Learning with Different Amounts of Annotation: From Zero to Many Labels

Shujian Zhang **Chengyue Gong** **Eunsol Choi**
The University of Texas at Austin
szhang19@utexas.edu, {cygong,eunsol}@cs.utexas.edu

Abstract

Training NLP systems typically assumes access to annotated data that has a single human label per example. Given imperfect labeling from annotators and inherent ambiguity of language, we hypothesize that single label is not sufficient to learn the spectrum of language interpretation. We explore new annotation distribution schemes, assigning multiple labels per example for a small subset of training examples. Introducing such multi label examples at the cost of annotating fewer examples brings clear gains on natural language inference task and entity typing task, even when we simply first train with a single label data and then fine tune with multi label examples. Extending a MixUp data augmentation framework, we propose a learning algorithm that can learn from training examples with different amount of annotation (with zero, one, or multiple labels). This algorithm efficiently combines signals from uneven training data and brings additional gains in low annotation budget and cross domain settings. Together, our method achieves consistent gains in two tasks, suggesting distributing labels unevenly among training examples can be beneficial for many NLP tasks.[1]

1 Introduction

Crowdsourcing annotations (Rajpurkar et al., 2016; Bowman et al., 2015) has become a common practice for developing natural language processing benchmark datasets. Even after thorough quality control, it is often infeasible to reach complete annotator agreement, as annotators make mistakes (Freitag et al., 2021) and ambiguity is a key feature of human communication (Asher and Lascarides, 2005). Rich prior works (Passonneau et al., 2012; Pavlick and Kwiatkowski, 2019; Nie et al., 2020; Min et al., 2020; Ferracane et al., 2021) show that disagreement among annotators is not an annotation artifact but rather core linguistic phenomena.

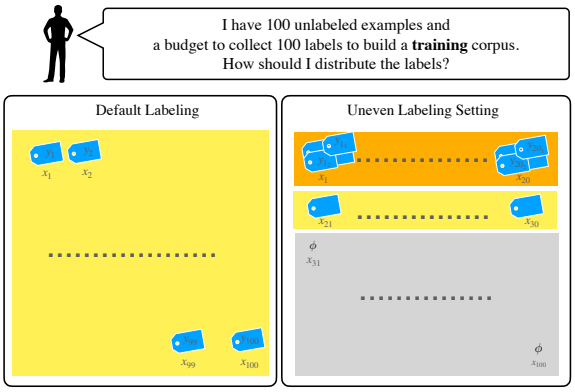

Figure 1: Re-thinking how to distribute annotation budget. Each blue tag represents a human annotation for the corresponding x. Examples in the orange shaded area are assigned many labels (multi label data), examples in the yellow shaded area are assigned a single label (single label data), and examples in grey shaded area are not assigned any labels. Models trained on a combination of multi label, single label and unlabeled data outperform models trained on single label data on both label accuracy and label distribution metrics for entailment and entity typing task.

Despite observing such inherent ambiguity, most work have not embraced ambiguity into the training procedure. Most existing datasets (Wang et al., 2019; Rajpurkar et al., 2016) present a single label per each training example while collecting multiple labels for examples in the evaluation set, with a few notable exceptions on subjective tasks (Passonneau et al., 2012; Ferracane et al., 2021). We challenge this paradigm and re-distribute annotation budget unevenly among training examples, generating small amount of training examples with multiple labels. Without changing mainstream model architectures (Vaswani et al., 2017), we change the annotation budget allocation. Figure 1 visualizes the standard scheme to our new label distribution

[1]Code and data split is available at https://github.com/szhang42/Uneven_training_data.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7620–7632
November 7–11, 2021.

scheme.

Under our uneven label distribution scheme, models are given a mixture of single label, multi label and unlabeled examples as a training corpus. How should we combine learning signals from distinct types of training examples? We explore simply combining and shuffling examples, upsampling multi label examples, and curriculum learning. Then, we introduce an algorithm based on recent data augmentation MixUp (Zhang et al., 2018) which generates virtual training examples by interpolating between different training examples.

We present a retrospective study (Liu et al., 2021) with datasets from prior work (Nie et al., 2020; Choi et al., 2018). We first evaluate our approach on densely annotated NLI datasets, where human disagreement is prevalent (Pavlick and Kwiatkowski, 2019). We report majority label accuracy and distribution metrics (e.g., KL divergence to measures models' ability to estimate human label distribution). Our experiment on a multi label task – fine-grained entity typing (Choi et al., 2018) – exhibits similar trend that acquiring multiple labels for a single example is more effective than labeling as many examples as possible.

Lastly, we present an in-depth study comparing models trained with multi label data and models trained with single label data. Training with single label examples leads the low entropy label distribution and unable to capture human disagreements. While calibration techniques such as smoothing distribution (Guo et al., 2018) can alleviate over confidence of model prediction and improves distributional metrics, it erroneously introduces uncertainty even for unambiguous examples. Our study suggests that introducing uneven label distribution scheme, paired with a learning architecture that combines three different types of training examples, can provide an efficient and effective solution.

2 Data Configuration

We first describe our training data configuration and then discuss our learning algorithms. We notate the input feature vector as x and output label distribution as y. We have three types of training example: unlabeled data set $\mathbf{X}_u = \{x_u^1, x_u^2 \ldots, x_u^{u_n})\}$, where u_n is the total number of unlabeled examples, single label data set $\mathbf{X}_s = \{(x_s^1, y_s^1), (x_s^2, y_s^2) \ldots, (x_s^{s_n}, y_s^{s_n})\}$ where s_n is the total number of single label examples, and multi label data set

$$\mathbf{X}_m = \{(x_m^1, (y_{m1}^1, y_{m_2}^1 \ldots y_{m_k}^1)) \ldots, (x_m^{m_n}, (y_{m_1}^{m_n}, y_{m_2}^{m_n} \ldots y_{m_k}^{m_n}))\},$$

where m_n is the total number of multi label examples and k is the number of annotations per example. For multi label examples, we will aggregate multiple annotations to generate y_m^*. Unlike y_s, which is a one-hot vector, y_m^* will now be a distribution over labels (for label distribution estimation problem, averaging $(y_{m_1}^i, y_{m_2}^i \ldots y_{m_k}^i)$, and for label prediction problem, taking $\arg\max_k(y_{m_1}^i, y_{m_2}^i \ldots y_{m_k}^i)$).

