. Liu, H. Song, J. Li, and S. Kurohashi, "Gpt-re: In-context learning for relation extraction using large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3534–3547, 2023.

[43] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, and G. Wang, "Gpt-ner: Named entity recognition via large language models," *arXiv preprint arXiv:2304.10428*, 2023.

[44] L. Reynolds and K. McDonell, "Prompt programming for large language models: Beyond the few-shot paradigm," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–7, 2021.

[45] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022.

[46] D. Zhang, C. Sen, J. Thadajarassiri, T. Hartvigsen, X. Kong, and E. Rundensteiner, "Human-like explanation for text classification with limited attention supervision," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 957–967, IEEE, 2021.

[47] D. Zhang, R. Hu, and E. Rundensteiner, "Colafier: Collaborative noisy label purifier with local intrinsic dimensionality guidance," in *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pp. 82–90, 2024.

[48] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "Fndnet–a deep convolutional neural network for fake news detection," *Cognitive Systems Research*, vol. 61, pp. 32–44, 2020.

[49] D. Zhang, J. Thadajarassiri, C. Sen, and E. Rundensteiner, "Time-aware transformer-based network for clinical notes series prediction," in *Machine Learning for Healthcare Conference*, pp. 566–588, PMLR, 2020.

[50] H. Zhang, S. Sun, Y. Hu, J. Liu, and Y. Guo, "Sentiment classification for chinese text based on interactive multitask learning," *IEEE Access*, vol. 8, pp. 129626–129635, 2020.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[53] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[54] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. 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*, pp. 1480–1489, 2016.

[55] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Advances in Neural Information Processing Systems*, pp. 3504–3512, 2016.

[56] Y. Sha and M. D. Wang, "Interpretable predictions of clinical outcomes with an attention-based recurrent neural network," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 233–240, ACM, 2017.

[57] Y. Bao, S. Chang, M. Yu, and R. Barzilay, "Deriving machine attention from human rationales," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1903–1913, 2018.

[58] C. Sen, T. Hartvigsen, B. Yin, X. Kong, and E. Rundensteiner, "Human attention maps for text classification: Do humans and neural networks focus on the same words?," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[59] T. Qiao, J. Dong, and D. Xu, "Exploring human-like attention supervision in visual question answering," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[60] L. Chen, M. Zhai, and G. Mori, "Attending to distinctive moments: Weakly-supervised attention models for action localization in video," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 328–336, 2017.

[61] S. Liu, Y. Chen, K. Liu, and J. Zhao, "Exploiting argument information to improve event detection via supervised attention mechanisms," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1789–1798, 2017.

[62] Y. Zhao, X. Jin, Y. Wang, and X. Cheng, "Document embedding enhanced event detection with hierarchical and supervised attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 414–419, 2018.

[63] M. Nguyen and T. Nguyen, "Who is killed by police: Introducing supervised attention for hierarchical lstms," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2277–2287, 2018.

[64] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[65] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *Syntax, Semantics and Structure in Statistical Translation*, p. 103, 2014.

[66] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Neural machine translation with supervised attention," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3093–3102, 2016.

[67] S. Kuang, J. Li, A. Branco, W. Luo, and D. Xiong, "Attention focusing for neural machine translation by bridging source and target embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1767–1776, 2018.

[68] S. Schuster and C. D. Manning, "Enhanced english universal dependencies: An improved representation for natural language understanding tasks," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 2371–2378, 2016.

[69] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117, 2016.

[70] M. 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)*, pp. 4085–4094, 2019.

[71] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Proceedings of International Conference on Learning Representations*, 2015.

[72] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[73] J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace, "Eraser: A benchmark to evaluate rationalized nlp models," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[74] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 455–465, 2013.

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[76] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[77] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

[78] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing*, pp. 1532–1543, 2014.

[79] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, and S. Ananiadou, "Distributional semantics resources for biomedical text processing," *Proceedings of LBM*, pp. 39–44, 2013.

[80] D. Zhang, L. Wang, X. Dai, S. Jain, J. Wang, Y. Fan, C.-C. M. Yeh, Y. Zheng, Z. Zhuang, and W. Zhang, "Fata-trans: Field and time-aware transformer for sequential tabular data," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3247–3256, 2023.

[81] H. Zhang, L. Cao, S. Madden, and E. Rundensteiner, "Lancet: labeling complex data at scale," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, 2021.

