Capture Human Disagreement Distributions by Calibrated Networks for Natural Language Inference

Yuxia Wang[1]*, **Minghan Wang** [2], **Yimeng Chen** [2], **Shimin Tao** [2],
Jiaxin Guo [2], Chang Su [2], Min Zhang[2], Hao Yang [2]
[1] The University of Melbourne, Victoria, Australia
[2] Huawei Translation Services Center, Beijing, China
yuxiaw@student.unimelb.edu.au {wangyuxia5,chenyimeng,taoshimin,
guojiaxin1,suchang8,zhangmin186,yanghao30}@huawei.com

Abstract

Natural Language Inference (NLI) datasets contain examples with highly ambiguous labels due to its subjectivity. Several recent efforts have been made to acknowledge and embrace the existence of ambiguity, and explore *how to capture the human disagreement distribution*. In contrast with directly learning from gold ambiguity labels, relying on special resource, we argue that the model has naturally captured the human ambiguity distribution as long as it's calibrated, i.e. the predictive probability can reflect the true correctness likelihood. Our experiments show that when model is well-calibrated, either by label smoothing or temperature scaling, it can obtain competitive performance as prior work, on both divergence scores between predictive probability and the true human opinion distribution, and the accuracy. This reveals the overhead of collecting gold ambiguity labels can be cut, by broadly solving *how to calibrate the NLI network*.

1 Introduction

Ambiguity is intrinsic to natural language. Previously, it's common to disregard it as noise or as a sign of poor-quality data, because we implicitly make the assumption that there is only one correct label given an example, indicating the unique class it belongs to. However, it is against the subjectivity of many natural language understanding (NLU) tasks, such as natural language inference (NLI) and semantic textual similarity (STS), as their annotations are heavily based on personal experience and opinions. More recent research has gravitated towards the necessity to acknowledge and embrace the existence of ambiguity in NLI.

Pavlick and Kwiatkowski (2019) shows that human disagreements, very often, are not dismissible as annotation noise, but rather persist as collecting more ratings and varying the amount of context provided to raters. Humans judgments cannot be adequately summarized by a single aggregate label or value, [1] but a distribution. ChaosNLI (Nie et al., 2020) provides an empirical distribution by collecting 100 annotations for each instance, to simulate true soft label distribution, which is always used as the ambiguity NLI benchmark.

Premise	Look, there's a legend here.
Hypothesis	See, there is a well known hero here.
Label	C: 1, **E: 57, N: 42**
Source	Chaos-MultiNLI
Premise	A group of onlookers glance at a person doing a strange trick on her head.
Hypothesis	A boy does a card trick.
Label	**C: 56,** E: 1, **N: 43**
Source	Chaos-SNLI

Table 1: Ambiguous examples from ChaosNLI. Label: the first element is the class, the second is the number of annotators among 100 choosing this class. C E N refers to classes of *Contradiction*, *Entailment* and *Neutral*.

For the first example in Table 1 extracted from ChaosNLI, it's totally reasonable to assign either *Neutral* or *Entailment*, depending on the annotators' understanding of relationship between "legend" and "hero". The second shows the disagreement between *Contradiction* and *Neutral* when differing from the context and the annotators' background knowledge.

The challenge is *how to capture this linguistic ambiguity?* In other words, how to model the human disagreement distribution? Pavlick and Kwiatkowski (2019) demonstrated NLI systems trained to predict an aggregate label do not learn the uncertainty that exists among humans' perceptions. So Meissner et al. (2021) explored to train directly on the crowd-sourcing soft label distribution of the annotators. They find training on the same amount of data but targeting the ambiguity distribution instead of one-hot labels can improve the

* Work carried out as an intern at Huawei 2012 Lab.

[1] It refers to previous gold-labels or one-hot (hard) labels.

Findings of the Association for Computational Linguistics: ACL 2022, pages 1524 - 1535

prediction accuracy, and narrow the ChaosNLI JSD scores. However, is the success of JSD reduction and accuracy improvement completely attributed to the usage of special resources with ambiguity labels? What essentially makes the contribution?