The annotation cost for generating training datasets can be described as the function of two factors (Sheng et al., 2008): the number of examples and the number of labels. Both can have impacts on the model performance and are highly associated with the annotation cost. In most existing studies (Wang et al., 2019), the training data is a set of annotated example with single label, $\mathbf{X}_s$. Supervised learning assumes an access to $\mathbf{X}_s$, and unsupervised learning assumes additional unlabeled examples $\mathbf{X}_u$, and semi-supervised learning assumes a mixture of $\mathbf{X}_u$ and $\mathbf{X}_s$. Here, we focus on annotation distribution over examples and make a simplifying assumption that annotation cost scales linearly to the number of labels.

We propose a set up where we distribute annotation label budget **unevenly** across training examples, resulting in unlabeled examples, single label examples, and multi label examples. We do not collect any new annotations in this work, and re-use dataset from prior work (Choi et al., 2018; Chen et al., 2020b) by resplitting existing datasets to simulate different label distribution scenarios. For each task, we study $\mathbf{X}_s$ setting, which consider a fixed number of supervised, single label example. Then, we introduce $\mathbf{X}_s$ + $\mathbf{X}_m$ setting, which includes multi label examples and single label examples (but fixing the amount of total annotation same as the $\mathbf{X}_s$ setting). Lastly, we study adding unlabeled examples $\mathbf{X}_u$ to both settings.

2.1 Task

We consider two classification tasks, Natural Language Inference (NLI) and fine-grained entity typing. Recent papers (Pavlick and Kwiatkowski, 2019; Nie et al., 2020) have shown that human annotators disagree on NLI task for its inherent ambiguity. Such disagreement is not an annota-

Premise	Hypothesis	Old Labels	New Labels
A woman in a tan top and jeans is sitting on a bench wearing headphones.	A woman is listening to music.	E E N N E	N (93) E (7)

Sentence with Target Entity	Entity Type Labels
During the Inca Empire, {the Inti Raymi} was the most important of four ceremonies celebrated in Cusco.	event, festival, ritual, custom, ceremony, party, celebration

Table 1: Examples of ChaosSNLI and Ultra-fine Entity Typing dataset. In NLI task, each label corresponds to one annotator's judgement (entailment (E) / neutral (N) / contradiction (C)). In fine-grained entity typing, the entity mention is in blue with the curly brackets. Each positive type label is treated a single label.

Task	Data Setup	# Single	# Multi	# Unlabel	Total # Labels	Total # Examples
Chaos S / MNLI	Original	549k / 392k	0	0	549k / 392k	549k / 392k
	$\mathbf{X}_s$	150k	0	0	150k * 1 = 150k	150k
	$\mathbf{X}_s + \mathbf{X}_m$	145k	0.5k	0	145k * 1 + 0.5k * 10 = 150k	145.5k
	$\mathbf{X}_s + \mathbf{X}_u$	150k	0	549k-150k	150k * 1 = 150k	549k
	$\mathbf{X}_s + \mathbf{X}_m + \mathbf{X}_u$	145k	0.5k	549k-145.5k	145k * 1 + 0.5k * 10 = 150k	549k
UFET	Original	151	1768	0	10.3k	1919
	$\mathbf{X}_s$	500	0	0	500 * 1 = 500	500
	$\mathbf{X}_s + \mathbf{X}_m$	100	200	0	100 * 1 + 200 * 2 = 500	300
	$\mathbf{X}_s + \mathbf{X}_u$	500	0	1919 - 500	500 * 1 = 500	1919
	$\mathbf{X}_s + \mathbf{X}_m + \mathbf{X}_u$	100	200	1919 - 300	100 * 1 + 200 * 2 = 500	1919

Table 2: Training data configurations. Each configuration is characterized by the number of labels and the number of examples. The number of labels are consistent in all settings. In NLI task, each multi label example contains 10 labels, and in UFET task, each multi label example contains 2 labels. For completeness, we also provide original training data configurations.

tion artifact but rather exhibits the judgement of annotators with differing interpretations of entailment (Reidsma and op den Akker, 2008).

Named entity recognition (Sang and Meulder, 2003), in its vanilla setting with a handful of classes, is a straightforwad task with high inter-annotator agreement. However, when the label set grows, comprehensive annotation becomes challenging and most distant supervision examples only offers partial labels. Many real world tasks (Bhatia et al., 2016) involve such complex large label space, where comprehensively annotating examples are often infeasible. We choose ultra-fine entity typing dataset (Choi et al., 2018) which provides typing into a rich ontology consisting of over 10K label candidates. Unlike NLI task, fine grained entity typing is a multi class classification task, where a single example is assigned to a **set of gold type labels**. Thus, acquiring multiple labels for the same example provides correlation among the labels (e.g., musicians are also artists).

Table 1 shows an example of each task, and Table 2 shows full experimental data configuration, which will be explained below.

NLI: Label Distribution Estimation NLI is a task (Dagan et al., 2005; Bowman et al., 2015) that involves deciding whether a hypothesis h is supported by a given premise p. It is a three-way classification task with "entailment", "contradiction", and "neutral" as labels, and recently reframed as a human label distribution prediction task.

We use the training data from the original SNLI (Bowman et al., 2015) and MNLI dataset (Williams et al., 2018), containing 549K and 392K instances respectively. Recent work presents ChaosNLI dataset (Nie et al., 2020), which collects 100 labels per example in the original SNLI/MNLI development set, (1,514 examples for SNLI, 1,599 examples for MNLI).[2]

For multi label data, we use ChaosNLI dataset to sample multi-annotation examples for SNLI and MNLI. We randomly sampled 500 examples from ChasSNLI and ChasMNLI respectively for evaluation set and use the rest of ChaosNLI for training.[3]

[2]It covers SNLI, MNLI, and αNLI (Bhagavatula et al., 2020), and we focus our study on the first two datasets as they show more disagreement among the annotators.

[3]The original datasets split data such that premise does not occur in both train and evaluation set. This random re-partition

For ChaosNLI in the training, We randomly sample 10 out of 100 annotations for each examples in the training set. For single label data, we directly sample from the original SNLI/MNLI data based on the annotation budget such as 150k or 6k examples.