[82] D. Hofmann, P. VanNostrand, H. Zhang, Y. Yan, L. Cao, S. Madden, and E. Rundensteiner, "A demonstration of autood: a self-tuning anomaly detection system," *Proceedings of the VLDB Endowment*, vol. 15, no. 12, pp. 3706–3709, 2022.

[83] X. Xia, T. Liu, B. Han, C. Gong, N. Wang, Z. Ge, and Y. Chang, "Robust early-learning: Hindering the memorization of noisy labels," in *International conference on learning representations*, 2020.

[84] H. Zhang, L. Cao, P. VanNostrand, S. Madden, and E. A. Rundensteiner, "Elite: Robust deep anomaly detection with meta gradient," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2174–2182, 2021.

[85] Y. Tu, B. Zhang, Y. Li, L. Liu, J. Li, Y. Wang, C. Wang, and C. R. Zhao, "Learning from noisy labels with decoupled meta label purifier," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19934–19943, 2023.

[86] M. E. Houle, "Local intrinsic dimensionality i: an extreme-value-theoretic foundation for similarity applications," in *Similarity Search and Applications: 10th International Conference, SISAP 2017, Munich, Germany, October 4-6, 2017, Proceedings 10*, pp. 64–79, Springer, 2017.

[87] X. Ma, B. Li, Y. Wang, S. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. Houle, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," *International Conference on Learning Representations,International Conference on Learning Representations*, Dec 2017.

[88] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," *International Conference on Machine Learning,International Conference on Machine Learning*, Jul 2018.

[89] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema, and J. Bailey, "Dimensionality-driven learning with noisy labels," in *International Conference on Machine Learning*, pp. 3355–3364, PMLR, 2018.

[90] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 630–645, Springer, 2016.

[91] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[92] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

[93] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[94] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," *University of Toronto*, 2009.

[95] X. Xia, T. Liu, B. Han, N. Wang, M. Gong, H. Liu, G. Niu, D. Tao, and M. Sugiyama, "Part-dependent label noise: Towards instance-dependent label noise," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7597–7610, 2020.

[96] Z. Zhu, T. Liu, and Y. Liu, "A second-order approach to learning with instance-dependent label noise," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, May 2021.

[97] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu, "Learning with noisy labels revisited: A study using real-world human annotations," in *International Conference on Learning Representations*, 2021.

[98] E. Malach and S. Shalev-Shwartz, "Decoupling" when to update" from" how to update"," *Advances in neural information processing systems*, vol. 30, 2017.

[99] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5552–5560, 2018.

[100] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?," in *International Conference on Machine Learning*, pp. 7164–7173, PMLR, 2019.

[101] K. Yi and J. Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7017–7025, 2019.

[102] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," *Advances in neural information processing systems*, vol. 33, pp. 20331–20342, 2020.

[103] C. Tan, J. Xia, L. Wu, and S. Z. Li, "Co-learning: Learning from noisy labels with self-supervision," in *Proceedings of the 29th ACM International Conference on Multimedia*, Oct 2021.

[104] E. Englesson and H. Azizpour, "Generalized jensen-shannon divergence loss for learning with noisy labels," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30284–30297, 2021.

[105] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1944–1952, 2017.

[106] Y. Xu, P. Cao, Y. Kong, and Y. Wang, "L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise," *Advances in neural information processing systems*, vol. 32, 2019.

[107] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama, "Are anchor points really indispensable in label-noise learning?," *Advances in neural information processing systems*, vol. 32, 2019.

[108] Y. Liu and H. Guo, "Peer loss functions: Learning from noisy labels without knowing noise rates," in *International conference on machine learning*, pp. 6226–6236, PMLR, 2020.

[109] H. Cheng, Z. Zhu, X. Li, Y. Gong, X. Sun, and Y. Liu, "Learning with instance-dependent label noise: A sample sieve approach," in *International Conference on Learning Representations*, 2021.

[110] S. Liu, Z. Zhu, Q. Qu, and C. You, "Robust training under label noise by over-parameterization," in *International Conference on Machine Learning*, pp. 14153–14172, PMLR, 2022.

[111] E. Scallan, R. M. Hoekstra, F. J. Angulo, R. V. Tauxe, M.-A. Widdowson, S. L. Roy, J. L. Jones, and P. M. Griffin, "Foodborne illness acquired in the united states—major pathogens," *Emerging infectious diseases*, vol. 17, no. 1, p. 7, 2011.

