석사학위논문
Master's Thesis

DynamicLabels: 크라우드 피드백을 활용한 머신러닝 레이블 셋 구축 지원 시스템

DynamicLabels: Supporting Informed Construction of Machine Learning Label Sets with Crowd Feedback

2023

박 정 언 (朴 貞 彦 Park, Jeongeon)

한국과학기술원

Korea Advanced Institute of Science and Technology

석사학위논문

DynamicLabels: 크라우드 피드백을 활용한 머신러닝 레이블 셋 구축 지원 시스템

2023

박정언

한국과학기술원

전산학부

DynamicLabels: 크라우드 피드백을 활용한 머신러닝 레이블 셋 구축 지원 시스템

박정언

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2023년 6월 1일

심사위원장 김주호 (인)

심사위원 오혜연 (인)

심사위원 송진영 (인)

DynamicLabels: Supporting Informed Construction of Machine Learning Label Sets with Crowd Feedback

Jeongeon Park

Advisor: Juho Kim

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Daejeon, Korea
June 1, 2023

Approved by

Juho Kim
Professor of School of Computing

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MCS

박정언. DynamicLabels: 크라우드 피드백을 활용한 머신러닝 레이블 셋 구축 지원 시스템. 전산학부 . 2023년. 33+iv 쪽. 지도교수: 김주호. (영문 논문)
Jeongeon Park. DynamicLabels: Supporting Informed Construction of Machine Learning Label Sets with Crowd Feedback. School of Computing . 2023. 33+iv pages. Advisor: Juho Kim. (Text in English)

초 록

레이블 셋 구축 – 서로 다른 레이블들의 그룹 및 위계를 정의하는 것 – 은 머신러닝 어플리케이션을 만들때 필수적인 단계이다. 잘못 설계된 레이블 셋은 머신러닝 어플리케이션의 다른 단계인 트레이닝 데이터 셋 구축, 모델 훈련, 모델 배치에 부정적인 영향을 미친다. 레이블 셋 구축의 중요성에도 불구하고, 머신러닝 실무자들은 특히 외부 레퍼런스가 없을때 잘 정의된 레이블 셋을 만드는 데에 어려움을 겪는다. 이러한 한계를 극복하기 위해 머신러닝 실무자들은 반복적인 구축 프로세스를 통해 피드백을 모으고 이를 활용하여 레이블 셋을 발전시킨다. 본 연구에서는 머신러닝 실무자들과의 인터뷰 (n=4)를 통해 해당 프로세스에서 필요한 피드백을 모아 최적의 선택을 하는 데에 여전히 어려움이 있음을 확인하고, 크라우드 피드백 기반 합리적인 레이블 셋 수정과정을 지원하는 DynamicLabels 시스템을 제안한다. 크라우드 워커들은 DynamicLabels를 통해 애노테이션을 진행 및 레이블을 제안하면서 머신러닝 실무자의 레이블 셋에 피드백을 남기고, 머신러닝 실무자는 다각도 분석을 통해 피드백을 검토하며 레이블 개선의 잠재적 결과를 확인할 수 있다. 두 가지의 데이터 타입을 사용한 피험자 내 실험 (n=16)을 통해 DynamicLabels가 머신러닝 실무자로 하여금 수집된 피드백을 더 잘 이해하고 탐색할 수 있도록 하며 보다 체계적이고 자신감 있는 개선 프로세스를 지원한다는 것을 발견했다. 머신러닝 실무자들은 놓칠 수 있었던 중복 케이스와 엣지 케이스들을 확인할 수 있었다. 또한, 크라우드 피드백을 통해 크라우드의 다양한 관점을 얻고, 현 레이블 셋의 약점을 발견하고, 참가자들의 레이블 제안들을 활용할 수 있었다. DynamicLabels을 통해 머신러닝 실무자는 크라우드로부터 구체적인 데이터에 대한 이해와 증거를 성공적으로 얻었고, 이를 바탕으로 합리적인 레이블 셋 개선을 성공적으로 수행했다.

핵 심 낱 말 인간-컴퓨터 상호작용, 크라우드소싱, 크라우드 피드백, 머신러닝, 머신러닝 실무자 지원, 레이블 셋 구축

Abstract

Label set construction — deciding on a group of distinct labels — is an essential stage in building a machine learning (ML) application, as a badly designed label set negatively affects other stages such as training dataset construction, model training, and model deployment. Despite its significance, it is challenging for ML practitioners to come up with a well-defined label set, especially when no external references are available. To mitigate this difficulty, ML practitioners often go through multiple iterations to gradually improve their label set. Through our formative study (n=4), we observed that there still remain challenges in collecting helpful feedback and utilizing them to make optimal refinement decisions. To support the iterative refinement, we present DynamicLabels, a system that aims to support a more informed label set-building process with crowd feedback. Crowd workers provide feedback as annotations and label suggestions to the ML practitioner's label set, and the ML practitioner can review the feedback through multi-aspect analysis and see the potential consequences of label refinements. Through a within-subjects study (n=16) using two datasets, we found that DynamicLabels enables better understanding and exploration of the collected feedback and supports a more structured, confident refinement process.

The ML practitioners were also able to see surfacing conflicts and edge cases that could have been ignored. In addition, the crowd feedback helped ML practitioners to gain diverse perspectives, spot current weaknesses, and shop from crowd-generated labels. With DynamicLabels, ML practitioners can successfully gain concrete understanding and evidence from the crowd and make informed refinements to iteratively improve the label set.

Keywords Human-Computer Interaction, Crowdsourcing, Crowd Feedback, Machine Learning, Machine Learning Practitioner Support, Label Set Construction

Contents

List of Tables

List of Figures

Chapter 1. Introduction

A ‘label’ or a class, is a word or a phrase that explains a piece of data in a machine learning (ML) model. A group of distinct labels works together as a ‘label set’ to provide the model with a set of candidate labels for classification [1]. For example, in classifying a clothing dataset, a label set may consist of four distinct labels: top, bottom, outer, and accessory. The label set is provided to the annotators to construct a training dataset and is utilized as model inputs and outputs.

Preparing a well-constructed label set is important to build a successful ML application. Building an ML application involves a multi-stage process, which includes (1) preparing the raw data, (2) building a label set, (3) using the label set to annotate the training data, (4) implementing and training the model, and (5) deploying the model. Every other stage in the process is highly interconnected with the label set building stage: an unclearly defined label set affects the outcome of the annotation, and an indistinct or low-coverage label set affects the performance of the model, which subsequently negatively affects the experience of the user in the deployment stage [2].

ML practitioners usually refer to existing labeled datasets or theories (e.g., referring to existing psychology taxonomies for emotion recognition models) to come up with the label set, and validate it with some additional data. However, this practice may not be sufficient in constructing a high-quality label set in many different situations. For example, applying a pre-established label set to real-world data requires revision of the label set to accurately represent the distribution of the data. When the model structure is decided prior to the data collection, the label set needs to be constructed in consideration of the model structure. Furthermore, building a label set from scratch for a new domain without a well-defined or representative label set requires a significant amount of feedback and consensus-building among ML practitioners. In the aforementioned situations, multiple iterations to refine the label set are essential to continuously improve the label set. With the iterations, ML practitioners collect bad signals (e.g., low coverage, unclear distinction) about the label set and revise based on the signals to prevent possible downstream issues, which is critical to the success of the ML application.

To understand the challenges of building label sets with iterative refinements, we conducted a formative study with 4 ML practitioners with experience constructing label sets from scratch and identified two major challenges. ML practitioners tended to iteratively develop the label set, yet found it challenging to collect large-scale, fresh-perspective feedback to improve the label set. Also, they found it difficult to confidently refine and decide on an optimal label set, due to many different aspects (e.g., clarity of each label, distribution of the data, clear boundary between the labels) they have to consider along with the uncertainty of each improvement decision.

To support collecting meaningful feedback and making informed decisions for refining the label set, we propose the idea of inviting crowd workers to provide feedback about the label set from varying perspectives. With the crowd as potential users of the deployed model, having the crowd’s collective opinions and suggestions on the ML practitioner-built label set will guide the refinements. Analyzing the opinions and suggestions will help ML practitioners make a more confident and knowledgeable refinement to the label set.

To explore the proposed idea, we propose DynamicLabels, a novel system that supports ML practitioners to iteratively construct their label set with label feedback collected from the crowd. When an ML practitioner provides an initial version label set, crowd workers produce feedback by annotating with the

ML practitioner's label set and making their own label set with the assigned data through the *feedback collection interface*. With the collected label feedback and suggestions, the ML practitioner is provided with multiple-aspect analyses of the feedback and a playground to test and iterate on their label set in the *label set refinement interface*.

We conducted a 2-day within-subjects study comparing DynamicLabels with the baseline annotation system to examine how DynamicLabels supports an informed label set refinement with crowd feedback. 16 ML practitioners built two label sets using two types of datasets (natural scene images and event fliers) to construct and refine two label sets through a round of iteration. The feedback collection interface of DynamicLabels enabled collecting large-scale, diverse feedback from the crowd, which participants identified as meaningful and useful. The refinement interface of DynamicLabels enabled a high-level understanding of the feedback, encouraged flexible refinements to be made, and supported a structured refinement process. In addition, it helped the participants spot possible issues and examine various refinement options. The crowd feedback helped the participants in understanding the crowd's perspectives and the weaknesses in their label set, and in making refinements. We also discuss how DynamicLabels can support various types of data as well as the goals of ML practitioners. In addition, we suggest further utilization of the crowd feedback in making better-informed decisions and discuss how DynamicLabels supports the construction of a user-centered model.

Our contributions are as follows:

- DynamicLabels, a system that supports ML practitioners' label set construction process with crowd feedback and feedback analysis. DynamicLabels consists of a feedback collection interface that collects annotation and label suggestions on the ML practitioner-built label set, and a label set refinement interface that supports ML practitioners to make comprehensive refinement decisions.

- Findings from a within-subjects study that compares DynamicLabels with a baseline system using two datasets, which shows that DynamicLabels supports an exploratory and structured refinement process, as well as in-depth analysis of how participants utilized crowd feedback in making label set refinements.

Chapter 2. Related Work

To situate our research, we review previous work related to label set construction, crowdsourcing feedback, and data-driven decision support. We first investigate challenges and issues in the label set construction process. Then, as we propose a system empowered by the crowd, we discuss how crowdsourcing is utilized to support experts. In the end, we review the decision-making support enabled by large-scale data.

2.1 Label set construction

When trying to train a model for their own task, an ML practitioner should examine existing data and construct a set of labels. When there are no external references, constructing a label set is more challenging as there is no standard in categorizing contents for a multi-class label set. One commonly used approach is applying clustering algorithms such as LDA [3] and EM with GMM model [4]. These algorithms work in an unsupervised manner and categorize data points to compose clusters. However, these algorithms are mostly limited to numerically represented structured data. When the data contents are complicated and unstructured, other additional numerically abstracting algorithms or models are required to use these algorithms. Furthermore, they often fail at achieving reliable performance because these algorithms may not work perfectly, and also machine-generated clusters may not have appropriate representations or labels for human understanding.

To mitigate the issues from machine-generated label sets, previous work has invited humans to participate in the label set building process [5, 6, 7, 8]. Cascade [5] presents a crowdsourcing workflow where workers provide suggestions and vote for the best descriptions over iterations, to generate reliable categories with the crowds. Deluge [6] employs a group of crowd workers to collaboratively produce a taxonomy comparable to that of experts. Alloy [7] suggests a human-machine hybrid workflow to cluster text clips. A machine categorizes the text clips leveraging the salient keywords identified by crowd workers, then they put additional effort into clustering machine-failed clips. Revolt [8] leverages disagreement of crowds' annotation on a data instance to build label sets. Based on the crowds' response in the annotation step, they generate an accurate and high-coverage label set.