Ultra Fine Entity Typing (UFET): Multi Label Classification UFET takes a sentence and an entity mention, and labels this mention with a set of entity types from the rich type ontology covering 10K types. Each example is annotated with average 5 labels: 0.9 general types , 0.6 fine-grained types , and 3.9 ultra-fine types. We consider each positive **type** annotation as a single label, thus original data setting is a combination of $\mathbf{X}_s$ and $\mathbf{X}_m$ examples (most of them are $\mathbf{X}_m$). We simulate $\mathbf{X}_s$ setting and $\mathbf{X}_s$ + $\mathbf{X}_m$ setting for our study.

The dataset consists of 6K crowd-sourced examples, randomly split evenly into train, development, and test sets. We fix the total number of training label budget as 500 labels. For $\mathbf{X}_s$ setting, we randomly sample 500 examples and sample one label for each example. For $\mathbf{X}_s$ + $\mathbf{X}_m$ setting, we sample 100 examples with one label, and 200 examples with two labels. We only modify training data and use the original evaluation dataset.

3 Learning

We introduce learning algorithms that can handle different types of training data. We describe feature extractors for both tasks, which maps natural language to a dense vector representation x then discuss learning algorithms. In the learning algorithms, we first discuss learning with annotated examples only (single label and multi label) and describe learning strategy to integrate unlabeled data. All learning configurations are optimized with the cross entropy (CE) loss.

3.1 Base Model

We present base models at here which is used to derive input feature vector x from natural language examples. Training details and hyperparameter settings can be found in the appendix.

NLI We use RoBERTa (Liu et al., 2019) based classification model, i.e., encoding concatenated hypothesis and premise and pass the resulting [CLS] representation through a fully connected layer to predict the label distribution.

UFET We follow the baseline architecture presented in Choi et al. (2018), a bidirectional LSTM which generates contextualized representation. The model computes weighted sum of contextualized representation for each word in the sentence to represent an example using attention. Then this representation is used to decide the membership of each label in 10K ontology.

3.2 Labeled Examples Only

Several learning settings are introduced here where model only learns from labeled examples (single and multi label) disregarding unlabeled data.

Combined Training Set: CE (combined) We shuffle single and multi labeled example sets together, and train the model with this combined set.

Upsampling: CE (upsampling) When we have fewer multi label examples, we upsample multi label data, to match single label data.

Curriculum Learning: CE ($\mathbf{X}_s$ then $\mathbf{X}_m$) We first train with single label data, where we often have abundant examples. Then we further fine-tune this model with multi-annotated data.

MixUp Recent work proposed MixUp (Zhang et al., 2018), a data augmentation method that encourages the model to behave linearly in-between labeled training examples for image data. Berthelot et al. (2019) extended to interpolate between the label and unlabeled data (after assigning a psuedo labels for them). Chen et al. (2020a) applied the MixUp to text classification tasks, showing MixUp outperforms other data augmentation techniques such as back translation (Sennrich et al., 2016; Zhang et al., 2021b) and word replacement. We describe original MixUp algorithm below.

Given two examples (x_m, y_m) and (x_n, y_n), where x is raw input vector and y is one-hot label encoding, it constructs augmented training examples by incorporating the intuition that linear interpolations of feature vectors should lead to linear interpolations of the associated targets:

$$\tilde{x} = \text{mix}(x_m, x_n) = \lambda x_m + (1-\lambda)x_n$$
$$\tilde{y} = \text{mix}(y_m, y_n) = \lambda y_m + (1-\lambda)y_n,$$

where λ is a scalar hyperparameter for mixing both the inputs and labels. It is sampled from a

breaks that assumption, now a premise can occur in both training and evaluation with different hypotheses. However, we find that the performance on examples with/without overlapping premise in the training set does not vary significantly.

Beta(η, η) distribution with a hyper-parameter η. The newly generated training data $(\tilde{x}, \tilde{y})$ are used as a training example, and the learning objective is:

$$L_{\text{mixup}} = \mathcal{L}(\tilde{y}, d(\tilde{x}, \theta)),$$

where $\mathcal{L}$ is the cross entropy loss and $d(.; \phi)$ is a classifier on top of the encoder model which take the mixed representation $\tilde{x}$ as input and returns a probability over a label set. Interpolated annotated data x_m and x_n can be either single label data or multi label data. We define the loss from interpolating single label example and multi label example as $L_{s,m}$, the loss from interpolating multi label example and multi label example as $L_{m,m}$, the loss from interpolating single label example and single label example as $L_{s,s}$. Thus the MixUp (Zhang et al., 2018) loss, in our $\mathbf{X}_s$ + $\mathbf{X}_m$ setting, is defined as

$$\text{Mixup}(\mathbf{X}_s, \mathbf{X}_m) = L_{s,s} + L_{m,m} + \alpha(L_{s,m}),$$

where α is a coefficient (Tarvainen and Valpola, 2017; Berthelot et al., 2019; Fan et al., 2020).

3.3 Semi-supervised Learning

Now we introduce unlabeled examples into training algorithm. Following prior work (Berthelot et al., 2019), we generate pseudo labels for each unlabeled example. For unlabeled x_u, we use hidden states of the model's prediction to generate the pesudo labels (Xie et al., 2020). Considering the unlabeled data set $\mathbf{X}_u = (x_u^1 \ldots, x_u^n)$ where $n \in \{1 \ldots N\}$, the classifier model generates a pseudo label distribution q^n for each data point x_u^n. We sharpen this distribution by taking the argmax of distribution q^n, making a one hot vector $\hat{q^n}$ over the labels. The classifier used to generate the pseudo labels trained jointly in a single end-to-end learning, using the learning signals from the labeled data.

MixUp Three Types of Data After generating the pseduo labels for unlabeled data, we have three types of input: single label examples $\mathbf{X}_s$, multi label examples $\mathbf{X}_m$, and unlabeled examples $\mathbf{X}_u$, all with corresponding labels. We introduce MixUp interpolation among three types of data, integrating all into the objective function as below:

$$\text{Mixup}(\mathbf{X}_s, \mathbf{X}_m, \mathbf{X}_u) = L_{s,s} + L_{m,m} + \alpha(L_{s,m} + L_{s,u} + L_{m,u}).$$

For all settings, we set the maximum value of loss weight α as 2.0 and linearly ramp up α from 0 to its maximum value over the first 100 iterations of training as is common practice (Tarvainen and Valpola, 2017; Berthelot et al., 2019).