[112] S. Hoffmann and E. Scallan Walter, "Acute complications and sequelae from foodborne infections: informing priorities for cost of foodborne illness estimates," *Foodborne pathogens and disease*, vol. 17, no. 3, pp. 172–177, 2020.

[113] R. L. Scharff, "The economic burden of foodborne illness in the united states," in *Food safety economics*, pp. 123–142, Springer, 2018.

[114] D. Tao, D. Zhang, R. Hu, E. Rundensteiner, and H. Feng, "Crowdsourcing and machine learning approaches for extracting entities indicating potential foodborne outbreaks from social media," *Scientific reports*, vol. 11, no. 1, p. 21678, 2021.

[115] C. Harrison, M. Jorder, H. Stern, F. Stavinsky, V. Reddy, H. Hanson, H. Waechter, L. Lowe, L. Gravano, and S. Balter, "Using online reviews by restaurant patrons to identify unreported cases of foodborne illness—new york city, 2012–2013," *MMWR. Morbidity and mortality weekly report*, vol. 63, no. 20, p. 441, 2014.

[116] J. K. Harris, R. Mansour, B. Choucair, J. Olson, C. Nissen, J. Bhatt, C. for Disease Control, and Prevention, "Health department use of social media to identify foodborne illness - chicago, illinois, 2013-2014.," *MMWR. Morbidity and mortality weekly report*, vol. 63, 2014.

[117] A. Sadilek, H. Kautz, L. DiPrete, B. Labus, E. Portman, J. Teitel, and V. Silenzio, "Deploying nemesis: Preventing foodborne illness by data mining social media," in *Twenty-Eighth IAAI Conference*, 2016.

[118] J. P. Schomberg, O. L. Haimson, G. R. Hayes, and H. Anton-Culver, "Supplementing public health inspection via social media," *PloS one*, vol. 11, no. 3, p. e0152117, 2016.

[119] T. Effland, A. Lawson, S. Balter, *et al.*, "Discovering foodborne illness in online restaurant reviews," *Journal of the American Medical Informatics Association*, vol. 25, no. 12, pp. 1586–1592, 2018.

[120] A. Sadilek, S. Caty, L. DiPrete, *et al.*, "Machine-learned epidemiology: real-time detection of foodborne illness at scale," *NPJ digital medicine*, vol. 1, no. 1, p. 36, 2018.

[121] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. V. Le, *et al.*, "Least-to-most prompting enables complex reasoning in large language models," in *The Eleventh International Conference on Learning Representations*, 2022.

[122] OpenAI, "Gpt-3.5-turbo." https://openai.com/, 2023. Accessed: 2024-03-29.

[123] OpenAI, "OpenAI Models Overview." https://platform.openai.com/docs/models/overview, 2023. Accessed: 2024-03-29.

[124] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[125] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *arXiv preprint arXiv:2311.05232*, 2023.

[126] OpenAI, "Openai pricing." https://openai.com/pricing, 2023. Accessed: 2024-03-29.

[127] S. Min, M. Lewis, L. Zettlemoyer, and H. Hajishirzi, "Metaicl: Learning to learn in context," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, 2022.

[128] M. Chen, J. Du, R. Pasunuru, T. Mihaylov, S. Iyer, V. Stoyanov, and Z. Kozareva, "Improving in-context few-shot learning via self-supervised training," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3558–3573, 2022.

[129] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[130] J. He, L. Wang, Y. Hu, N. Liu, H. Liu, X. Xu, and H. T. Shen, "Icl-d3ie: In-context learning with diverse demonstrations updating for document information extraction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19485–19494, 2023.

[131] S. Sia and K. Duh, "In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models," in *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pp. 173–185, 2023.

[132] J. K. Harris, R. Mansour, B. Choucair, J. Olson, C. Nissen, and J. Bhatt, "Health department use of social media to identify foodborne illness—chicago, illinois, 2013–2014," *MMWR. Morbidity and mortality weekly report*, vol. 63, no. 32, p. 681, 2014.

[133] J. K. Harris, J. B. Hawkins, L. Nguyen, E. O. Nsoesie, G. Tuli, R. Mansour, and J. S. Brownstein, "Research brief report: using twitter to identify and respond to food poisoning: The food safety stl project," *Journal of Public Health Management and Practice*, vol. 23, no. 6, p. 577, 2017.