In this work, we extend from prior works supporting the label set construction process through crowd feedback and support the iterative label set construction process with the crowd. Prior works involving the crowd to create and categorize labels or taxonomies either only utilize the crowd to create the final product without the intervention of the expert or the ML practitioners or do not involve the expert or the ML practitioners to utilize the crowd work to make iterative refinements. In DynamicLabels, we give the ML practitioner the main authority in constructing the label set, and have the crowd support the process by producing feedback based on the ML practitioner-built label set.

2.2 Utilizing the crowd to support expert work

We define the label set construction as an open problem where no one best solution exists, so offering a diverse range of responses would help ML practitioners find an optimal label set satisfying their needs.

Previous studies have shown that crowd inputs can help expert work by providing feedback on their work [9, 10, 11] and inspiration for improvements [12, 13, 14].

There exists previous work that leveraged crowd input as feedback to expert work. Voyant [9] collected structured crowd feedback on visual designs by providing five feedback types to the crowd. ProtoChat [10] collected multiple levels of feedback including utterance-level feedback and overall conversation feedback by asking questions while testing the conversation. CrowdCrit [11] introduced key sources in visual design for the crowd to refer to in making a critique, in order to collect detailed and actionable feedback.

Some previous work emphasizes the importance of incorporating crowd opinion in high-level concept or design of a product whose end user is a wide range of the public. Sutton and Lawson [12] proposed democratizing emoji design and selection by reflecting on how the public recognizes and uses emojis. Brambilla et al. [13] proposed a collaborative development process of Domain-Specific Modeling Languages (DSMLs) in which end users and crowd workers are invited to provide feedback on diverse concepts of the language.

The feedback collection interface in DynamicLabels is designed to collect a wide range of crowd feedback on a label set designed by the ML practitioner. DynamicLabels collects crowd annotations which can illustrate potential problems with the ML practitioner-built label set, such as confusion between labels or limited coverage of the label set. At the same time, by asking crowds to design their own label set, DynamicLabels collects diverse perspectives regarding the dataset.

2.3 Data-driven decision-making support

In order to make an informed decision based on crowd feedback, ML practitioners need to understand the feedback thoroughly. Much previous work has explored ways to present data in a way that users can easily comprehend and utilize. Voyant [9] automatically generates a word cloud with the collected feedback to make the feedback more helpful to users. Decipher [15] aggregates multiple feedback and provides a visualization tool to help the feedback interpretation process. Mudslide [16] helps teachers interpret the students' muddy points by visualizing students' feedback on lecture slides. Some studies have emphasized the importance of presenting a multi-faceted data analysis beyond aggregation. OpinionSeer [17] provides analysts with an interactive opinion visualization to easily explore the mined opinions. Kairam and Heer [18] used clustering techniques to leverage disagreement between crowd workers and showed that identified patterns could illustrate the worker characteristics as well as potential task problems.

Data-driven decision-making also enables users to consider various alternatives before making a decision. For conference session scheduling, Cobi [19] uses preference and constraints data on papers and sessions and presents the preview of changes in the number of conflicts for each move or assignment action that users consider. ConceptVector [20] supports an interactive construction of lexicon-based concepts by showing relevant documents and keywords regarding concepts the user considers. In designing a content-based image retrieval system for pathologists, Cai et al. [21] introduced tools that users can refine the image search by region, example, and concept.

The label set refinement interface of DynamicLabels presents varying levels of crowd feedback, ranging from raw crowd annotation to estimated coverage and confusion, so that users can consider diverse aspects of the label set simultaneously. Also, users can preview the consequence of each change before making a refinement and construct and compare multiple versions of the label set.

Chapter 3. Formative Study

To understand the current practice and challenges of ML practitioners in building and refining label sets in iteration-required situations, we conducted hour-long interviews with four ML practitioners. The participants had experience constructing label sets from scratch for multi-class classification models, as well as building training datasets and ML models. During the interview, we asked questions regarding their experience, the aspects they consider important in a label set, and the challenges and needs in constructing and refining label sets. After the interview, we used thematic coding to induct codes for each question. While our number of participants was relatively small (n=4), we were able to see repeatedly occurring themes.

3.1 Practice & Challenges

During the interview, all participants described label set building as one of the most challenging processes in constructing a training dataset, as the process involves coming up with an entirely new taxonomy on their own, including a set of clearly described labels as well as detailed descriptions of each label. Unlike constructing label sets in domains that already have an established label set or domain experts to rely on, they described the construction process in domains without external reference as "more ambiguous, less predictable, and time-consuming" (P3).

The participants described their practice of iteratively refining the label set as going through verify-refine cycles. In the beginning, the ML practitioners first sample a small proportion of the data to construct an initial version of the label set. Then, they sample a larger amount of data and use the data to annotate with the label set. By looking at the annotations made, they decide whether the current label set is clear and appropriate to construct label sets and models. In the earlier feedback loop, this is primarily done by the main person in charge, but with an increased number of sampled data, it is also common to ask other ML practitioners in the team or recruit annotators to verify the label set. When the feedback from the verification issue shows issues with the label set (e.g., too many incorrect annotations, mixed use of certain labels), the participants then make refinements to address the issues. They mentioned that this cycle continues until no critical issues are found in the label set, then proceeds to dataset building, which involves annotating the data.

The participants also emphasized the importance of building a robust label set that can prevent latent issues. The issues include wrong annotations made due to a misunderstanding of the label set, biased dataset construction, or even failed user interaction with the model. The issues often lead to poor quality of the model and bad user experience in the deployment stage. While some issues can be handled on the model side using existing techniques such as data augmentation, sometimes starting again from scratch is costly but inevitable. When the issue is detected too late and is impossible to begin from scratch, they sometimes adjust the model to fit the dataset due to the cost of the label set and dataset construction. Thus, they mentioned that having a robust label set is important for later stages. Some additional important aspects of the label set mentioned include adhering to the client's request, ensuring generalizability against many tasks, making clearly distinguished labels, being easily mutable, etc.

We also identified three major challenges ML practitioners face in the label set construction process, which we describe in detail below.

C1: Lack of helpful feedback to improve the label set

To improve their label set, it is common for ML practitioners to go through multiple feedback loops – by themselves, within a team of ML practitioners, or with a group of annotators. The most prevalent way to collect feedback on the label set is by trying out annotations using the constructed label set and spotting problematic data that cannot be covered or those that can be annotated using multiple labels. With the spotted data, the ML practitioner would make refinements to the label set until they are convinced that all uncaptured data can find a label and no data causes confusion.

While this approach helps in collecting problematic data and making refinements such as including edge cases or adding additional descriptions to the labels for exceptions, the participants mentioned the limited help that annotations can provide. When the annotation is done by the ML practitioner themselves or their team members, it is more difficult to spot uncovered or conflicting data as the annotations are made from similar perspectives and they know the labels and their meanings too well. P4 mentioned that "even if they go through multiple iterations within the team, there are always unexpected questions asked by the annotators."

To get a fresh perspective on the label set, ML practitioners sometimes recruit external annotators to verify it through annotation. This is more effective than having the team annotate as the problematic data are collected based on the annotator's perspective However, often the ML practitioners "end up adding a bunch of rule-based descriptions" (P2), which results in inefficiency and confusion for the annotators in making the training dataset later. P4 also mentioned that "they could understand that certain labels are misleading or conflicting through annotations, but sometimes are unsure of how to make the right refinements." In addition, recruiting external annotators to perform annotation can be a troublesome and costly process.

C2: Difficulty in comprehending meaningful insights from the feedback

As mentioned previously, constructing an optimal label set is difficult due to the many aspects (e.g., clear distinction of the labels, clear description of the labels) and multiple stakeholders (e.g., annotators, the requester of the model, users of the model) that need to be considered at the same time. While each ML practitioner has a set of criteria they consider important, there is no clear guideline on making an optimal label set, making it difficult for them to decide on the best label set.

This complex nature of label set construction inherently makes extracting meaningful insights from feedback difficult. When feedback about a label set is collected in the form of issues or annotation results, ML practitioners need to examine each piece of feedback and organize them to come up with a concrete revision item. However, it is challenging for them to both find critical feedback from a bunch of collected feedback and group them into a meaningful revision item. More specifically, P2 mentioned that "edge cases that lead to adding a description is relatively easy, while [those] that lead to a change in hierarchical structure and definition is very difficult to spot and decide." For conflicting labels, the participants mentioned that when annotators use a different label than their original intention, they know that something is confusing. However, it is difficult to decide whether the situation is common and whether changes should be made to the labels.

C3: Difficulty in utilizing the insights to make satisfactory changes

After ML practitioners organize key insights from the feedback, they need to apply the insights to modify the label set. Understanding the insights does not mean that a suitable modification can be made to the label set, as a complex set of criteria must be considered when making changes. As a result, ML practitioners are often not sure about their changes and their consequences.

ML practitioners try out approaches to increase certainty in the decision-making process such as discussing with a team of ML practitioners. However, this is largely inefficient in that it is time-consuming and difficult to reach a consensus due to the different opinions each ML practitioner holds. Even when a consensus is reached, there is no guarantee that the decision is optimal. The only way to check whether the decision is optimal or not is by getting to the later process of the ML model construction process (e.g., dataset construction, model training) and see if any issues occur. Participants struggled to choose the right moment to proceed to dataset construction, and mentioned that "[they are] afraid that the label set will end up creating issues later in the model building process" (P3).

3.2 Design Goals

Based on the interview results, we came up with the following design goals for a system that addresses the challenges ML practitioners face in iteratively building and refining label sets.

DG1: Collect helpful feedback on the label set from the crowd

ML practitioners identified a need for collecting nutritious feedback in their label set construction process to find problems in the label set and make appropriate refinements. Specifically, during the interview, the participants mentioned the need to receive feedback from fresh perspectives, that more actively suggest possible changes, and on a larger scale to address as many issues as possible. Through crowdsourcing, a group of people having fresh, diverse perspectives can be recruited to collect large-scale feedback on the label set. In addition, the crowd can also provide their own labels as suggestions to support the refinement process as well.

DG2: Provide multi-aspect analysis to derive meaningful insight

One major characteristic of label set construction is that there is no single set of criteria for an optimal label set. Because of that, ML practitioners face difficulty interpreting and obtaining meaningful insight from the feedback they collect during construction. During the interview, participants stated that they mainly get the sense of problematic labels by seeing edge cases and conflicting labels, which can also be presented to ML practitioners. Similarly, showing the collected feedback in multiple aspects (e.g., highlighting conflict, showing edge cases, providing summary) can support ML practitioners to thoroughly understand the feedback and extract the ones that are valuable to improving the label set.

DG3: Help understand possible changes and consequences in the label set

Even after extracting meaningful insights from the feedback, ML practitioners struggle to make confident changes to the label set due to the uncertainty of their action consequence. Actively supporting ML practitioners with possible label candidates or showing them the consequence of the label set with the changes will help the refinement process be more informed and confident, and will motivate them to quickly test and try out multiple versions of the label set.

Chapter 4. Proposed System: DynamicLabels

We present the design of DynamicLabels, a system that aims to support ML practitioners' label set construction. DynamicLabels supports iterative refinement of the label set through two separate interfaces: the **feedback collection interface** and the **label set refinement interface**. The former is provided to the crowd workers to collect annotations and label suggestions on the ML practitioner-built label set, and the latter is provided to the ML practitioners to refine their label set with multiple analyses of crowd feedback. The label set construction workflow and the role of each interface in DynamicLabels are described in Figure 4.1. In DynamicLabels, label sets are constructed in tree form (Refer to Fig. 6.2), consisting of labels and groups to group the labels.