4 Experiments

We present performances of our labeling scheme and learning framework in this section. All experimental results are rerun three times with different random seeds to determine the variance, which is small.[4]

4.1 Evaluation Metrics

NLI We follow evaluation metrics from original papers (Bowman et al., 2015; Nie et al., 2020). We report classification accuracy, which is computed twice, once against aggregated gold labels in the original 5-way annotated dataset (old), and against the aggregated label from 100-way annotated dataset (new). Distributional evaluation metrics, Jensen-Shannon Divergence (Endres and Schindelin, 2003), and Kullback-Leibler Divergence (Kullback and Leibler, 1951) are also reported. We present analysis on different evaluation metrics in Section 4.5.

UFET We compute macro-averaged precision, recall, and F1, and the average mean reciprocal rank (MRR), following prior work.

4.2 NLI Results

In Table 3, we evaluate the impact of introducing multi label datasets in the full data setting. Even with a large annotation budget, learning with single label data shows a limited performance, and we see substantial gains on both accuracy and distribution metrics by replacing 5K single label examples with a small amount of multi label data (500 examples). $\mathbf{X}_s$ + $\mathbf{X}_m$ outperforms previously published results (X_s) from Nie et al. (2020). Here we try vanilla curriculum learning, which first trains a model with $\mathbf{X}_s$ data and then fine tune with $\mathbf{X}_m$ data.

With this encouraging initial results, we further explore different learning objectives in more constrained annotation budget scenarios (150K and 6K). The results on ChaosMNLI dataset is presented in Table 4.[5] Across all settings, having only single label data results in inferior performances

[4]The standard deviation value of KL on all method / dataset pairs is lower than 0.02 and the standard deviation of F1 is lower than 0.01.

[5]The results on ChaosSNLI dataset can be found in appendix Table 8. It shows the same trends as the results on ChaosMNLI dataset.

	ChaosSNLI			ChaosMNLI		
	JSD↓	KL ↓	acc (old/new)↑	JSD ↓	KL ↓	acc (old/new) ↑
$\mathbf{X}_s$ (all)	0.229	0.505	0.727 / 0.754	0.307	0.781	0.639 / 0.592
$\mathbf{X}_s$ (our reimpl.,subset)	0.242	0.548	0.684 / 0.710	0.308	0.799	**0.670** / 0.604
$\mathbf{X}_s$ + $\mathbf{X}_m$ ($\mathbf{X}_s$ then $\mathbf{X}_m$)	**0.183**	**0.211**	**0.698 / 0.748**	**0.192**	**0.180**	0.646 / **0.691**

Table 3: Results on ChaosNLI datasets in a high label budget setting. The top block results are from Nie et al. (2020), and the row in grey color are not strictly comparable due to different evaluation sets. Single (our reimpl.,subset) is our implementation of Nie et al. (2020) and evaluate the results on the 500 examples evaluation set sampled from ChaosSNLI and ChaosMNLI.

Data	Learning	Number of Total Labels					
		150k			6k		
		JSD ↓	KL ↓	acc (old/new) ↑	JSD↓	KL ↓	acc (old/new)↑
$\mathbf{X}_s$	CE	0.312	0.572	0.628 / 0.578	0.330	0.753	0.516 / 0.526
$\mathbf{X}_s$	MixUp ($\mathbf{X}_s$)	0.300	0.567	0.628 / 0.580	0.321	0.696	0.518 / 0.528
$\mathbf{X}_s$ + $\mathbf{X}_m$	CE (combined)	0.256	0.370	0.626 / 0.584	0.302	0.422	0.520 / 0.532
$\mathbf{X}_s$ + $\mathbf{X}_m$	CE (upsampling)	0.249	0.293	0.614 / 0.610	0.285	0.421	0.506 / 0.528
$\mathbf{X}_s$ + $\mathbf{X}_m$	CE ($\mathbf{X}_s$ then $\mathbf{X}_m$)	**0.213**	**0.216**	**0.638 / 0.646**	0.298	0.414	0.519 / 0.531
$\mathbf{X}_s$ + $\mathbf{X}_m$	MixUp ($\mathbf{X}_s$, $\mathbf{X}_m$)	0.243	0.288	0.598 / 0.602	0.271	0.409	0.520 / 0.539
$\mathbf{X}_s$ + $\mathbf{X}_u$	MixUp ($\mathbf{X}_s$, $\mathbf{X}_u$)	0.294	0.537	0.626 / 0.566	0.309	0.617	0.519 / 0.529
$\mathbf{X}_s$ + $\mathbf{X}_m$ + $\mathbf{X}_u$	MixUp ($\mathbf{X}_s$, $\mathbf{X}_u$) then $\mathbf{X}_m$	0.290	0.510	0.626 / 0.570	0.295	0.571	0.521 / 0.533
$\mathbf{X}_s$ + $\mathbf{X}_m$ + $\mathbf{X}_u$	MixUp ($\mathbf{X}_s$, $\mathbf{X}_m$, $\mathbf{X}_u$)	0.241	0.287	0.596 / 0.610	**0.266**	**0.384**	**0.522 / 0.540**

Table 4: Results on the ChaosMNLI datasets under limited annotation budget (150K, 6K). Each column block shows the number of total training annotations. All results use the same amount of annotations, and each row block uses roughly same amount of training examples (bottom row block incorporates large unlabeled data). CE represents cross entropy.

compared to dedicating even a small amount of budget to generate multi annotated data (500 examples, each 10-way annotated).

Now we compare different methods to integrate multi label data and single label data. As a baseline, we notate simply combined multi label and single label data as CE (combined). Simple combination does not work when the number of multi label data (0.5K) is much smaller than the total number of single label data (145K), but shows comparable performance in 6K setting where multi label and single label data are more balanced (0.5K multi label data vs. 1K single label data). Upsampling multi label data shows improvement over the CE combined. CE ($\mathbf{X}_s$ then $\mathbf{X}_m$) which is first training the model with single label data and then fine tune with multi label data works better, consistently achieving strong performances in different experimental settings.

Next, we discuss gains from using MixUp data augmentation methods. We observe small yet consistent gains from using example MixUp in single label setting (i.e., $\mathbf{X}_s$: MixUp ($\mathbf{X}_s$) vs. $\mathbf{X}_s$: CE) confirming findings from the previous studies (Zhang et al., 2018). Integrating multi label training examples into MixUp objective shows gains in low annotation budget setting. In high annotation budget settings, where we have fewer multi label examples (500 multi vs. 145K single), CE ($\mathbf{X}_s$ then $\mathbf{X}_m$) yields better results. Nonetheless, MixUp augmentation shows consistent gains compared to shuffling (MixUp($\mathbf{X}_s$, $\mathbf{X}_m$) vs. CE(combined)).