[134] Fanbooster, "The ideal length of everything online, backed by research." https://fanbooster.com/blog/social-media-post-lengths/, 2020. Accessed: 2024-03-29.

[135] OpenAI, "What are tokens and how to count them." https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them, 2024. Accessed: 2024-03-29.

[136] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?," in *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures* (E. Agirre, M. Apidianaki, and I. Vulić, eds.), (Dublin, Ireland and Online), pp. 100–114, Association for Computational Linguistics, May 2022.

[137] D. Vilar, M. Freitag, C. Cherry, J. Luo, V. Ratnakar, and G. Foster, "Prompting palm for translation: Assessing strategies and performance," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15406–15427, 2023.

[138] Langchain, "Vector stores - python langchain documentation." https://python.langchain.com/docs/modules/data_connection/vectorstores/, 2023. Accessed: 2024-03-29.

[139] OpenAI, "text-embedding-ada-002." https://openai.com/, 2023. Accessed: 2024-03-29.

[140] Refuel AI, "Guide to large language models in autolabel." https://docs.refuel.ai/autolabel/guide/llms/llms/, 2023. Accessed: 2024-03-29.

[141] Refuel AI, "Introduction to autolabel." https://docs.refuel.ai/autolabel/introduction/, 2023. Accessed: 2024-03-29.

[142] OpenAI, "Openai api." https://openai.com/api/, 2023. Accessed: 2024-03-29.

[143] W. McKinney *et al.*, "pandas: Powerful python data analysis toolkit," 2010.

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

[145] H. Nakayama, "seqeval: A python framework for sequence labeling evaluation," 2018.

[146] J. D. Hunter, "Matplotlib: A 2d graphics environment." Computing in Science & Engineering, 2007.

[147] R. Wang, X. Jia, Q. Wang, and D. Meng, "Learning to adapt classifier for imbalanced semi-supervised learning," *arXiv preprint arXiv:2207.13856*, 2022.

[148] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "BERTweet: A pre-trained language model for English Tweets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 9–14, 2020.

[149] E. Simpson and I. Gurevych, "A bayesian approach for sequence tagging with crowds," 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)*, (Hong Kong, China), pp. 1093–1104, Association for Computational Linguistics, nov 2019.

[150] Refuel AI, "Announcing refuel llm." https://www.refuel.ai/blog-posts/announcing-refuel-llm, 2023. Accessed: 2024-03-29.

[151] G. Algan and I. Ulusoy, "Label noise types and their effects on deep learning," *arXiv preprint arXiv:2003.10471*, 2020.

[152] S. Coles, "An introduction to statistical modeling of extreme values," *Springer London eBooks,Springer London eBooks*, Aug 2001.

[153] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

Appendix A

Appendix for Task 2

A.1 Local Intrinsic Dimensionality (LID)

Local Intrinsic Dimensionality (LID) serves as an expansion-centric metric, capturing the intrinsic dimensionality of a data's underlying subspace or submanifold [86]. Within intrinsic dimensionality theory, expansion models quantify the growth rate of in the number of data objects encountered as the distance from a reference sample expands [87]. To provide an intuitive perspective, consider a Euclidean space where the volume of an m-dimensional ball scales in proportion to r^m as its size is adjusted by a factor of r. Given this relationship between volume growth and distance, dimension m can be inferred using:

$$\frac{V_2}{V_1} = (\frac{r_2}{r_1})^m \Rightarrow m = \frac{\ln(V_2/V_1)}{\ln(r_2/r_1)} \tag{A.1.1}$$

By interpreting the probability distribution as a volume surrogate, traditional expansion models offer a local perspective on data's dimensional structure, as their estimates are confined to the vicinity of the sample of interest. Adapting the expansion dimension concept to the statistical realm of continuous distance distributions results in LID's formal definition [86]:

Definition 1 (Local Intrinsic Dimensionality). *For a data sample $x \in X$, let $r > 0$ represent the distance from x to its neighboring data samples. If the cumulative distribution function $F(r)$ is both positive and continuously differentiable at a distance $r > 0$, then the LID of x at distance r is expressed as:*

$$\mathrm{LID}_F(r) \triangleq \lim_{\epsilon \to 0} \frac{\ln\left(F((1+\epsilon)r)/F(r)\right)}{\ln(1+\epsilon)} = \frac{rF'(r)}{F(r)} \tag{A.1.2}$$

whenever the limit exists. The LID at x is subsequently defined as the limit as radius $r \to 0$:

$$\mathrm{LID}_F = \lim_{r \to 0} \mathrm{LID}_F(r) \tag{A.1.3}$$

Estimation of LID. Consider a reference sample point $x \sim \mathcal{X}$, where $\mathcal{X}$ denotes a global data distribution. Each sample $x_* \sim \mathcal{X}$ being associated with the distance value $d(x, x_*)$ relative to x. When examining a dataset X derived from $\mathcal{X}$, the smallest k nearest neighbor distances from x can be interpreted as extreme events tied to the lower end of the induced distance distribution[89]. Delving into the statistical theory of extreme values, it becomes evident that the tails of continuous distance distributions tend to align with the Generalized Pareto Distribution (GPD), a type of power-law distribution[152]. In this work, we adopt the methodology from [89], and employ the Maximum Likelihood Estimator, represented as:

$$\widehat{\mathrm{LID}}(x) = -\left(\frac{1}{k}\sum_{i=1}^{k} \log \frac{r_i(x)}{r_{\max}(x)}\right)^{-1} \tag{A.1.4}$$

Here, $r_i(x)$ signifies the distance between x and its i-th nearest neighbor, while $r_{\max}(x)$ represents the maximum of these neighbor distances. It's crucial to understand that the LID defined in (A.1.3) is a *distributional* quantity, and the $\widehat{\mathrm{LID}}$ defined in (A.1.4) serves as its *estimate*.

However, in practice, computing neighborhoods with respect to the entire feature set X can be prohibitively expensive, we will estimate LID of a training example x from its

k-nearest neighbor set within a batch randomly selected from X. Consider a L-layer neural network $h : \mathcal{X} \to \mathbb{R}^c$, where h_j is the transformation at the j-th layer, and given a batch $X_B \subset X$ and a reference point x, the LID score of x is estimated as[89]:

$$\widehat{\mathrm{LID}}(x, X_B) = -\left(\frac{1}{k}\sum_{i=1}^{k}\log\frac{r_i(h_j(x), h_j(X_B))}{r_{\max}(h_j(x), h_j(X_B))}\right)^{-1} \tag{A.1.5}$$

In this equation, $h_j(x)$ is the output from the j-th layer of the network. The term $r_i(h_j(x), h_j(X_B))$ represents the distance of $h_j(x)$ to its i-th nearest neighbor in the transformed set $h_j(X_B)$, and $r_{\max}$ is the neighborhood's radius. The value $\widehat{\mathrm{LID}}(x, X_B)$ indicates the dimensional complexity of the local subspace surrounding x after the transformation by h_j. If the batch is adequately large, ensuring the k-nearest neighbor sets remain in the vicinity of $h_j(x)$, the estimate of LID at $h_j(x)$ within the batch serves as an approximation to the value that would have been computed within the full dataset $h_j(X)$.

A.2 The Pseudo Code of CoLafier

The pseudo-code for CoLafier is presented in Algorithm 1. Initially, CoLafier undergoes a warm-up phase for T_0 epochs. Subsequent epochs involve loss weight assignment and label update influenced by LID scores. To counteract error accumulation, CoLafier integrates two augmented views for each sample, using their respective LID scores to guide weight calculation and label update.

A.3 The Design of of Equation 3.3.13-3.3.15

The design of $w_{i,c}$, $w_{i,h}$, $w_{i,n}$ in equations aims to ensure that the sum of $w_{i,c}$, $w_{i,h}$, and $w_{i,n}$ equals 1.

Algorithm 1 CoLafier algorithm.