In our view, it's the label smoothing of soft labels that plays an important role, owing to its effectiveness on regularisation, re-calibration and loss correction (Patrini et al., 2016; Lukasik et al., 2020; Müller et al., 2019). Specifically, the ambiguity distribution employed by Meissner et al. is just a special soft label, targeting label smoothing outputs is believed to have comparable performance, much cheaper than collecting crowdsourcing labels.

We further posit that the linguistic ambiguity have been learned by a well-calibrated model, trained either hard or soft labels. The predictive probability of a perfectly-calibrated model can reflect the true correctness likelihood, i.e. empirical accuracy is equal to the prediction confidence. Empirical accuracy is obtained from observations across the human judges. That is, predictive confidence (uncertainty) can represent the human judgment distribution when model is calibrated. To this end, not only label-smoothing, but other re-calibration approaches such as temperature scaling can reach the same goal. Our experiments confirmed our hypothesis, when model is calibrated, it can obtain competitive ChaosNLI divergence scores and bring accuracy boost.

Our contributions are two folds: (1) We propose the hypothesis: a well-calibrated network can naturally capture linguistic ambiguity, regardless of using special resource. It reasonably explains the success of training with ambiguity labels, and converts question of "how to capture human disagreement distribution?" to a more general one "how to train a calibrated model?" Our experiments confirm that commonly-used re-calibration methods are as effective as targeting at ambiguous annotations. (2) Knowledge of linguistic ambiguity learned from the general domain benefits biomedical domain as well, which suggests ambiguity signals can be transferred across domains. But calibration is not an intrinsic property of a model, it's data-dependent.

2 Background

Label smoothing (LS) is a mixture of one-hot label vector $\mathbf{y}_{hot}$ and the uniform distribution:

$$\mathbf{y}_{ls} = (1 - \alpha)\mathbf{y}_{hot} + \alpha/K$$

where K is the number of label classes, α is a hyper-parameter that determines the amount of smoothing. α = 0, $\mathbf{y}_{ls} = \mathbf{y}_{hot}$, α = 1, $\mathbf{y}_{ls}$ is the uniform distribution.

In the setting, where the loss function L is cross entropy, and the model applies the *softmax* (σ_{SM}) to the penultimate layer's logit vector $\mathbf{z}$ to compute probability $\mathbf{p}$, the gradient of the cross entropy loss function with respect to the logits is:

$$\nabla L = \sigma_{SM}(\mathbf{z}) - \mathbf{y}; \partial L/\partial z_i = p_i - y_i$$

We can see that gradient descent will try to make $\mathbf{p}$ as close to $\mathbf{y}$ as possible. When $\mathbf{y}$ is the one-hot label, models will classify every training example correctly with the confidence of almost 1. This not only conflicts with the inherent disagreement of NLI, but tends to result in over-confident and less-generalised models as well.

Concretely, suppose K = 3, $\mathbf{z} = [z_1, z_2, z_3]$, the consequence of using one-hot encoded label $y = [1, 0, 0]$ is that z_1 will be extremely large and the other logits will be extremely small: $z_1 \gg z_2$ and $z_1 \gg z_3$. In other words, one-hot labels encourage the largest possible logit gaps to be fed into the *softmax* function. Moreover, the gradient is bounded between -1 and 1, as p_i and y_i is probability value $\in [0, 1]$. Large logit gaps combined with the bounded gradient lead models to be less adaptive and too confident. In contrast, smoothed labels encourages small logit gaps. Label smoothing restrains the largest logit from becoming much bigger than the rest, improving model generalisation ability, and prevents overconfident predictions, making model more calibrated instead of over-confidence.