Our results suggest that annotation budget should be distributed carefully. Even under same label budget and the same learning objective, distribution of labels among examples resulted in performance differences (i.e., $\mathbf{X}_s$: CE vs. $\mathbf{X}_s$ + $\mathbf{X}_m$: CE (combined)). Incorporating unlabeled examples (MixUp ($\mathbf{X}_s$, $\mathbf{X}_u$) vs MixUp ($\mathbf{X}_s$)) improves the performances in low label budget settings (6K), but is detrimental in high label budget settings (150K). We hypothesize that imperfect pseudo label for unlabeled examples can interfere the learning.

4.3 UFET Results

Table 5 reports performances on ultra fine entity typing dataset. Instead of using both crowd-sourced data and distant supervision data (Choi et al., 2018), we focus on crowd-sourced data to

Data	Learning	Development Set				Test Set			
		MRR	P	R	F1	MRR	P	R	F1
Choi et al. (2018) (w / full crowd data)	CE	0.181	46.2	15.7	23.4	0.178	44.7	15.3	22.8
Choi et al. (2018) (w / full crowd data)	MixUp ($\mathbf{X}_s$, $\mathbf{X}_m$)	**0.197**	**46.4**	**19.7**	**27.7**	**0.198**	**45.3**	**20.3**	**28.0**
$\mathbf{X}_s$	CE	0.172	45.1	9.1	15.2	0.172	45.8	9.3	15.4
$\mathbf{X}_s$	MixUp ($\mathbf{X}_s$)	0.174	45.5	9.2	15.4	0.176	46.0	9.5	15.7
$\mathbf{X}_s + \mathbf{X}_m$	CE (combined)	0.177	45.6	10.0	16.4	0.180	46.1	10.3	16.8
$\mathbf{X}_s + \mathbf{X}_m$	CE ($\mathbf{X}_s$ then $\mathbf{X}_m$)	0.179	46.2	9.9	16.3	0.181	48.5	10.1	16.7
$\mathbf{X}_s + \mathbf{X}_m$	MixUp ($\mathbf{X}_s$, $\mathbf{X}_m$)	**0.181**	**48.7**	10.2	16.9	**0.183**	**49.6**	10.3	17.1
$\mathbf{X}_s + \mathbf{X}_u$	MixUp ($\mathbf{X}_s$, $\mathbf{X}_u$)	0.172	47.0	9.5	15.8	0.173	47.4	9.6	16.0
$\mathbf{X}_s + \mathbf{X}_m + \mathbf{X}_u$	MixUp ($\mathbf{X}_s$, $\mathbf{X}_m$, $\mathbf{X}_u$)	0.180	48.5	**10.6**	**17.4**	0.181	49.1	**10.6**	**17.4**

Table 5: Results on UFET dataset. Top two rows use the full crowd-sourced data and the bottom rows are based on smaller label annotation budgets, thus results are not comparable (see Table 2 for details).

simulate single label and multi label settings. Similar to previous results, each row block represents different annotation label budgets. Top two rows use the full crowd-sourced data and the results are not comparable to the bottom rows. The bottom rows are based on different annotation budgets such as 500 single label data (see Table 2 for details). Again in this task, using a single label per example results in inferior performances compared to having multiple labels per example ($\mathbf{X}_s$ + $\mathbf{X}_m$: CE ($\mathbf{X}_s$ then $\mathbf{X}_m$) vs. $\mathbf{X}_s$: CE) as multi label data helps model to learn label-label interaction. Similar to NLI task, adding MixUp objective to the single label setting shows gains ($\mathbf{X}_s$: MixUp ($\mathbf{X}_s$) vs. $\mathbf{X}_s$: CE). Having multi label data is crucial for high performances, and MixUp again shows gains in this low resource setting.

4.4 Analysis

How does different learning algorithm compares under domain shift? We compare two promising methods – single and then multi (CE ($\mathbf{X}_s$ then $\mathbf{X}_m$)) and MixUp (MixUp ($\mathbf{X}_s$, $\mathbf{X}_m$)) for their performance in out of domain setting. Prior work suggested MixUp approaches can effectively compensate for the mixmatch between test data and training data (Zhu et al., 2019). Table 6 shows the performances of model trained on SNLI and tested on MNLI dataset. We observe improved accuracy with MixUp compared to training with the curriculum approach (train with single label data and then fine tuning with multi label data).

Should we carefully select which examples to have multiple annotations? Maybe. We experiment on how to select examples to have multiple annotations, using the ideas from Swayamdipta et al. (2020). We finetune with 1K most hard-to-learn, most easy-to-learn, most ambiguous, and randomly sampled examples. Easy-to-learn examples, with lowest label distribution entropy, are the least effective, but the difference is small in our settings. Similarly, our experiments of changing the number of labels (5-way, 10-way, 20-way) did not result in meaningful differences. The experimental results can be found in Table 10 in the appendix.

Learning	JSD	KL	acc (old/new)
CE ($\mathbf{X}_s$ then $\mathbf{X}_m$)	0.339	**0.479**	0.432 / 0.324
MixUp ($\mathbf{X}_s$, $\mathbf{X}_m$)	**0.324**	0.489	**0.490 / 0.480**

Table 6: Out of domain evaluation results: trained on SNLI dataset and tested on MNLI dataset. All used the same amount of annotation (6K labeled data).

Can we use multi label data exclusively without any single label data? In our main experiments, we mixed multi label data with single label data. Here we present a study comparing a setting with X_m only and X_s only on the NLI task, while keeping **small** annotation budget steady (1K labels). On ChaosSNLI dataset, the model trained with single label data (1000 examples, 1-way annotated) achieves JSD: 0.3578, KL: 0.4671, and acc (old/new): 0.581/0.602. For multi label data (500 examples, 2-way annotated), we get JSD: 0.3355, KL: 0.4529, and acc (old/new): 0.592/0.614. We observe a similar trend for ChaosMNLI dataset as well. We cannot claim that X_m only will outperform X_s only in all settings – as models will benefit from being exposed to diverse examples, but in this low resource setting, we observe gains from using multi annotated data alone.