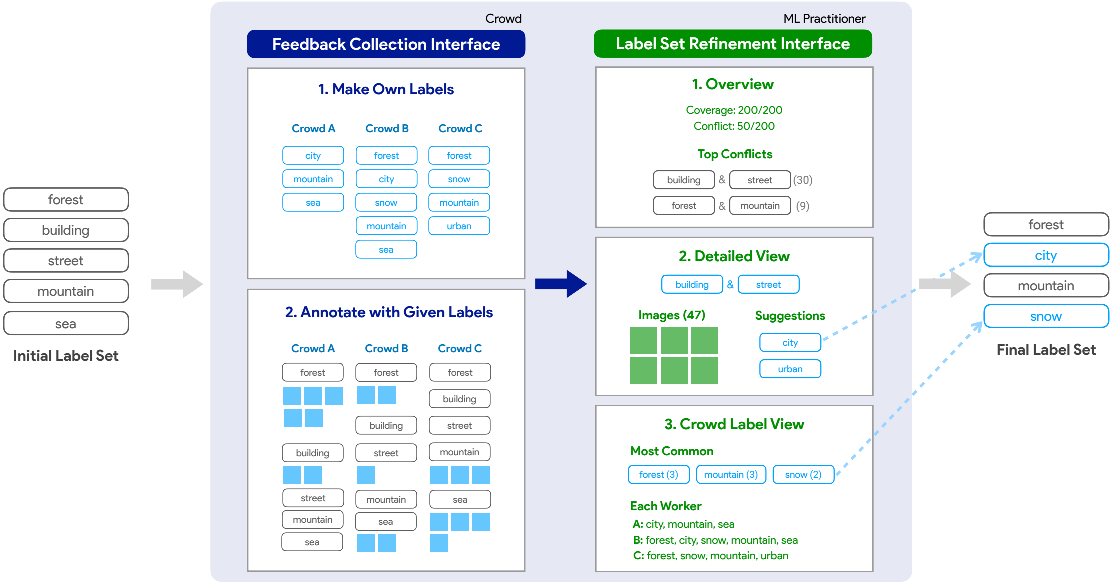


Figure 4.1: A label set construction workflow using DynamicLabels. The initial label set of the ML practitioner is given to crowd workers with the feedback collection interface. The collected feedback is presented to the ML practitioner with the label set refinement interface to refine their label set.

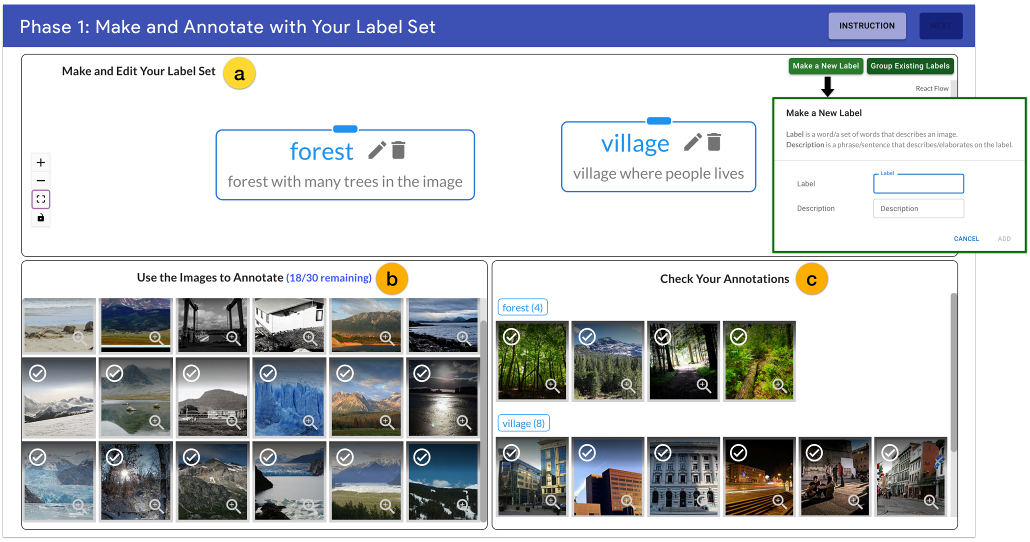
4.1 Feedback collection interface

For the crowd workers to use the feedback collection interface, the practitioner needs to have a constructed label set beforehand. This is similar to the standard practice in real-life settings, in that the practitioners first build an initial version label set before going through iterations.

In the feedback collection interface, the crowd is asked to provide feedback through two phases: (1) providing label suggestions by making the crowd's own label set and (2) annotating with the ML practitioner-built label set (Fig. 4.1-Feedback Collection Interface). We collect two types of feedback; passive (annotation results) and active (new label creation as suggestions), where an active suggestion is collected in addition to the annotations (the current practice in collecting feedback) to collect diverse perspectives on the label set. To prevent biases, we ask the crowd to build their own label set before annotating with the ML practitioner-built label set.

4.1.1 Phase 1 - Providing label suggestions by making the crowd's own label set

Crowd workers are asked to proceed to the Phase 1 task: creating their own label set (Fig. 4.2). The workers are first asked to take a look at 30 assigned images (Fig. 4.2-b) and come up with a set of labels (Fig. 4.2-a). Then, they are instructed to use the labels to make annotations (Fig. 4.2-c).


Phase 1: Make and Annotate with Your Label Set
INSTRUCTION
NEXT
Make and Edit Your Label Set
a
Make a New Label
Group Existing Labels
React Flow
forest
forest with many trees in the image
village
village where people lives
Make a New Label
Label
Label
Description
Description
CANCEL
ADD
Use the Images to Annotate (18/30 remaining)
b
Check Your Annotations
c
forest (4)
village (8)

Label is a word/a set of words that describes an image.
Description is a phrase/sentence that describes/elaborates on the label.

Figure 4.2: Phase 1 of the feedback collection interface. The crowd workers are instructed to check the assigned images through (b) a grid of images on the bottom left and make their own label set on the (a) top component by adding, revising, and deleting the labels. For created labels, they can select the images in (b) to annotate the images, which will show up in (c), under each label.

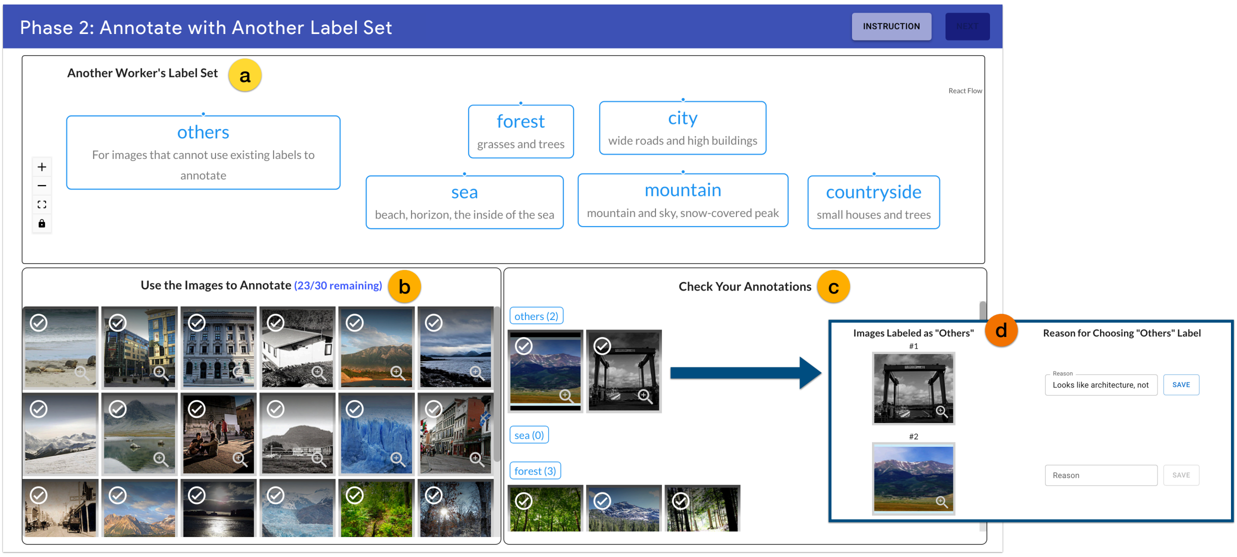
4.1.2 Phase 2 - Annotating with the ML practitioner-built label set

The crowd workers then proceed to the next phase and use ML practitioner-built label set to annotate the same 30 images (Fig. 4.3). In addition to the ML practitioner-built label set (Fig. 4.3-a), the workers are provided with an additional `others` label to annotate images that do not fit into the provided label set to spot edge cases. For each image labeled with `others`, the workers are asked to provide a brief reason (Fig. 4.3-d) to justify their choice.

4.1.3 Post-processing of crowd feedback

We post-processed the crowd annotations and crowd-made labels to extract meaningful information. First, to avoid suggesting redundant crowd-made labels, we merged multiple crowd-made labels into one if they are identical after stemming and lemmatizing.

As each crowd worker makes their own labels based on 30 images, the number of crowd annotations, or the number of images, for each crowd-made label is limited to 30 at most. To help ML practitioners better estimate the coverage and potential confusion of crowd-made labels, we established extended annotation for crowd-made labels. We first made similarity relationships between crowd-made labels


Phase 2: Annotate with Another Label Set
INSTRUCTION
Another Worker's Label Set
a
React Flow
others
For images that cannot use existing labels to annotate
forest
grasses and trees
city
wide roads and high buildings
sea
beach, horizon, the inside of the sea
mountain
mountain and sky, snow-covered peak
countryside
small houses and trees
+
−
Use the Images to Annotate (23/30 remaining)
b
Check Your Annotations
c
others (2)
sea (0)
forest (3)
Images Labeled as "Others"
d
Reason for Choosing "Others" Label
#1
#2
Reason
SAVE

Reason: Looks like architecture, not

Figure 4.3: Phase 2 of the feedback collection interface. The crowd workers are instructed to take a look at the (a) ML practitioner's label set and use the labels to annotate the (b) assigned images. Annotations will show up in (c), under each label. For images annotated using the "others" label, the workers are asked to provide a (d) brief reason each as an additional step.

and ML practitioner-made labels. For each crowd worker, we calculated the Jaccard similarity coefficient for each pair of a crowd-made label and an ML practitioner-made label, based on the crowd worker's annotation of 30 images for ML practitioner-made labels and crowd-made labels. For pairs with a similarity higher than 0.8, we assumed that the crowd label and the ML practitioner-made label are similar. Then, for each ML practitioner-made label, we filtered out images whose majority vote (of crowd annotation) match the label and established extended annotation between those images and crowd-made labels with high similarity with the ML practitioner-made label.

4.2 Label set refinement interface

When a sufficient amount of feedback is collected for each image, the ML practitioner can improve upon their initial label set through the label set refinement interface. The interface (Fig. 4.4) has the following components: Your label set, Overview, Detailed view, Crowd label view, and Saved label sets. The interface supports reviewing and understanding the crowd feedback by showing three analyses of the feedback (Fig. 4.4-b,c,d), and adopting the feedback to make necessary changes through crowd-made labels (Fig. 4.5).