Pereyra et al. (2017) explains label smoothing by connecting it to a maximum entropy based confidence penalty through the direction of the KL divergence. Specifically, to penalize confident (low entropy) output distributions, adding the negative entropy to the negative log-likelihood L_{NLL} during training as Eq (1). Applying label smoothing is interpreted as adding a confidence penalty $D_{KL}(u||\mathbf{p})$ to original loss as a regularizer, where u is uniform distribution.

$$L = L_{NLL} - \beta H(\mathbf{p}) \quad (1)$$

$$L = L_{NLL} - D_{KL}(u||\mathbf{p}) \quad (2)$$

Müller et al. shows label smoothing implicitly calibrates learned models so that the confidences of their predictions are more aligned with the accuracy of predictions. Beside, it's functional in backward

loss correction and denoising (Patrini et al., 2016; Lukasik et al., 2020).

Classification Re-calibration Apart from label smoothing, post-hoc scaling, such as matrix and vector scaling and temperature scaling, is demonstrated to be effective to re-calibrate DNNs (Guo et al., 2017). They apply a linear transformation $\mathbf{W}\mathbf{z}_i + \mathbf{b}$ to the logits $\mathbf{z}_i$: $\hat{\mathbf{q}}_i = \sigma_{SM}(\mathbf{W}\mathbf{z}_i + \mathbf{b})$. The parameters $\mathbf{W}$ and $\mathbf{b}$ are optimized with respect to NLL on the validation set. As the number of parameters for matrix scaling grows quadratically with the number of classes K, vector scaling is defined as a variant where $\mathbf{W}$ is restricted to be a diagonal matrix.

Temperature scaling uses a single scalar parameter T>0 for all classes. T is optimized with respect to NLL on the validation set as well.

$$\hat{\mathbf{q}}_i = \sigma_{SM}(\mathbf{z}_i/T)$$

Because the parameter T does not change the maximum of the *softmax* function, the class prediction remains unchanged, temperature scaling does not affect the model's accuracy.

Related Work Several recent studies, in parallel with ours, explore to capture NLI label distribution. The most similar work is Zhang et al. (2021). They also train with multi-annotated examples, label smoothing and temperature scaling, but differ from the motivation, implementation and results. Specifically, we pay much attention to analyzing why re-calibration approaches are useful, and investigating the connection between model calibration error, sharpness with NLI distribution distance, while they merely conduct an empirical case study without any deep analysis. In the experiment, they train with the majority of ChaosNLI and test on only 500 examples sampled from ChaosNLI, but we train using SNLI/MNLI corpus and evaluate on the whole ChaosNLI. Statistically, our results are more convincing; In addition to the result, their distribution divergence declines at the cost of declining accuracy of 4 points, from 0.72 to 0.68, whereas our accuracy remains the same level.

Zhou et al. (2021) paid more attention to Bayesian estimation and model distillation to learn label distribution, without focusing on label smoothing on which we concentrate and analyzed deeply, including uncertainty metrics and soften factor selection. Overall, our work complements concurrent studies with lots of comprehensive and useful analysis in terms of label smoothing and temperature scaling.

3 Hypothesis

While softmax of NLI models trained with hard labels allows the model to represent predictive confidence, this probability does not necessarily mimic the uncertainty that exists among humans' perceptions (Pavlick and Kwiatkowski, 2019). We speculate targeting one-hot labels leads model to be over-confident. The miscalibration of over-confidence results in the disability to represent human disagreement distributions correctly.

Meissner et al. (2021) explores to train on the empirically-gold soft labels collected by crowdsourcing annotations. They find training on the same amount of data but targeting the ambiguity distribution instead of hard labels can reduce ChaosNLI divergence scores (JSD) and achieve higher performance. So they advocate to use crowdsourcing techniques to obtain a label distribution by collecting multiple annotations given an instance, instead of only one as before.

However, is the success completely attributed to the usage of empirically-gold label distribution as training target? What essentially makes difference? In our view, it's the soft label — output of label-smoothing, as an effective technique on re-calibration, regularisation and loss correction that plays an important role in this success. The ambiguity distribution they employed is just a special soft label, targeting other label smoothing outputs is believed to have comparable performance, but much cheaper than crowdsourcing distribution.