4.5 Calibration: Alternative Approach to Improve Label Distribution Prediction

We introduce using multi label training examples as an efficient way to estimate the distribution of

	JSD	KL	acc (old/new)	H
$\mathbf{X}_s$	0.308	0.799	0.670 / 0.604	0.414
+ temp. scaling	0.233	0.324	0.670 / 0.604	0.720
+ pred smoothing	0.245	0.347	0.670 / 0.604	0.722
+ train smoothing	0.252	0.372	**0.680** / 0.602	0.701
$\mathbf{X}_s$ then $\mathbf{X}_m$	**0.192**	**0.180**	0.646 / **0.691**	0.868

Table 7: Results on ChaosMNLI dataset with calibration methods. The entropy value of human label distribution for ChaosMNLI is 0.732. H represents the predicted label entropy. Lower entropy indicates higher confidence.

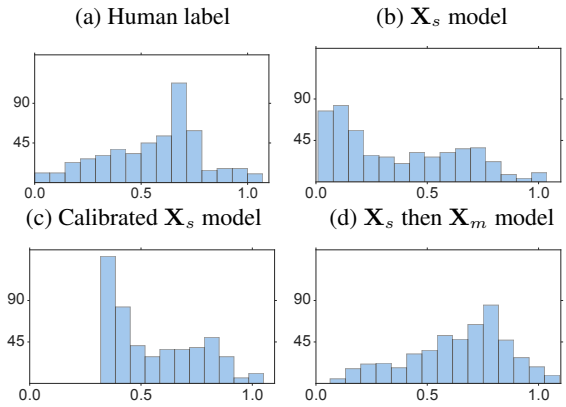

Figure 2: The empirical distribution of label/prediction entropy on ChaosSNLI dataset, where x-axis denotes the entropy value and y-axis denotes the example count on the entropy bin. Initial model prediction (b) shows low entropy values for many examples, being over-confident. Post-hoc calibration nicely shifts the distribution to be less confident, but with artifacts of not being confident on any examples. Finetuning on the small amount of multi-annotated data in (d) successfully simulate the entropy distribution of human labels in (a).

labels. Here, we provide a study of alternative ways to improve label distribution prediction, borrowing ideas from calibration literature, and compare the calibration with training with multi label data.

The key observation is that the predicted label distribution from model trained with single label was over confident, with smaller predicted label entropy 0.414 in Table 7 compared to the human annotated label entropy 0.732. Thus, we smooth the output distribution with three calibration methods (Guo et al., 2018; Miller et al., 1996). The temp. scaling and pred smoothing are post-hoc and do not require re-training of the model. For all methods, we tuned a single scalar hyperparameter per dataset such that the entropy of prediction label distribution matching the entropy of human label distribution.

- **temp. scaling**: scaling by multiplying non-normalized logits by a scalar hyperparameter.
- **pred smoothing**: process softmaxed label distribution by moving α probability mass from the label with the highest mass to the all labels equally.
- **train smoothing**: process training label distribution by shifting α probability mass from the gold label to the all labels equally.

Table 7 reports performances of calibration methods. We find all calibration methods improve performance on both distribution metrics (JSD and KL). Temperature scaling yields slightly better results than label smoothing, consistent with the findings from Desai and Durrett (2020) which shows temperature scaling is better for in-domain calibration compared to label smoothing. Nonetheless, all these results were substantially worse than using multi label data during the training.

Can we estimate the distribution of ambiguous and less ambiguous examples? Figure 2 shows the empirical example distribution over entropy bins. The leftmost plot (a) shows the annotated human label entropy over our evaluation set, and the plot (b) next to it shows the prediction entropy of the baseline RoBERTa model predictions. The model is over-confident about its prediction with single label examples. With label smoothing (plot c), the over-confidence problem is relieved, but the entropy distribution still does not match the distribution of ground truth. Training with multi label data (plot d) makes the prediction distribution similar to the ground truth.

5 Related Work

Assessing the annotation cost associated with learning has long been studied (Turney, 2002). Sheng et al. (2008) studies the tradeoff between collecting multiple labels per example vs. annotating more examples. Researchers have also explored different data labeling strategies, such as active learning (Fang et al., 2017), providing fine-grained rationales (Dua et al., 2020), retrospectively studying the amount of training data necessary for generalization (Mishra and Sachdeva, 2020), and the policy learning approach (Kratzwald et al., 2020). In this work, we study uneven distribution of label annotation budget for training examples, which has not been explored to our knowledge.

Label propagation has been extensively used to infer pseudo-labels for unlabeled data, which are

used to train the classifier (Zhou et al., 2004; Li et al., 2016). Our use of MixUp can be viewed as a way to propagate label information between the single labeled, multi labeled, and unlabeled data.

Rich prior work studies ambiguity in language interpretations (Aroyo and Welty, 2015). A few studies (Passonneau et al., 2012; Ferracane et al., 2021) frame diverging, subjective interpretations as a multi label classification, and few studies (Glickman et al., 2005; Zhang et al., 2017; Chen et al., 2020b) introduce graded human responses. Mayhew et al. (2020) studies training machine translation system with the goal of generating diverse set of reference translations. Pavlick and Kwiatkowski (2019) examines the distribution behind human references for NLI and Nie et al. (2020) presents a larger-scale data collection that we build on.

Earlier version of this paper (Zhang et al., 2021a) study capturing inherent human disagreement in the NLI task through calibration and using a small amount of multi-annotated training examples. This paper expands upon it, introducing a new learning framework for such uneven label distribution schemes. Concurrent to our work, Zhou et al. (2021) introduces distributed NLI, a new NLU task with a goal to predict the distribution of human judgements by applying additional distribution estimation methods such as Monte Carlo (MC) Dropout and deep ensemble methods. While we share a similar goal, our work focuses on how to distribute training labels across examples and how to learn under this new label distribution scheme.

6 Conclusion

Our work demonstrates the benefits from introducing a small amount of multi label examples at the cost of annotating fewer examples. The proposed learning algorithm, extended from MixUp, flexibly takes signals from different types of training examples (single label data, multi label data, and unlabeled data) and show gains upon simply combining different datasets in low annotation budget settings. In this work, we retrospectively study with existing data to question original annotation collection designs. Exploring reinforcement learning or active learning to predict an optimal distribution of annotation budget will be an exciting avenue for future work.

7 Acknowledgements

The authors thank Greg Durrett, Raymond Mooney, Kaj Bostrom, Yasumasa Onoe, and Kenton Lee for helpful comments on the paper draft.

References

Lora Aroyo and Chris Welty. 2015. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Mag.*, 36:15–24.

Nicholas Asher and A. Lascarides. 2005. Logics of conversation. In *Studies in natural language processing*.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations (ICLR)*.