Input: noisy dataset $\tilde{D} = \{(x_i, \tilde{y}_i)\}$, start epoch T_0, total epochs $T_{\max}$, total number of batches $B_{\max}$, LID-dis $f_{\text{LD}}(\Theta_{\text{LD}})$, LID-gen $f_{\text{GE}}(\Theta_{\text{GE}})$, $\lambda^*, \lambda_{cons}, \epsilon^W_{\text{low}}, \epsilon^W_{\text{high}}, \epsilon^U_{\text{low}}, \epsilon^U_{\text{high}}, \tau, \epsilon_k$.
Output: LID-gen $f_{\text{GE}}(\Theta_{\text{GE}})$
for $T = 1, ..., T_{\max}$ **do**
  for $B = 1, ..., B_{\max}$ **do**
    obtain a mini-batch $\tilde{D}_B = \{(x_i, \tilde{y}_i)\}_{i=1}^{N_B}$
    obtain view sets V_B^1 and V_B^2, and input pair sets $\tilde{D}_B^1, \tilde{D}_B^2, \tilde{D}_B^{1*}, \tilde{D}_B^{2*}$
    obtain prediction sets: $\hat{Y}_B^{k,G}$ from f_{GE}, and $\hat{Y}_B^{k,D}, \hat{Y}_B^{k*,D}$ from f_{LD}, where $k \in \{1, 2\}$
    if $T \leq T_0$ **then**
      obtain $\mathcal{L}_{\text{GE}} = \sum_{k=1}^{2} \left(\mathcal{L}_{\text{CE}}(\tilde{y}_i, \hat{y}_i^{k,G}) \right)$, $\mathcal{L}_{\text{GE}} = \sum_{k=1}^{2} \left(\mathcal{L}_{\text{CE}}(\tilde{y}_i, \hat{y}_i^{k,D}) + \lambda^* \mathcal{L}_{\text{CE}}(\tilde{y}_i, \hat{y}_i^{k*,D}) \right)$ {Warm-up}
    else
      obtain $\widehat{\text{LID}}^W(\tilde{D}_B^1)$, and $\widehat{\text{LID}}^W(\tilde{D}_B^2)$ {Using Equation 3.3.3-3.3.4 to get LID scores for weight assignment}
      obtain $\hat{D}_B^1, \hat{D}_B^2$ {Input pairs for predictions from f_{GE}}
      obtain $\hat{U}_B^k = \tilde{D}_B^k \cup \hat{D}_B^k$, and $\widehat{\text{LID}}^U(\hat{U}_B^k)$ {Using Equation 3.3.5-3.3.8 to get LID scores for label update}
      obtain $\{w_{i,c}\}, \{w_{i,h}\}$, and $\{w_{i,n}\}$ {Using Equation 3.3.9 - 3.3.15 to get weights for each loss term}
      obtain $\mathcal{L}_{\text{clean,GE}}, \mathcal{L}_{\text{hard,GE}}, \mathcal{L}_{\text{noisy,GE}}, \mathcal{L}_{\text{clean,LD}}, \mathcal{L}_{\text{hard,LD}}, \mathcal{L}_{\text{noisy,LD}}$ {Using Equation 3.3.16 - 3.3.27 to calculate weighted clean, hard, and noisy loss}
      obtain $\mathcal{L}_{\text{GE}} = \mathcal{L}_{\text{clean,GE}} + \mathcal{L}_{\text{hard,GE}} + \mathcal{L}_{\text{noisy,GE}}, \mathcal{L}_{\text{LD}} = \mathcal{L}_{\text{clean,LD}} + \mathcal{L}_{\text{hard,LD}} + \mathcal{L}_{\text{noisy,LD}}$
    end if
    $\Theta_{\text{GE}}^{B+1} = \text{AdamW}(\mathcal{L}_{\text{GE}}, \Theta_{\text{GE}}^B)$, and $\Theta_{\text{LD}}^{B+1} = \text{AdamW}(\mathcal{L}_{\text{LD}}, \Theta_{\text{LD}}^B)$
    if $T > T_0$ **then**
      for $i = 1, ..., N_B$ **do**
        obtain $\Delta\tilde{y}_i^k, \Delta\hat{y}_i^k$, where $k \in 1, 2$ {Using Equation 3.3.30 and 3.3.31 to calculate prediction difference}
        obtain $\tilde{t}_i^k, \hat{t}_i^k, \acute{y}_i^{k,G}$, where $k \in 1, 2$, then determine whether to update label $\tilde{y}_i$ with $\acute{y}_i^{k,G}$ or not {Using Equation 3.3.32-3.3.40 to make decision on label update}
      end for
    end if
  end for
end for

Proof. Without loss of generality, assume that $w_{i,1} > w_{i,2}$. Then:

$$w_{i,c} = w_{i,2},$$

$$w_{i,h} = w_{i,1} - w_{i,2},$$

$$w_{i,n} = 1 - w_{i,2}.$$

Hence $w_{i,c} + w_{i,h} + w_{i,n} = 1$. □

A.4 The Design of Equation 3.3.36 and 3.3.37

The design of $(2 - \Delta\hat{y}_i^k)/2$ and $(2 - \Delta\tilde{y}_i^k)/2$ terms in equations aims to map both $\Delta\hat{y}^{k_i}$ and $\Delta\tilde{y}^{k_i}$ into the interval $[0, 1]$. This is based on the fact that the range of Δ is $[0, 2]$.