4.2.1 Showing varying levels of analysis for the collected feedback

When the ML practitioner enters the label set refinement interface, they can find their initial label set on the top left, under "Your label set". Right next to it is an **overview** (Fig. 4.4-b) that shows a summary created with the crowd feedback. Inside the overview, we provide four metrics motivated by the formative study, (1) Coverage: number of images with annotation, (2) Conflict: number of images annotated with multiple labels, (3) Top conflicts: top 3 label pairs with the highest number of conflicts, and (4) Unlabeled images: number of images without annotation to spot the main issue in their label

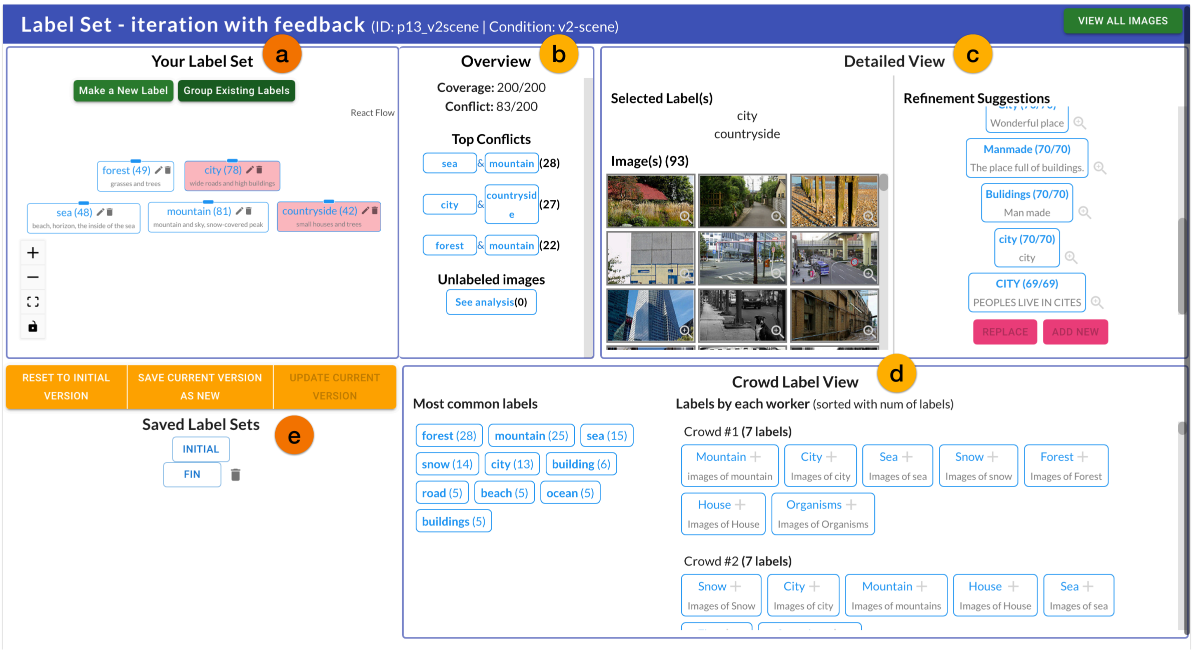


Figure 4.4: Overview of the label set refinement interface. The (a) current version label set is displayed at the top left, along with an (b) overview of the collected feedback. By clicking the labels in (a) or top conflicts and unlabeled images in (b), the ML practitioner can see a (c) detailed view. On the bottom right, you can see the (d) crowd label view. During the refinement, you can save different versions of the label sets, which are displayed through the (e) saved label sets.

set. The metrics are re-calculated when any changes are made to the label set. These metrics help ML practitioners understand how each label would be perceived and understood by the crowd and the coverage of labels as a set.

In addition, for the ML practitioner to understand the collected annotation in detail, the they can select label(s), top conflicts, or unlabeled images to see a **detailed view** (Fig. 4.4-c). Here, the ML practitioner can see images annotated with the selected label(s) (current label set), images annotated using the conflicting labels (top conflicts), or images that are not labeled (unlabeled images) on the left. On the right, they can see possible refinement suggestions – a list of crowd-made labels which overlaps the most with the selected set of images.

On the bottom right, the ML practitioner can explore refinement options, through an analysis of the crowd-made labels through the **crowd label view** (Fig. 4.4-d). DynamicLabels shows crowd-made labels in two different aspects: (1) Most common labels and (2) Labels by each worker. The Most common labels component shows the top 10 frequently-made crowd labels, and the Labels by each worker component show a list of crowd-made label sets, sorted by the number of labels made.

4.2.2 Providing refinement support with crowd-made labels

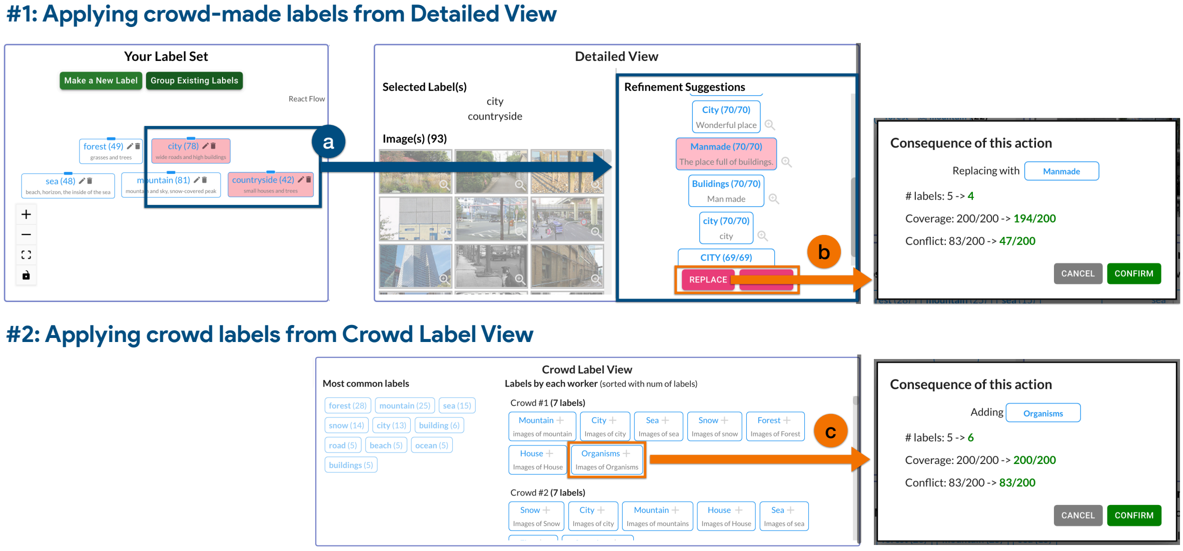
To support a more informed refinement, we allow ML practitioners to make additions or replacements to their label set using crowd-made labels. The refinement actions can take place from refinement suggestions or labels by each worker, which we illustrate in Figure 4.5.

If the ML practitioner wants to apply refinement suggestions, they can select the suggested label and choose to replace with or add the suggested label(s) to their current label set. They can also add the suggested labels for images with top conflicts or no labels.

On each refinement action, we display the **action consequence modal** (right of Fig. 4.5), where the change in the overview (the number of labels, coverage, conflict) is shown before making the change. After looking at the modal, the ML practitioner can decide whether to apply the refinement or not.

In addition to the detailed view, the ML practitioner can add crowd-made labels individually through the Crowd label view component (Fig. 4.4-d). The same action consequence modal is shown for this refinement as well.

The ML practitioner can also directly add new labels, edit existing labels, or delete existing labels (Fig. 4.4-a). For adding and editing labels, no changes are made to the overview as no crowd annotations are added to or removed.



Consequence of this action

Replacing with Manmade

labels: 5 -> 4

Coverage: 200/200 -> 194/200

Conflict: 83/200 -> 47/200

Consequence of this action

Adding Organisms

labels: 5 -> 6

Coverage: 200/200 -> 200/200

Conflict: 83/200 -> 83/200

Figure 4.5: Two possible ways to apply crowd-made labels to the current label set. In the top example, the ML practitioner can select (a) two labels in the current label set (city and countryside) to see a detailed view. From the refinement suggestions in the detailed view, the ML practitioner can (b) select crowd-made labels (Manmade) and click on the action (*replace*) to trigger the action consequence modal and make refinement decisions. In the bottom example, the ML practitioner can (c) click the plus icon next to the crowd-made label (Organisms) to add the label, which will trigger the action consequence modal.

4.2.3 Creating and exploring multiple label set candidates

On the bottom left corner, there is a **saved label sets** component, which enables version control of the label sets. With the buttons provided, the ML practitioner can go back to the initial version label set, save a new version label set, or update the current version label set.

Chapter 5. Evaluation

We conducted a 2-day study with 16 ML practitioners to investigate how DynamicLabels assists the ML label set construction process. We conducted a within-subjects study to minimize effects coming from ML expertise and label set/dataset construction experience.

Through the study, we aimed to answer the following research questions:

1. Can crowd workers produce helpful feedback with the feedback collection interface?
2. How do ML practitioners use crowd feedback to refine their label sets?
3. How do ML practitioners use DynamicLabels to make informed refinement decisions?

For the first question, we compare DynamicLabels of the collected feedback with that the baseline system (Described in Section 5.2.2). For the latter two questions, we compare and additionally explore how ML practitioners utilized crowd feedback and the refinement interface through our suggested system.

5.1 Participants

We recruited 16 participants by making an open call in several universities' online communities and social media targeting ML practitioners. Participation was limited to those with experience (1) manually constructing or utilizing label sets for ML models and (2) conducting industry or research projects using multi-class classification models, which require label sets with multiple labels.

Among 16 participants, 3 were undergraduate students, 7 were graduate students, and 6 were industry workers. Out of them, 14 out of 16 participants had experience manually building label sets, and 7 of them indicated that they had experience building label sets from scratch. Each participant was compensated KRW 120,000 ($94.00) for a total of 4.5-hour 2-day participation.

5.2 Study Setup

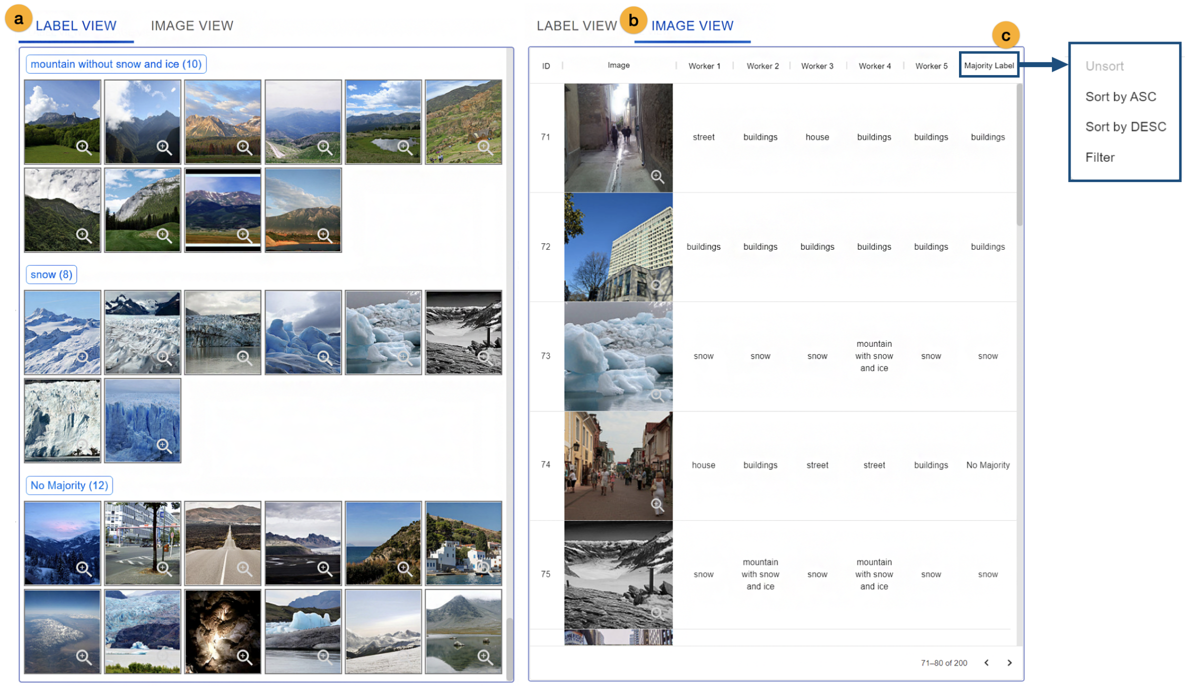
5.2.1 Task and Datasets

For the study, we asked participants to design a label set for a multi-class classification model. We selected two types of data: natural scene image dataset [22] and event flier dataset (manually collected by the authors). We refer to the natural scene image dataset as *scene* and the event flier dataset as *flier* from below. We chose datasets that do not require a high level of domain ML expertise to understand in order to ensure that the crowd can understand and provide quality feedback. We selected two datasets that vary in their modality, *scene* having images only, and *flier* having images and texts.