K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, abs/1508.05326.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020a. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, abs/2004.12239.

Tongfei Chen, Zhengping Jiang, Keisuke Sakaguchi, and Benjamin Van Durme. 2020b. Uncertain natural language inference. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 87–96.

I. Dagan, Oren Glickman, and B. Magnini. 2005. The pascal recognising textual entailment challenge. In *MLCW*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *Conference on Empirical Methods in Natural Language Processing*, abs/2003.07892.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194.

Dheeru Dua, Sameer Singh, and Matt Gardner. 2020. Benefits of intermediate annotations in reading comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dominik Maria Endres and Johannes E Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860.

Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. 2020. Bayesian attention modules. *arXiv preprint arXiv:2010.10604*.

Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Copenhagen, Denmark. Association for Computational Linguistics.

Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2021. Did they answer? subjective acts and intents in conversational discourse. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1626–1644, Online. Association for Computational Linguistics.

Markus Freitag, George F. Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *ArXiv*, abs/2104.14478.

Oren Glickman, I. Dagan, and Moshe Koppel. 2005. A probabilistic classification approach for lexical textual entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2018. On calibration of modern neural networks. *Proceedings of the International Conference on Machine Learning (ICML)*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Bernhard Kratzwald, Stefan Feuerriegel, and Huan Sun. 2020. Learning a cost-effective annotation policy for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3051–3062.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Shoushan Li, Jian Xu, Dong Zhang, and Guodong Zhou. 2016. Two-view label propagation to semi-supervised reader emotion classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2647–2655.

Nelson F. Liu, T. Lee, Robin Jia, and Percy Liang. 2021. Can small and synthetic benchmarks drive modeling innovation? a retrospective study of question answering modeling approaches. *ArXiv*, abs/2102.01065.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Stephen Mayhew, K. Bicknell, Chris Brust, Bill McDowell, and Will Monroe. 2020. Simultaneous translation and paraphrase for language education. In *NGT@ACL*.

David J. Miller, A. Rao, K. Rose, and A. Gersho. 1996. A global optimization technique for statistical classifier design. *IEEE Trans. Signal Process.*, 44:3108–3122.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, abs/2004.10645.

Swaroop Mishra and Bhavdeep Singh Sachdeva. 2020. Do we need to create big datasets to learn a task? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 169–173.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143.

Rebecca J. Passonneau, Vikas Bhardwaj, Ansaf Salleb-Aouissi, and Nancy Ide. 2012. Multiplicity and word sense: evaluating and learning from multiply labeled word sense annotations. *Language Resources and Evaluation*, 46:219–252.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

D. Reidsma and Rieks op den Akker. 2008. Exploiting 'subjective' annotations. In *COLING 2008*.

E. T. K. Sang and F. D. Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *ArXiv*, cs.CL/0306050.

Rico Sennrich, B. Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. *ArXiv*, abs/1511.06709.

V. Sheng, F. Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Conference on Empirical Methods in Natural Language Processing*.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1195–1204.

Peter D. Turney. 2002. Types of cost in inductive concept learning. *ArXiv*, cs.LG/0212034.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *naacl*, abs/1704.05426.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Qizhe Xie, Zihang Dai, E. Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training. *arXiv: Learning*.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics*, 5:379–395.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021a. Capturing label distribution: A case study in nli. *arXiv preprint arXiv:2102.06859*.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021b. Knowing more about questions can help: Improving calibration in question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1958–1970, Online. Association for Computational Linguistics.

Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328.

Xiang Zhou, Yixin Nie, and Mohit Bansal. 2021. Distributed nli: Learning to predict human opinion distributions for language reasoning. *arXiv preprint arXiv:2104.08676*.

Yingke Zhu, Tom Ko, and Brian Mak. 2019. Mixup learning strategies for text-independent speaker verification. In *Interspeech*, pages 4345–4349.

Appendix

A Hyperparameters and Experimental Settings

NLI Hyperparameters and Experimental Settings

Our implementation is based on the *HuggingFace Transformers* (Wolf et al., 2020). We optimize the KL divergence as objective with the Adam optimizer (Kingma and Ba, 2014) and batch size is set to 128 for all experiments. The Roberta-base is trained for $3,500$ iterations on single-annotated data. For the finetuning phase, the model is trained for another 30 iterations. The learning rate, 10^{-5}, is chosen from AllenTune (Dodge et al., 2019). For MixUp, the number of training iteration is $3,500$. The η of the Beta(η, η) distribution is 1. We choose the same batch size 128 for single label, multi label, and unlabeled data. Thus it will generate evenly interpolated examples. We set the maximum value of loss weight α as 2.0 and linearly ramp up α from 0 to its maximum value over the first 100 iterations of training as is common practice (Tarvainen and Valpola, 2017; Berthelot et al., 2019).

UFET Hyperparameters and Experimental Settings

Following the settings from Choi et al. (2018), we set the LSTMs' dimension as 100. For word vectors, we use 300 dimensional pretrained Glove. For location vectors, we use 50 dimensions. For sentence length, we cut off the sentence after 50 tokens. For mentions spans, we cut off after 25 characters and ignore mentions longer than 10 words during training. Dropout is use for regularization with a probability of 0.5 for mention representations and 0.2 for input sentences. We set the batch size as 1000. Adam optimizer (Kingma and Ba, 2014) is utilized for optimizing the model parameter with initial learning rate of 0.001. For MixUp, we follow the same settings in the NLI experiments. The number of training iteration is $10,000$. The η of the Beta(η, η) distribution is 1. Same batch sizes are chosen for single label, multi label, and unlabeled data. The maximum value of loss weight α is set as 2.0.