Proof. 1. Consider vectors $y = [y_1, y_2, ..., y_n]$ and $u = [u_1, u_2, ..., u_n]$, where $y_i \in [0, 1], u_i \in [0, 1]$. $\sum_i^n y_i = 1$, and $\sum_i^n u_i = 1$. Define Δ as $\Delta = \sum_{i=1}^n |y_i - u_i|$.

2. Without loss of generality, assume that $y_i \geq u_i$ for $i \in [1, 2, ..., m]$ and $y_j < u_j$ for $j \in [m + 1, m + 2, ..., n]$. Then:

$$\Delta = \sum_{i=1}^{m} (y_i - u_i) + \sum_{j=m+1}^{n} (u_j - y_j).$$

3. Rearranging and utilizing the fact that the summation of each vector is 1, we deduce:

$$\Delta = \sum_{i=1}^{m} y_i - \sum_{i=1}^{m} u_i + \sum_{j=m+1}^{n} u_j - \sum_{j=m+1}^{n} y_j,$$

$$\sum_{i=1}^{m} y_i + \sum_{j=m+1}^{n} y_j = \sum_{i=1}^{m} u_i + \sum_{j=m+1}^{n} u_j,$$

$$\sum_{i=1}^{m} y_i - \sum_{i=1}^{m} u_i = \sum_{j=m+1}^{n} u_j - \sum_{j=m+1}^{n} y_j,$$
$$\Delta = 2(\sum_{i=1}^{m} y_i - \sum_{i=1}^{m} u_i).$$

4. Since $\sum_{i=1}^{m} y_i \in [0,1], \sum_{i=1}^{m} u_i \in [0,1]$ and $\sum_{i=1}^{m} y_i \geq \sum_{i=1}^{m} u_i \geq 0$, it is implied that $(\sum_{i=1}^{m} y_i - \sum_{i=1}^{m} u_i) \in [0,1]$.

5. Hence, $\Delta \in [0,2]$ and consequently, $\frac{2-\Delta}{2} \in [0,1]$. □

A.5 Experiment Setup

All experiments are executed using A100 GPUs and PyTorch 1.13.1. We use an AdamW [153] optimizer with a learning rate of 0.001 and a weight decay of 0.001. The training epochs are 200 and the batch size is 128. CoLafier first warms up for 15 epochs, during the warm-up stage, CoLafier is optimized with cross entropy loss from two views, without weight assignment or label update.

Inspired by [35], for CoLafier, we employ two separate augmentation strategies to produce two views. The first approach involves random cropping combined with horizontal flipping, and the second incorporates random cropping, horizontal flipping, and RandAugment [103]. The value of $\epsilon_{\text{low}}^{W}$ and $\epsilon_{\text{low}}^{U}$ are both 0.001. The value of $\epsilon_{\text{high}}^{W}$ and $\epsilon_{\text{high}}^{U}$ start at values of 0.05 and 0.5 respectively, linearly increase to 1.0 in 30 epochs. The value of ϵ_k is fixed at 0.1. The values of λ^* and λ_{cons} are 0.5 and 10 respectively.

In real-world applications, the noise ratio and pattern are often unknown. Earlier studies [32, 100, 36] assumed the availability of prior knowledge about the noise ratio or pattern, and they based their hyper-parameter settings on these assumptions. Recent works [35] contend that such information is typically inaccessible in practice. Even though these works claim not to rely explicitly on noise information, their hyper-parameters still change as the noise ratio

and type shift. For the sake of a fair comparison, we re-evaluated certain methods (indicated by a * symbol after the method name) using their open-sourced code. However, if these methods originally assumed unknown noise ratios or types, we kept their hyper-parameters consistent. The hyper-parameter settings we adopted were based on their medium noise ratio configurations for each noise type (50% symmetric noise, 40% asymmetric noise, 40% instance dependent noise). We executed each method five times and recorded the average of the top three accuracy scores obtained during the training process. We employ a distinct backbone model for instance-dependent noise and real-world noise to ensure our results are comparable with those presented in [110, 35].