From each dataset, we randomly sampled 200 images for the study. Among the 200, we randomly selected and used 50 images for the initial label set construction. All 200 images were used in the feedback collection and refinement stage. The two numbers (50 and 200) were decided based on the common practice taken from the formative study, where practitioners mentioned that they would go through the first iteration with hundreds of data.

5.2.2 Baseline System

Our baseline system (Figure 5.1) is designed based on the verify-refine feedback loop that ML practitioners described during the formative study, where (1) crowd workers annotate each image to one of the ML practitioner-designed labels, and (2) the raw annotations and majority voting results are presented in the refinement phase as in Figure 5.1. We designed the baseline system to follow the common practice of verifying and refining label sets – through annotation – but made it more competitive as we collect larger-scale feedback through crowdsourcing compared to the standard practice. The main difference between DynamicLabels and the baseline system is (1) collecting active feedback through crowd-made labels and (2) supporting the understanding and refinements with feedback through analysis.



ID	Image	Worker 1	Worker 2	Worker 3	Worker 4	Worker 5	Majority Label
71		street	buildings	house	buildings	buildings	buildings
72		buildings	buildings	buildings	buildings	buildings	buildings
73		snow	snow	snow	mountain with snow and ice	snow	snow
74		house	buildings	street	street	buildings	No Majority
75		snow	mountain with snow and ice	snow	mountain with snow and ice	snow	snow

Figure 5.1: How the feedback was provided to the practitioners in the refinement interface of the baseline system. On the (a) label view, the user can see each label with images whose majority winner is the label and those without majority winners. On the (b) image view, the user can see raw annotations and majority voting results for each image. The (c) majority label column on the image view can be clicked to sort or filter results.

5.2.3 Task Procedure

The main task of the study was to construct a label set based on 200 images for each dataset. Each participant conducted the task using baseline condition for one dataset and DynamicLabels condition for the other dataset. The order and the image types assigned to the conditions were counterbalanced.

The study was conducted in two sessions to simulate a single iteration of the label set construction process. In the first session, the participants were asked to create an initial version of the label set with 50 images. Constructed label sets were used to collect crowd feedback (DynamicLabels) or crowd annotation (baseline) for 200 images. In the second session, the participants were asked to refine their label set with the crowd feedback (DynamicLabels) or the crowd annotations (baseline) collected. We describe the study procedure in Figure 5.2.

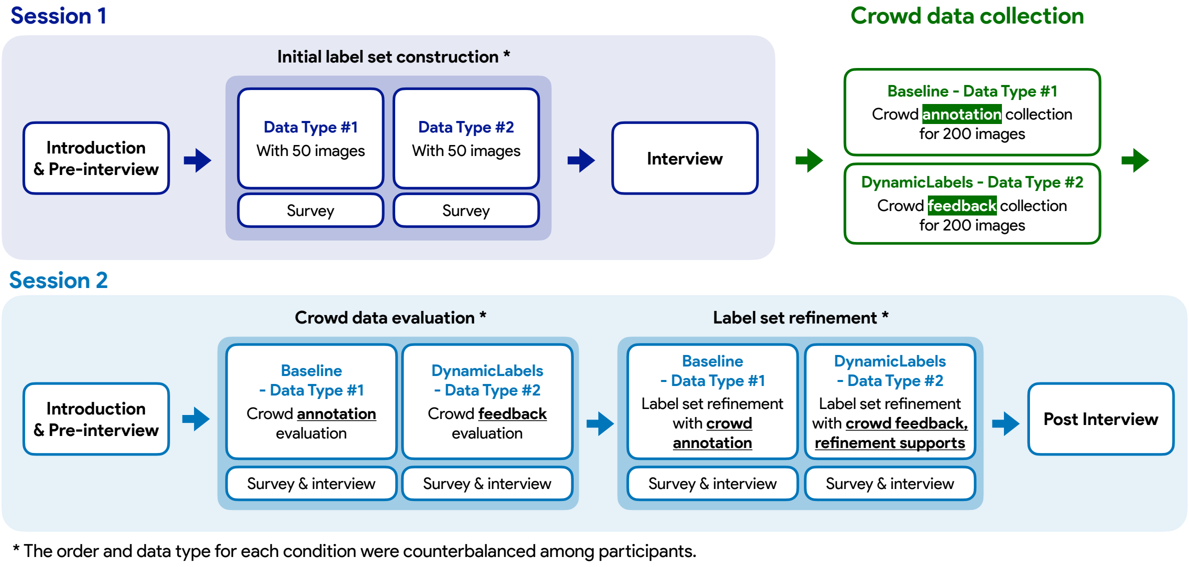

Session 1
Initial label set construction *
Introduction & Pre-interview
Data Type #1
With 50 images
Data Type #2
With 50 images
Survey
Survey
Interview
Crowd data collection
Baseline - Data Type #1
Crowd annotation collection for 200 images
DynamicLabels - Data Type #2
Crowd feedback collection for 200 images
Session 2
Crowd data evaluation *
Label set refinement *
Introduction & Pre-interview
Baseline - Data Type #1
Crowd annotation evaluation
DynamicLabels - Data Type #2
Crowd feedback evaluation
Survey & interview
Survey & interview
Baseline - Data Type #1
Label set refinement with crowd annotation
DynamicLabels - Data Type #2
Label set refinement with crowd feedback, refinement supports
Survey & interview
Survey & interview
Post Interview
* The order and data type for each condition were counterbalanced among participants.

Figure 5.2: Tasks and procedure for each session. In session 1, each participant creates two initial label sets for each dataset. The label sets are given to the crowd workers to collect annotation or feedback depending on the condition. In session 2, the participant refines their label set with the collected crowd data presented.

In **session 1**, the participants were first introduced to the background of the research and the study, then were asked to describe their experience and challenges in constructing label sets. Afterward, they were asked to construct two initial version label sets with 50 images based on an imaginary classification model in mind for each dataset. The label set construction was done in an interface we implemented, in which participants could make labels and groups in the form of nodes, similar to phase 1 of the feedback collection interface in Figure 4.2. After each label set construction, we asked the participants to fill out a 7-scale Likert scale survey regarding the construction process. After the participants finished constructing the two initial version label sets, we conducted a semi-structured interview regarding the construction process, the challenges in the process, and the participants' expectations of crowd feedback.

In between sessions 1 and 2, we collected crowd feedback (DynamicLabels) and annotations (baseline) on the label sets that participants constructed. The feedback and annotations were collected through Amazon Mechanical Turk [1]. For each label set, we recruited 34 crowd workers, and each image was annotated by five different workers. Each worker was assigned 30 images with a 24-image overlap with the previous worker. The workers were paid $8.0 per hour for their work. We limited participation to U.S. workers who had completed at least 1000 HITs with an approval rate of at least 97%.

In **session 2**, the participants were first asked to conduct a brief data evaluation for the crowd feedback and the annotations collected. They were instructed to look at the raw data of the collected feedback/annotation for about 5-10 minutes each and were asked to fill out a 7-scale Likert scale to evaluate the helpfulness of the collected data. Followed by an explanation of the refinement interface for either condition, the participants were asked to make refinements to their label set using 200 images and the collected feedback/annotations. After each refinement task, we asked the participants to fill out a 7-scale Likert scale survey regarding the refinement process. Then we conducted a semi-structured

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

interview with the participant regarding the overall refinement, how they utilized the collected crowd feedback/annotation, how they utilized the features in the refinement interface, and their opinions on the refinement process and the final label set. After completing the two refinement tasks, we conducted a structured interview with a focus on comparing the refinement process and the refinements they made using the DynamicLabels and the baseline system, and on their overall experience in utilizing the crowd feedback and the refinement interface to refine their label sets.

5.3 Measures

Throughout the study, we collected a wide range of data from both the participants and the crowd: crowd annotations, participants' session 1&2 label sets and refinement logs, task observations and interview responses, and survey results on crowd feedback/annotation and label set construction & refinement process. We describe how we analyzed crowd annotations, participants' refinement logs, and interview responses to answer our research questions.

Crowd annotations. We measure the collected crowd feedback in terms of diversity, helpfulness, and quality. We use the total and unique number of crowd-made labels to measure the diversity, and survey and interview responses on crowd feedback/annotation to measure the helpfulness.

As a quality measure, we measured the accuracy of crowd annotations. We first filtered out workers who showed clear trolling behavior. Out of 1,073 workers, 52 workers who used two or fewer ML practitioner-made labels to annotate 30 images or made labels that are out of context (e.g., making jacket or example1 in *flier*) were excluded. Then, we randomly sampled 2,000 crowd annotations for participant-made labels (500 for each dataset and condition) and 1,000 crowd annotations with crowd-made labels (for DynamicLabels condition, 500 for each dataset) among 47973 annotations in total. With 3,000 sampled annotations, two of the authors coded if each annotation was correct.

Refinement actions. We extracted each participant's refinement actions by analyzing the session logs and recordings. Action logs made from exploratory use of the system or correcting typos were excluded during the extraction. Then we categorized each refinement action into seven categories: three label changes (add, revise, delete), one description-level change, and three group-level changes (add, revise, delete).

There were cases that a set of multiple refinement actions were made to achieve one high-level refinement, such as split and merge. For example, P15 deleted the label education , added new labels job/career and books , and added new group education . The high-level goal of these actions was to split the label education into job/career and books . To analyze such high-level refinement actions made by participants, we grouped refinement actions made for one high-level split and merge refinement. Three of the authors analyzed three (out of 32) sessions together and then analyzed the remaining sessions individually.

We also coded whether each refinement action was made based on crowd-made labels. In addition to the cases in that participants directly used (added or replaced) crowd-made labels, we also marked down the cases in which participants adopted crowd-made labels or descriptions to make a new label.

Interview responses and session observations. Participants' think-aloud and interview responses were transcribed and analyzed through an open coding process, followed by focused coding. Two authors individually developed themes by open coding and then conducted focused coding to collapse or narrow down the developed themes.

Chapter 6. Results

We first present an overview of the study results and then discuss the result analysis of each RQ in detail. The overview presents descriptive statistics of the label sets that participants made in sessions 1 and 2, refinements made in session 2, and the crowd labels and annotations collected for each participant's session.

6.1 Descriptive Statistics

Label set construction and refinements

In session 1, the participants constructed an initial label set with 50 images. The median number of labels was 7 (min: 3, max: 11) for *scene* and 9.5 (min: 4, max: 14) for *flier*. The initial label set construction on average took 21.4 minutes (σ=6.8) for *scene* and 41.9 minutes (σ=16.4) for *flier*.

In session 2, the participants refined their label set using the crowd annotations (baseline) or feedback (DynamicLabels) collected with 200 images. Table 6.1 shows the median number of labels and groups in the initial and revised label sets, and the number of net changes made in the labels and groups for each condition and dataset. Participants generated more labels with *flier* dataset (median: 9.5 with min: 4, max: 14) than with *scene* dataset (median: 7 with min: 3, max: 11). However, within each dataset, there was no statistically significant difference in the number of generated labels, groups, and net changes in labels between conditions.

Participants spent significantly more time refining the label sets with DynamicLabels than baseline and with *event* than with *scene* (two-way repeated ANOVA, F=8.38 with p¡0.05 between conditions and F=4.84 with p¡0.05 between datasets). With *scene* dataset, participants spent 16.8 (σ=10.6) minutes for the baseline and 19.5 (σ=11.4) minutes for DynamicLabels to refine their label sets, respectively. For the *flier* dataset, the average time spent was 17.4 (σ=13.1) minutes for the baseline and 32.4 (σ=16.3) minutes for DynamicLabels.