B Full Experimental Results

		Number of Total Annotations								
		150k			15k			6k		
Data	Learning	JSD↓	KL ↓	acc (old/new)↑	JSD ↓	KL ↓	acc (old/new) ↑	JSD↓	KL ↓	acc (old/new)↑
$\mathbf{X}_s$	CE	0.252	0.548	0.670 / 0.670	0.264	0.569	0.648 / 0.650	0.283	0.556	0.632 / 0.626
$\mathbf{X}_s$	MixUp ($\mathbf{X}_s$)	0.251	0.470	0.672 / 0.682	0.263	0.566	0.646 / 0.654	0.277	0.544	0.628 / 0.626
$\mathbf{X}_s + \mathbf{X}_m$	CE (combined)	0.240	0.355	0.676 / 0.672	0.268	0.438	0.642 / 0.654	0.279	0.502	0.633 / 0.628
$\mathbf{X}_s + \mathbf{X}_m$	CE (upsampling)	0.245	0.292	0.664 / 0.674	0.261	0.371	0.620 / 0.660	0.270	0.491	0.618 / 0.620
$\mathbf{X}_s + \mathbf{X}_m$	CE ($\mathbf{X}_s$ then $\mathbf{X}_m$)	**0.217**	**0.227**	0.685 / **0.722**	0.254	**0.285**	0.628 / **0.668**	0.272	0.496	0.636 / 0.629
$\mathbf{X}_s + \mathbf{X}_m$	MixUp ($\mathbf{X}_s, \mathbf{X}_m$)	0.233	0.285	0.682 / 0.682	0.252	0.384	0.662 / 0.658	0.267	0.490	0.610 / 0.636
$\mathbf{X}_s + \mathbf{X}_u$	MixUp ($\mathbf{X}_s, \mathbf{X}_u$)	0.251	0.472	0.672 / 0.670	0.264	0.492	0.660 / 0.656	0.275	0.504	**0.638** / 0.628
$\mathbf{X}_s + \mathbf{X}_m + \mathbf{X}_u$	MixUp ($\mathbf{X}_s, \mathbf{X}_u$) then $\mathbf{X}_m$	0.250	0.454	0.674 / 0.674	0.263	0.461	0.662 / 0.660	0.270	0.496	0.632 / 0.636
$\mathbf{X}_s + \mathbf{X}_m + \mathbf{X}_u$	MixUp ($\mathbf{X}_s, \mathbf{X}_m, \mathbf{X}_u$)	0.232	0.283	**0.686** / 0.694	**0.248**	0.341	**0.668** / 0.666	**0.266**	**0.392**	0.602 / **0.642**

Table 8: Performance on the **ChaosSNLI** dataset development set. Each column block (150k, 15k, 6k) shows the number of total training annotations. All results use the same amount of annotations, and each row block uses roughly same amount of training examples (bottom row block incorporates large unlabeled data).

		Number of Total Annotations								
		150k			15k			6k		
Data	Learning	JSD↓	KL ↓	acc (old/new)↑	JSD ↓	KL ↓	acc (old/new) ↑	JSD↓	KL ↓	acc (old/new)↑
$\mathbf{X}_s$	CE	0.312	0.572	0.628 / 0.578	0.319	0.686	0.552 / 0.528	0.330	0.753	0.516 / 0.526
$\mathbf{X}_s$	MixUp ($\mathbf{X}_s$)	0.300	0.567	0.628 / 0.580	0.315	0.694	0.555 / 0.530	0.321	0.696	0.518 / 0.528
$\mathbf{X}_s + \mathbf{X}_m$	CE(combined)	0.256	0.370	0.626 / 0.584	0.269	0.393	0.550 / 0.530	0.302	0.422	0.520 / 0.532
$\mathbf{X}_s + \mathbf{X}_m$	CE(upsampling)	0.249	0.293	0.614 / 0.610	0.251	0.341	0.545 / **0.588**	0.285	0.421	0.506 / 0.528
$\mathbf{X}_s + \mathbf{X}_m$	CE($\mathbf{X}_s$ then $\mathbf{X}_m$)	**0.213**	**0.216**	**0.638 / 0.646**	**0.246**	**0.258**	**0.560** / 0.562	0.298	0.414	0.519 / 0.531
$\mathbf{X}_s + \mathbf{X}_m$	MixUp ($\mathbf{X}_s, \mathbf{X}_m$)	0.243	0.288	0.598 / 0.602	0.254	0.357	0.534 / 0.568	0.271	0.409	0.520 / 0.539
$\mathbf{X}_s + \mathbf{X}_u$	MixUp ($\mathbf{X}_s, \mathbf{X}_u$)	0.294	0.537	0.626 / 0.566	0.301	0.539	0.544 / 0.560	0.309	0.617	0.519 / 0.529
$\mathbf{X}_s + \mathbf{X}_m + \mathbf{X}_u$	MixUp ($\mathbf{X}_s, \mathbf{X}_u$) then $\mathbf{X}_m$	0.290	0.510	0.626 / 0.570	0.290	0.491	0.554 / 0.564	0.295	0.571	0.521 / 0.533
$\mathbf{X}_s + \mathbf{X}_m + \mathbf{X}_u$	MixUp ($\mathbf{X}_s, \mathbf{X}_m, \mathbf{X}_u$)	0.241	0.287	0.596 / 0.610	0.252	0.348	0.548 / 0.570	**0.266**	**0.384**	**0.522 / 0.540**

Table 9: Performance on the **ChaosMNLI** dataset development set. Each column block (150k, 15k, 6k) shows the number of total training annotations. All results use the same amount of annotations, and each row block uses roughly same amount of training examples (bottom row block incorporates large unlabeled data).

C Label Count Comparison

# Multi	# Single	JSD	KL	acc (old/new)	H
0	150K	0.25	0.55	0.676 / 0.688	0.363
0.5K (20-way)	130K	0.20	0.22	0.676 / 0.726	0.695
1K (10-way)	140K	0.19	0.22	0.684 / 0.732	0.643
5K (5-way)	145K	0.19	0.22	0.676 / 0.732	0.701

Table 10: Label count comparison on ChaosSNLI dataset. The total number of labels is consistent among different rows (150K). H represents the predicted label entropy.

D Training Data Configuration for 6K NLI

Task	Data Setup	# Single	# Multi	# Unlabel	Total # Labels	Total # Examples
Chaos S / MNLI	Original	549k / 392k	0	0	549k / 392k	549k / 392k
	$\mathbf{X}_s$	6k	0	0	6k * 1 = 6k	6k
	$\mathbf{X}_s + \mathbf{X}_m$	1k	0.5k	0	1k * 1 + 0.5k * 10 = 6k	1.5k
	$\mathbf{X}_s + \mathbf{X}_u$	6k	0	549k-6k	6k * 1 = 6k	549k
	$\mathbf{X}_s + \mathbf{X}_m + \mathbf{X}_u$	1k	0.5k	549k-1.5k	1k * 1 + 0.5k * 10 = 6k	549k

Table 11: Training data configurations for 6k NLI. Each configuration is characterized by the number of labels and the number of examples. The number of labels are consistent in all settings. In NLI task, each multi label example contains 10 labels. For completeness, we also provide original training data configurations.