Moreover, we argue that even training with gold soft labels as AmbiNLI (Meissner et al., 2021), cannot always obtain improvements. It may bring degradation when model has been under-confident or calibrated, as continuous training with soft labels will exacerbate under-confidence, which deviates prediction away from the correct one. Besides, AmbiNLI only showed performance on one special benchmark — ChaosNLI which concentrates on ambiguous cases. How about performance on other corpus that consist of both ambiguous and extreme non-ambiguous instances?

Therefore, we posit that the linguistic ambiguity have been learned by models that is well-calibrated, even if just trained on previous one-hot labels. And not only label-smoothing, but other re-calibration approaches can reach the same goal.

Dataset	Train	Dev	Test	Prem
SNLI	550,152	10,000	10,000	14.10
MultiNLI-matched	392,702	10,000	10,000	22.25
MultiNLI-mismatched	0	10,000	10,000	22.54
MedNLI	11,232	1,395	1,422	20.00
UNLI	55,517	3,040	3,040	
Chaos-SNLI	–	–	1,514	
Chaos-MNLI	–	–	1,599	
AmbiNLI-S	18,152	–	–	
AmbiNLI-M	18,048	–	–	
AmbiNLI-U	55,517	–	–	

Table 2: Statistics information of NLI datasets. Prem is the mean token count among premise sentences.

So we conduct a case study to explore: 1) Can other soft labels generated from the label smoothing achieve competitive results as crowdsourcing ambiguity distribution? 2) Is training using soft labels always better than using one-hot labels? and vice versa? 3) Without soft labels, can other recalibration methods narrow the distance to true correctness probability either?

4 Dataset and Metric

This section gives descriptions of datasets throughout this work, and metrics to assess predictions.

4.1 Datasets

SNLI (Bowman et al., 2015) is a large-scale (570k pairs) NLI resource based on image captioning, in which 56,951 (10%) pairs are validated after the first-stage construction of three hypothesis sentences given a premise towards "definitely true", "may be true" and "definitely false". In validation, additional four annotators are asked to assign labels for a pair of *(premise, hypothesis)*, yielding five annotations. If any one of the three labels was chosen by at least three of the five annotators, it was chosen as the gold label. All examples in the test and development sets have been validated.

MultiNLI (MNLI) (Williams et al., 2018) improves upon SNLI in both its coverage and difficulty by offering data from ten distinct genres of written and spoken English, making it possible to capture more of the complexity of modern English, and supplying explicit setting for evaluating cross-genre domain adaptation.

The disagreement and subjectivity among humans in NLI annotation results in the reflection of ambiguous labels. Many efforts have been made to embrace linguistic ambiguity, regarding them as an intrinsic property of the populations, instead of noise of the data collection and the uncertainty of individual annotators.

ChaosNLI (Nie et al., 2020) is created, by collecting 100 annotations per example for 3,113 examples in SNLI (1,514) and MNLI (1,599), denoting as Chaos-SNLI and Chaos-MNLI respectively. Two sets are a subset of the SNLI development set and a subset of MultiNLI-matched development set, in which the examples satisfy the requirement that their majority label agrees with only three out of five individual labels collected by the original work. It's extensively used as a standard benchmark in ambiguity evaluation.

UNLI (Chen et al., 2020) shifts NLI task away from categorical labels, targeting subjective probability assessment (a numerical value $\in [0, 1]$). UNLI re-annotated a subset of SNLI, resulting in 55,517, 3,040 and 3,040 for train, validation and test set, where annotators are asked to estimate how likely the situation described in the hypothesis sentence would be true given the premise.