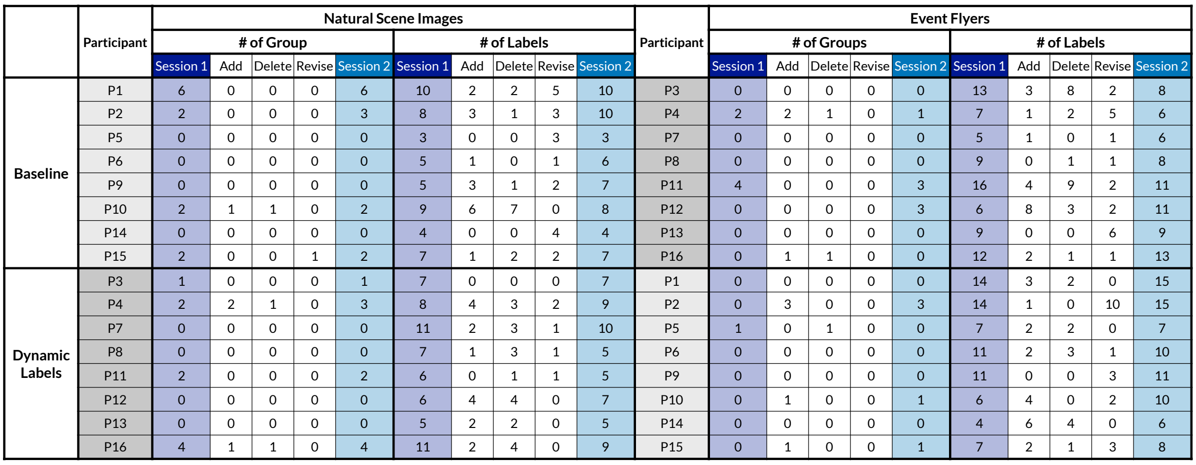
			Natural Scene					Event flier				
			Session 1	Changes (net)			Session 2	Session 1	Changes (net)			Session 2
				Added	Deleted	Revised			Added	Deleted	Revised	
Baseline	# of labels	Median	6	1.5	1	2.5	7	9	1.5	1.5	2	8.5
		[Min, Max]	[3, 10]	[0, 6]	[0, 7]	[0, 5]	[3, 10]	[5, 16]	[0, 8]	[0, 9]	[1, 6]	[6, 13]
	# of groups	Median	1	0	0	0	1	0	0	0	0	0
		[Min, Max]	[0, 6]	[0, 1]	[0, 1]	[0, 1]	[0, 6]	[0, 4]	[0, 3]	[0, 2]	[0, 2]	[0, 3]
DynamicLabels	# of labels	Median	7	2	3	0.5	7	9	2	1.5	1.5	10
		[Min, Max]	[5, 11]	[0, 4]	[0, 1]	[0, 4]	[5, 10]	[4, 14]	[0, 6]	[0, 4]	[0, 10]	[6, 15]
	# of groups	Median	0.5	0	0	0	0.5	0	0	0	0	0
		[Min, Max]	[0, 4]	[0, 2]	[0, 1]	[0, 0]	[0, 4]	[0, 1]	[0, 3]	[0, 1]	[0, 0]	[0, 3]

Table 6.1: Median number of labels and groups made in Sessions 1 and 2 and the median number of net changes in the label set, for each dataset and condition. Participants created label sets with more labels and a higher range in *flier* compared to *scene*, while there was no difference in the labels, groups, and net changes between DynamicLabels and baseline.

Table 6.1 shows the median number of refinement actions made by participants in each condition for each dataset. With the baseline, a median of 8.5 (min: 2, max: 24) and 9.5 (min: 2, max: 23) refinement actions were made by participants *scene* and *flier* datasets, respectively. With the DynamicLabels, participants made a median of 7 (min: 0, max: 25) and 11 (min: 5, max: 15) refinement actions for *scene* and *flier*, respectively. The specific refinement actions made by each participant under each condition are shown in Figure 6.1. Figure 6.2 shows an illustrative example of label set refinement made with DynamicLabels.

		Label			Description	Group			Total
		Add	Delete	Revise	Revise	Add	Delete	Revise	
Natural Scene	Baseline	1	0	2.5	3.5	0	0	0	8.5
		[0, 5]	[0, 6]	[0, 8]	[0, 9]	[0, 1]	[0, 1]	[0, 1]	[2, 24]
	DynamicLabels	2	2	1.5	0.5	0	0	0	7
		[0, 8]	[0, 9]	[0, 6]	[0, 2]	[0, 1]	[0, 1]	[0, 0]	[0, 25]
Event Flyer	Baseline	0.5	1.5	3	1	0	0	0	9.5
		[0, 8]	[0, 5]	[1, 9]	[0, 7]	[0, 3]	[0, 4]	[0, 0]	[2, 23]
	DynamicLabels	2.5	1.5	1	3	0	0	0	11
		[0, 5]	[0, 8]	[0, 5]	[0, 11]	[0, 1]	[0, 2]	[0, 0]	[5, 15]

Table 6.2: Median number of refinement actions made by participants in each condition for each dataset (Median [Min, Max]).



	Participant	Natural Scene Images										Participant	Event Flyers									
		# of Group					# of Labels						# of Groups					# of Labels				
		Session 1	Add	Delete	Revise	Session 2	Session 1	Add	Delete	Revise	Session 2		Session 1	Add	Delete	Revise	Session 2	Session 1	Add	Delete	Revise	Session 2
Baseline	P1	6	0	0	0	6	10	2	2	5	10	P3	0	0	0	0	0	13	3	8	2	8
	P2	2	0	0	0	3	8	3	1	3	10	P4	2	2	1	0	1	7	1	2	5	6
	P5	0	0	0	0	0	3	0	0	3	3	P7	0	0	0	0	0	5	1	0	1	6
	P6	0	0	0	0	0	5	1	0	1	6	P8	0	0	0	0	0	9	0	1	1	8
	P9	0	0	0	0	0	5	3	1	2	7	P11	4	0	0	0	3	16	4	9	2	11
	P10	2	1	1	0	2	9	6	7	0	8	P12	0	0	0	0	3	6	8	3	2	11
	P14	0	0	0	0	0	4	0	0	4	4	P13	0	0	0	0	0	9	0	0	6	9
	P15	2	0	0	1	2	7	1	2	2	7	P16	0	1	1	0	0	12	2	1	1	13
Dynamic Labels	P3	1	0	0	0	1	7	0	0	0	7	P1	0	0	0	0	0	14	3	2	0	15
	P4	2	2	1	0	3	8	4	3	2	9	P2	0	3	0	0	3	14	1	0	10	15
	P7	0	0	0	0	0	11	2	3	1	10	P5	1	0	1	0	0	7	2	2	0	7
	P8	0	0	0	0	0	7	1	3	1	5	P6	0	0	0	0	0	11	2	3	1	10
	P11	2	0	0	0	2	6	0	1	1	5	P9	0	0	0	0	0	11	0	0	3	11
	P12	0	0	0	0	0	6	4	4	0	7	P10	0	1	0	0	1	6	4	0	2	10
	P13	0	0	0	0	0	5	2	2	0	5	P14	0	0	0	0	0	4	6	4	0	6
	P16	4	1	1	0	4	11	2	4	0	9	P15	0	1	0	0	1	7	2	1	3	8

Figure 6.1: Number of groups and labels in sessions 1 and session 2 for each participant in each data type, and the number of groups and labels added/deleted/revised in between the sessions.

Crowd feedback

Between sessions 1 and 2, the crowd made annotations (baseline) or feedback – annotations and label suggestions (DynamicLabels) using the feedback collection interface. Table 6.1 summarizes the crowd feedback collected for a single label set. With the baseline system, 280 crowd workers made 8729 annotations for *scene* data and 272 crowd workers made 8190 annotations for *flier* data on average, using a single ML practitioner-made label set. In addition, for DynamicLabels the workers made 179.75 labels (72.13 unique labels) for *scene* and 199.25 labels (106.00 unique labels) for *flier* on average.

For the baseline task, the average time spent was 827.57 seconds (σ = 1044.30) in *scene* and 1770.96 seconds (σ = 1484.30) in *flier*. For the DynamicLabels task, the average time spent was 1269.08 seconds (σ = 1401.17) in *scene* and 2181.93 seconds (σ = 1679.29) in *flier*. While the phase 1 task in DynamicLabels can be more mentally demanding than the phase 2 task as workers have to come up with new

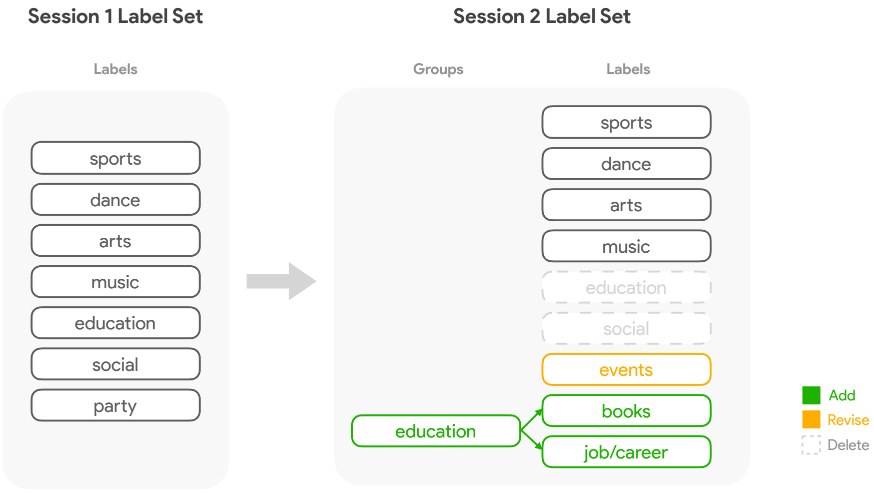


Figure 6.2: An illustrative example of how one participant's label set changed in sessions 1 and 2 (P15). Two labels (education, social) are deleted, a label is revised (from social to events), and two labels (books, job/career) are added and grouped with a newly added group (education)

labels on their own, the time spent is similar to or less than twice the time spent for the task in the baseline system according to Table 6.1. We presume that this was because the workers utilized the same set of images in phases 1 and 2 of DynamicLabels, decreasing the time needed to understand and become familiar with the data in phase 2.

	Natural Scene		Event flier	
	Baseline	**DynamicLabels**	**Baseline**	**DynamicLabels**
# of crowd labels	-	179.75 [143, 206]	-	199.25 [143, 229]
# of unique crowd labels	-	72.13 [47, 92]	-	106.00 [83, 137]
# of annotations with crowd-made labels	-	969.38 [770, 1026]	-	991.00 [813, 1031]
# of annotations with ML practitioner-made labels	1053.63 [990, 1294]	969.25 [784, 1024]	1023.75 [1021, 1030]	989.63 [812, 1028]

Table 6.3: Average number of crowd feedback collected per single label set (Mean [Min, Max]): number of crowd labels, number of unique crowd labels, number of annotations with crowd-made labels, number of annotations with ML practitioner-made labels, for each dataset and condition.

6.2 RQ1: Can the crowd produce helpful feedback with the feedback collection interface of DynamicLabels?

The crowd workers were able to annotate the images with moderate accuracy ranging between 69.18% and 90.52% and come up with diverse crowd-made labels. The participants perceived the crowd data collected through DynamicLabels were helpful in refining the label set compared to the baseline system. The collected annotations contained diverse perspective of the crowd, which was later utilized by the ML practitioners to spot potential confusions, and the collected crowd-made labels from each worker and a group of worker were both diverse, covering a wide range in coming up with both synonyms and possible labels within the domain (*scene* and *flier*).

Quality of Annotation

The accuracy of annotations with ML practitioner-made labels was 88.59% (*scene*), 69.18% (*flier*) with the baseline, and 90.52% (*scene*), 72.88% (*flier*) with DynamicLabels. The accuracy of annotations with crowd-made labels was 91.60% in *scene* and 73.40% in *flier*. Annotations in *flier* in both conditions had lower accuracy than those in *scene* which could be interpreted as *flier* had more information (text and illustrative visual objects) in a single image.

Diversity of Crowd-made Labels

A single crowd worker on average created 5.58 labels for *scene* and 6.18 labels for *flier*, summing up to on average 179.97 labels (*scene*) and 199.25 labels (*flier*) created per each label set created by participants. Crowd workers' diverse viewpoints were captured in the label sets they created. For example, one crowd worker made a label set consisting of night, building, hill, ocean, and road, while another crowd worker made a label set consisting of mountain, urban, coast, valley, ocean, forest, and rural when creating labels for *scene* dataset. After performing lemmatization on the crowd labels, the average number of unique labels was 72.13 in *scene* and 106.00 in *flier*.

In follow-up interviews regarding the crowd feedback, participants noted that the crowd-made labels helped them understand how crowd workers perceive and recognize the datasets in various ways. P15 commented that "through the crowd-made labels, I can see how people perceive and categorize the datasets, which I cannot understand through looking at the consequent annotations."

Perceived Helpfulness

All participants commented that the crowd feedback was meaningful and useful in making refinements to the label sets. P14 was able to see a difference in the crowd's perception and their own definition of the label by saying "I was able to see a difference in my understanding and the crowd's understanding of the label, through seeing that the annotation is different from what I expected. However, this will be helpful [to me in the refinement] as this tells me that my label is poorly defined." Participants stated that seeing the crowd-made labels for each image in DynamicLabels makes up for the difficult-to-understand annotations, functioning as reasons. For example, P12 was confused about why a running flier was not being annotated as activity rather than sports, but understood the reason by seeing exercise and yoga labels made by the workers.

Participants especially liked the various granularity of the crowd-made label sets in DynamicLabels. P1 perceived different workers' label sets as "an evolution of the label sets from the most general to the most specific." As participants examined the crowd-made labels, they were thinking about ways to apply the labels to their own label sets.

However, there also exist participants not satisfied with the quality of the annotations due to trolling behaviors of the workers. Also, some participants noted that having feedback on a larger scale (¿1000 data) would have been more helpful.

6.3 RQ2: How do ML practitioners use crowd feedback to refine their label sets?

The crowd feedback helped ML practitioners to (1) understand the crowd's general opinions and perspectives, (2) spot the weakness of their label sets, and (3) explore and apply the crowd labels to their label sets. (1) and (2) were also visible in baseline through the crowd annotations, while (3) was unique for DynamicLabels with the crowd-made labels.

Understanding the crowd's general opinions and perspectives

Participants mentioned that they could observe both converging and diverging opinions of the crowd in the crowd feedback. The collective opinions were visible through label suggestions or most common crowd-made labels, and the participants compared their label set with the crowds' labels to see the similarity of their labels to the general crowd's labels. P1 mentioned, "I was glad to observe the crowds' perspective of the task through the crowd-made labels, which was similar to my perspective." Further, P16 said that "looking at the most common labels helps to deal with the ambiguity in constructing a label set by oneself, and if there is a label that many crowd workers made, then that shows the necessity of that particular label."

In contrast, the participants also could observe the diverging opinions of the crowd in the collected annotations and labels. While there were overlapping crowd-made labels, participants found the labels to be overall diverging, which informed the participants how an image can be perceived differently by the workers. P13 said, "by looking at the crowd annotation results for labels and images I was unsure of in session 1, I am more confident that my label set should be defined more clearly."

Understanding the weakness of their label set

They were also able to realize the weaknesses of their label set through the feedback. This was mostly done by looking at the actual annotated images using their label set. P2 commented, "I would not have realized how poorly built my label set is without looking at the annotation results", and changed their label set entirely during session 2. A common weakness identified by participants was the lack of good label descriptions, found by looking at the detailed view of each label, and ambiguous boundaries between labels, found by looking at the top conflicts. P9 commented, in *flier*, that they "would not have known that yoga fliers could fit into the 'nature' category without crowd annotations." P9 modified the description of the 'nature' category to only include plant and forest during session 2.

Exploring alternative labels and applying perspectives of the crowd

Participants were also able to incorporate more perspectives from the crowd into their label sets. As DynamicLabels made crowd-made labels more visible throughout the entire refinement process, the participants were able to easily refer to the refinement suggestions, most common labels, or each worker's labels. A big portion of adding/revising refinements (*scene*: 35.1%, *flier*: 29.0%) in DynamicLabels utilized crowd-made labels. On the other hand, in baseline, no participants directly referenced crowd expressions (reasons for others) in adding/revising labels.

Participants were also able to make more satisfactory refinements by referring to the crowd-made labels in DynamicLabels. In baseline, participants eventually got a sense of the labels that needed to be revised by extracting summative information from the annotation results, but the next challenge they faced was in making satisfactory refinements. Participants struggled to come up with satisfactory label names, which led to more label name changes happening in the baseline system. For example, in the baseline, P10 revised a single label three times, from snow to snow/glacier without mountain, extreme cold with snow and glacier, then to extreme cold with snow, glaciers, and mountains and explicitly said that the crowd-made labels would have been helpful to decide the label name.

As a result, participants made more changes in label name in the baseline (with a median of 2.5 for *scene* 3 for *flier*) than in the DynamicLabels (with a median of 1.5 for *scene* and 1 for *event*) The difference between conditions is statistically significant (two-way ART ANOVA [23, 24], F=5.34 with p¡0.05).

6.4 RQ3: How do ML practitioners use DynamicLabels to make informed refinement decisions?

Throughout the study, we were able to observe the distinctive benefits of DynamicLabels over the baseline system in making more informed refinement decisions. When asked to compare the two conditions on how they helped their refinement decisions, most participants (13/16) rated DynamicLabels better than baseline. The results state that DynamicLabels supports (1) an understanding of the feedback at a high level through metrics, (2) a more flexible refinement, and (3) a structured refinement process. In addition, DynamicLabels (4) surfaces issues that might have been missed, and (5) supports ideation and grants assurance.

Metrics support understanding of and refinement from the feedback

The most frequently identified strength of DynamicLabels was the existence of the metrics (coverage, conflict) in the Overview component. The participants liked how the metrics summarized the collected annotations and described the metrics as an efficient and intuitive way to understand their label set without looking at the raw data. P1 mentioned that, in the baseline, they had to make much more judgments by themselves such as understanding the reason behind images with no majority winner, deciding whether to change the label or not by estimating the expected effect of the change, and verifying whether the changes can fix the issues by going through the images again.

Among the metrics, the participants particularly found the conflict metric helpful and many (10/16) aimed at reducing the number of conflicts during the refinement task. They complimented the intuitiveness of the metric, in that "it intuitively shows the labels that are controversial to the crowd (P5)" and utilized the metric to identify which refinements should be prioritized. When a particular label existed in all three top conflicts (e.g., career & socializing, career & volunteering, sports & career for P9 in *flier*), the participants began their refinement by clarifying and examining the feedback from that label (e.g., career).

Examining various refinement options

With DynamicLabels, participants were able to examine various refinement options before making a decision. With the consequence modal in DynamicLabels that shows the expected changes in the metrics for each refinement action, participants were able to examine each refinement action they considered before applying it. Participants found this consequence modal helpful, as knowing the expected change in the overview helped them make the decision more confidently. In addition, P13 mentioned that "the (action consequence) modal prevented them from making a wrong refinement choice.” When P13 tried to replace the label manmade for the conflict between the labels city and countryside, they saw the rise in the conflict and decided to take back the decision. They later merged the two labels into the label manmade. Even P3, who made no refinements with DynamicLabels, examined two refinements they considered but decided not to apply them.

The version control feature in DynamicLabels helped participants examine various label sets. Three out of eight participants in the *flier* made multiple versions and compared them before finalizing their refinement. P6 used version control to compare label sets before and after some major changes, and P10 made multiple versions of the label set refined with different purposes. No participant in *scene* used the version control feature, as the dataset was not complicated enough to have multiple label set alternates. However, when we asked the participants regarding the potential use of the version control, they said it would be useful when the dataset is more complex (P4) and when there exists high uncertainty in the decision process (P3).

Establishing a structured refinement process

The refinement process with DynamicLabels was perceived to be more structured than with the baseline. With DynamicLabels, participants began their refinement process from the overview, then looked into the detailed view for further understanding, and referred to the crowd-made labels whenever they needed more assurance or references when making decisions. Meanwhile, with the baseline, participants went back and forth between the label view and the image view until they identified the need for change. P6 noted that “in [v1], deciding on the starting or ending points was very challenging as I have to check the image and the annotations repeatedly to understand the outcome of the annotations.”. The participants found this implicitly conveyed workflow helpful, as they were able to “prioritize the refinement decisions (P15).” P4 also described DynamicLabels as supporting a more structured process that he could follow to figure out if it was the boundary of the label or the label name that needed to be changed.

Encouraging flexible refinement

Participants noted that having various forms of crowd feedback in DynamicLabels helped them understand the relationship between labels, such as potential conflict or inclusion among labels. During the refinement session, P5 said that ”by seeing the number of conflicts between expo and social and going through images with the conflict, I decided to split those labels into more [specific] ones”.

Participants also noted that with DynamicLabels, they could focus on how their label set represent the data, whereas they focused on clarifying each label and description in the baseline system. For example, P16 made three merge refinements (e.g., merging city street and buildings into city) with DynamicLabels whereas no high-level refinements were made with the baseline.

With both datasets, more participants made at least one high-level refinement in the DynamicLabels condition (6 out of 8) than in the baseline condition (3 out of 8). With both dataset, the median number of high-level refinements made by participants was 0 (min: 0, max: 3) in the baseline and 1 (min: 0, max: 5) in the DynamicLabels Table 6.4 summarizes the number of participants who made the split and merge refinement(s) in each condition and dataset.

	Natural Scene			Event Flyer		
	Split	Merge	Total	Split	Merge	Total
Baseline	2	1	3	1	3	3
DynamicLabels	2	5	6	3	4	6

Table 6.4: Number of participants who made high-level refinements in each condition for each dataset

Surfacing conflicts and edge cases that might have been ignored

In addition to the refinements made, many (n=7) participants were also able to spot possible conflicts and edge cases that they might have ignored. When refining with the baseline, most participants made refinements centered around the issues that they expected. P4 mentioned, "I checked that the labels that I assumed to be problematic actually had issues by looking at the annotations, and only revised those labels." However, when refining with DynamicLabels, participants identified unexpected conflicts, and were able to understand where the conflicts were coming from and make suitable changes. P6 removed the label Professional after seeing the label appearing in all three top conflicts. They commented "I did not expect Professional to be a controversial label. However, by looking into the conflicting images, I was able to understand the vagueness that the label [Professional] gives." P4 also captured edge cases and created more labels by looking at individual crowd-made labels, commenting that "the crowd helped in detecting edge cases in the 200 images." They added the labels cave and desert at the end of their refinement, after seeing images annotated with these labels and realizing their current label set couldn't cover them.

Chapter 7. Discussion

7.1 Potential use of DynamicLabels in different domains

Based on our observations with *scene* and *flier* datasets, we believe that DynamicLabels can be generally expanded to domains that do not require special expertise. Among them, we suggest a few domains where the benefit could be further amplified with references to the domains that participants suggested. For subjective domains where rules are decided based on collective human judgments (e.g., sentiment classification (P3)), the ML practitioner can effectively understand the general crowd's converging opinions and identify a convincing distinction between labels. For complicated domains where having a large number of labels are necessary (e.g., receipt information extraction (P15)), the practitioner can effectively use DynamicLabels to identify potential edge cases. If crowdsourcing with a group of experts is possible, the ML practitioner without the domain expertise can also utilize DynamicLabels to more effectively understand the data and construct the label set. For example, P7 mentioned that if they could crowdsource feedback with a group of graduate students, they want to try out label set building for topic classification of research papers.

During the study, different participants evaluated their final label set differently. Some participants identified the need for additional iterations, and others were confident about their label set and indicated that they are ready to move on to dataset construction.

7.2 Providing various forms of crowd feedback and giving more control to ML practitioners over them

In DynamicLabels, we present crowd feedback in various aspects through *overview*, *detailed view*, and *crowd label view*). In our study, ML practitioners flexibly utilized these features in combinations to meet their needs, which can change over the process, of understanding issues and the evidence behind them. At the same time, some participants expressed the need to see raw crowd feedback, such as raw annotations of each image and crowd-made labels and annotations before post-processing. Seeing that the participants made use of multiple features of their choice and were not reluctant to examine more data with increasing complexity, presenting them with a more dynamic version of analyses that covers a wide variety of crowd feedback will be useful in making confident refinements.

While the collected feedback was provided to the ML practitioners to explore, participants expressed the need for more direct control over the collected feedback to reflect in the analysis. For example, participants wanted to filter out trolling workers' feedback for the analysis, or give weights to the workers to explore how the conflict changes with more reliable workers' opinions valued. Inspired by Jury Learning's approach [25] in allowing the model builder to compose a group of juries and their opinions, DynamicLabels can give control to the ML practitioners so that they can explore and focus on a particular group's perspectives to construct the label set. In addition to having control over the workers and their data, participants also wished to have more control over the crowd annotations by fixing annotations they think are wrongly annotated or should be annotated to a single label, to indicate that particular labels or images have been examined. With this control, ML practitioners can make sure that all feedback is considered and applied in the process.

7.3 Designing a more human-centered model with the crowd

As the label set constitutes the primary structure of the model, aligning the user's mental model earlier in the label set construction stage can be effective in incorporating the potential user's mental model into the model. An interesting transition of the participants' behaviors we saw during the study was that they were considering and reflecting more on the crowd's opinions by looking at the crowd feedback, contrary to session 1 where they were more focused on creating a clear distinction among the labels for better model training.

We believe that DynamicLabels's impact can be enhanced far beyond building label sets with the crowd opinion, but further in incorporating the crowd's – or the real users' – opinions in the whole process of building machine learning applications. For example, in the dataset construction stage, the crowd can provide additional explanations or rationales for each annotation, which can be utilized to train the model to generate more human-like explanations or logic in a similar way to humans. In addition, crowd feedback can be used to collect large-scale opinions about the performance of large models, and to come up with human-centered metrics to evaluate those models. When crowd feedback is utilized in the later stages of model building (e.g., model building, model evaluation), we believe that the crowd can naturally learn about how the ML model functions while providing feedback, which can support a more natural human-AI interaction.

Chapter 8. Limitations and Future Work

We acknowledge several limitations of this work and discuss possible future work. While we applied quality control methods by providing the crowd workers with tutorial tasks or warnings regarding poor work, the accuracy of the collected annotations was lower than the ML practitioners' expectations. Seeing the lower quality of feedback, some participants lost trust in the crowd feedback — which led to the participants intentionally choose not to utilize the features in the label set refinement interface, or not making any refinements based on crowd feedback. We believe that additional quality control methods such as giving the crowd a task with ground truth in the middle of the task to check the workers' attention could improve the quality of the crowd feedback.

To measure the effect of crowd feedback and the label set refinement interface, the study was carried out in the form of a comparison study, with many factors such as the number of images to look at and the scale of crowd feedback controlled throughout the study. Since different participants exhibited different patterns in incorporating crowd feedback into their label sets, we wish to observe the long-term effects of DynamicLabels by deploying the system in a real-world setting.

Chapter 9. Conclusion

In this paper, we present DynamicLabels, a system that supports the process of label set construction by collecting crowd feedback and showing analysis of the crowd feedback. Our study with 16 participants shows that DynamicLabels enables a more exploratory, flexible, and structured refinement process with fine-grained analysis and crowd-made labels. The crowd feedback helped the participants understand the general crowd's opinions as well as the weaknesses of their label set, and utilized the crowd-made labels to make refinements. DynamicLabels suggests a new approach to building label sets for machine learning models, by incorporating the crowd's–or the general user's–feedback to reflect the user's diverse opinions and perspectives.

Bibliography

[1] Creating label sets. https://cloud.google.com/ai-platform/data-labeling/docs/label-sets. Accessed: 2023-01-15.

[2] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.

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

[4] Guorong Xuan, Wei Zhang, and Peiqi Chai. Em algorithms of gaussian mixture model and hidden markov model. In *Proceedings 2001 international conference on image processing (Cat. No. 01CH37205)*, volume 1, pages 145–148. IEEE, 2001.

[5] Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1999–2008, New York, NY, USA, 2013. Association for Computing Machinery.

[6] Jonathan Bragg, Daniel S Weld, et al. Crowdsourcing multi-label classification for taxonomy creation. In *First AAAI conference on human computation and crowdsourcing*, 2013.

[7] Joseph Chee Chang, Aniket Kittur, and Nathan Hahn. Alloy: Clustering with crowds and computation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3180–3191, 2016.

[8] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2334–2346, 2017.

[9] Anbang Xu, Shih-Wen Huang, and Brian Bailey. Voyant: generating structured feedback on visual designs using a crowd of non-experts. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 1433–1444, 2014.

[10] Yoonseo Choi, Toni-Jan Keith Palma Monserrat, Jeongeon Park, Hyungyu Shin, Nyoungwoo Lee, and Juho Kim. Protochat: Supporting the conversation design process with crowd feedback. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–27, 2021.

[11] Kurt Luther, Jari-Lee Tolentino, Wei Wu, Amy Pavel, Brian P Bailey, Maneesh Agrawala, Björn Hartmann, and Steven P Dow. Structuring, aggregating, and evaluating crowdsourced design critique. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 473–485, 2015.

[12] Selina Sutton and Shaun Lawson. A provocation for rethinking and democratising emoji design. In *Proceedings of the 2017 ACM Conference Companion Publication on Designing Interactive Systems*, pages 7–12, 2017.

[13] Marco Brambilla, Jordi Cabot, Javier Luis Cánovas Izquierdo, and Andrea Mauri. Better call the crowd: using crowdsourcing to shape the notation of domain-specific languages. In *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*, pages 129–138, 2017.

[14] Kosa Goucher-Lambert and Jonathan Cagan. Crowdsourcing inspiration: Using crowd generated inspirational stimuli to support designer ideation. *Design Studies*, 61:1–29, 2019.

[15] Yu-Chun Grace Yen, Joy O Kim, and Brian P Bailey. Decipher: an interactive visualization tool for interpreting unstructured design feedback from multiple providers. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[16] Elena L Glassman, Juho Kim, Andrés Monroy-Hernández, and Meredith Ringel Morris. Mudslide: A spatially anchored census of student confusion for online lecture videos. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1555–1564, 2015.

[17] Yingcai Wu, Furu Wei, Shixia Liu, Norman Au, Weiwei Cui, Hong Zhou, and Huamin Qu. Opinionseer: interactive visualization of hotel customer feedback. *IEEE transactions on visualization and computer graphics*, 16(6):1109–1118, 2010.

[18] Sanjay Kairam and Jeffrey Heer. Parting crowds: Characterizing divergent interpretations in crowdsourced annotation tasks. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1637–1648, 2016.

[19] Juho Kim, Haoqi Zhang, Paul André, Lydia B Chilton, Wendy Mackay, Michel Beaudouin-Lafon, Robert C Miller, and Steven P Dow. Cobi: A community-informed conference scheduling tool. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 173–182, 2013.

[20] Deokgun Park, Seungyeon Kim, Jurim Lee, Jaegul Choo, Nicholas Diakopoulos, and Niklas Elmqvist. Conceptvector: text visual analytics via interactive lexicon building using word embedding. *IEEE transactions on visualization and computer graphics*, 24(1):361–370, 2017.

[21] Carrie J Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S Corrado, Martin C Stumpe, et al. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–14, 2019.

[22] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[23] Lisa A Elkin, Matthew Kay, James J Higgins, and Jacob O Wobbrock. An aligned rank transform procedure for multifactor contrast tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 754–768, 2021.

[24] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 143–146, 2011.

[25] Mitchell L Gordon, Michelle S Lam, Joon Sung Park, Kayur Patel, Jeff Hancock, Tatsunori Hashimoto, and Michael S Bernstein. Jury learning: Integrating dissenting voices into machine learning models. In *CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022.

Acknowledgment

I would like to express my gratitude to the friends, collaborators, and family members who have helped me throughout my M.S. journey. This would not have been possible without their support.

- First of all, I thank Prof. Juho Kim for introducing me to the field of HCI research. Taking your Intro to HCI course back in 2019 has been one of the best choices I have made in my life. Working with you, I have grown not only as a researcher but also developed a professional work ethic that will benefit me in the future.
- I thank my collaborators, Eun-Young and Yeon Su, for being my motivators during the DynamicLabels project. The numerous discussions we had throughout the project led to a meaningful submission to the community. Special thanks to Hobak, who has remained patient and adorable throughout. :)
- I thank the wonderful members of KIXLAB for being the best labmates I can ever ask for. I will cherish all the meaningful research discussions we had, as well as the funniest jokes on Slack. I want to express special thanks to: my mentor and lifelong friend Yoonseo; my first-ever and still best friend Juhoon; my M.S. friends DaEun, Seulgi, and Yoonsu; and my sig-meme friends Hyoungwook and Kihoon. I will miss each and every one of you.
- I thank Mina and Haesoo for going through the first first-author submission process together. You two have always inspired me to pursue valuable research.
- I thank my friends Youngil, Sejoon, Heeju, Hyerim, Sangkyung, and Changsun for being my go-to whenever I needed a break from research. I also thank the owners of Dubal-nebal and Fall in the Pool for serving the best soju, coffee, and vegan desserts I could ask for.
- Last but not least, I thank my family – Byung Hong Park, Jinhee Lee, and Changhwi Park – for their endless support and love. I feel extremely blessed to have a family who is so supportive of my passions.

Curriculum Vitae in Korean

이름:	박 정 언
생년월일:	1998년 02월 19일
전자주소:	jeongeon.park@kaist.ac.kr

학력

2013. 8. – 2016. 6.	McLean High School
2016. 9. – 2021. 8.	한국과학기술원 전산학부 (학사)
2021. 8. – 2023. 8.	한국과학기술원 전산학부 (석사)

경력

2021. 8. – 2023. 8.	한국과학기술원 전산학부 조교

연구업적

1. **Jeongeon Park**, Eun-Yeong Ko, Yeon Su Park, Jinyeong Yim, and Juho Kim, "DynamicLabels: Supporting Informed Construction of Machine Learning Label Sets with Crowd Feedback," under review.

2. **Jeongeon Park**, Eun-Yeong Ko, Donghoon Han, Jinyeong Yim, and Juho Kim, "DynamicLabels: Supporting Dynamic Construction of Datasets with Annotator Suggestions," *In Work-in-Progress of the 9th AAAI Conference on Human Computation and Crowdsourcing (HCOMP WiP '21),* ACM.

3. Yoonseo Choi, Toni-Jan Keith Monserrat, **Jeongeon Park**, Hyungyu Shin, Nyoungwoo Lee, and Juho Kim, "ProtoChat: Supporting the Conversation Design Process with Crowd Feedback," *Proceedings of the ACM on Human-Computer Interaction 4 (CSCW3) (CSCW '20),* ACM.

4. Yoonseo Choi, Hyungyu Shin, Toni-Jan Keith Monserrat, Nyoungwoo Lee, **Jeongeon Park** and Juho Kim, "Supporting an Iterative Conversation Design Process," *In Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20),* ACM.