AmbiNLI (Meissner et al., 2021) is constructed based on existing datasets, converting one-hot or regression numerical labels to a probability distribution. On SNLI development, test set, and MNLI (matched and mismatched) development set with 5 annotations, it converts an ambiguity distribution by simply counting the number of annotations for each label and then scaling it down into probabilities, denoting to AmbiSNLI and AmbiMNLI. Their combination is named as AmbiSM. To avoid overlap with ChaosNLI, they remove the samples used in ChaosNLI. Samples of UNLI training set are converted by a simple linear approach. [2]

MedNLI (Romanov and Shivade, 2018) is a dataset annotated by doctors, grounded in the medical history of patient. Statistical information is exhibited in Table 2.

Overall, gold labels of SNLI, MNLI and MedNLI are one-hot, or refer to hard labels. The label of UNLI is continuous value, which are not directly applied in our experiment, but the corresponding discrete converted version in AmbiNLI. The gold label of AmbiNLI and ChaosNLI is a distribution, we denote as gold ambiguous label, ambiguity label or distributional labels. Note that these two ambiguity datasets have hard label as well, i.e. the largest-probability class. And AmbiNLI is split and employed in both training and test, while ChaosNLI is only utilized as test set.

[2] See details in original paper or appendix.

4.2 Metrics

Accuracy and F1 only focus on the class type of maximum probability, which is inadequate to evaluate ambiguity distributions.

Jensen-Shannon Distance (JSD) (Endres and Schindelin, 2003) is used to measure the distance between the softmax multinomial distribution of the models and the distributions over human labels.

Calibration Metrics: Calibration is a frequentist notion of uncertainty which measures the discrepancy between subjective forecasts and empirical frequencies. In perfect calibration, neural networks produce confidences that do represent true probabilities. It can be measured by expected calibration error (**ECE**), and proper scoring rules such as negative log likelihood (**NLL**) (Guo et al., 2017). ECE is defined as:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |acc(B_m) - conf(B_m)|$$

To estimate the expected accuracy from finite N samples, we group predictions into M interval bins (each of size 1/M) and calculate the accuracy of each bin. Let B_m be the set of indices of samples whose prediction confidence falls into the interval $I_m = (\frac{m-1}{M}, \frac{m}{M}]$. The accuracy and the average confidence within bin B_m is

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{I}(\hat{y}_i = y_i)$$

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

where $\hat{y}_i$ and y_i are the predicted and true class labels for sample i, $\hat{p}_i$ is corresponding confidence.

Reliability Diagram is a visual representation of model calibration in classification, plotting expected sample accuracy as a function of confidence. If the model is perfectly calibrated, the diagram should plot the identity function. Any deviation from a perfect diagonal represents miscalibration.

Sharpness measures the average confidence on the dataset as a whole, rather than on a bin. Aligning with empirical accuracy, it tells the model overall is under- or over-confident.

$$sharpenss = \frac{1}{N} \sum_{i=1}^{N} max([p_1, p_2.p_3])$$

5 Case Study

This section empirically verifies our hypotheses.

Experimental Setup For fair comparison, we follow the setup of AmbiNLI (Meissner et al., 2021), and reproduce the experiments completely. We use BERT-base (Devlin et al., 2019) with pre-trained weights and a softmax classification head. We use a batch size of 64 and learning rate of 1e-5. [3]

We first obtain a base-NLI model by pre-training 3 epochs on the gold-labels of the combination of SNLI and MNLI training sets. Meissner et al. observed that this pre-training step is necessary to provide the model with a general understanding of the NLI task. We then finetune the model on other NLI dataset, setting the training objective to be the minimization of the cross-entropy between the output probability distribution and the target ambiguity distribution. For evaluation, we compute the ChaosNLI divergence scores, calibration error and sharpness. The reproduced results are closely similar to the original paper shown in Table 4. [4]

5.1 Label Smoothing VS Ambiguity Labels

We decompose the comparison between label smoothing and gold ambiguity soft labels into three sub-problems: (1) Label smoothing empirically has been shown to improve both predictive performance and model calibration in image classification and machine translation (Müller et al., 2019; Lukasik et al., 2020), is it effective on NLI task either, decreasing ECE?
(2) How to search for an optimal soften factor α?
(3) Can label smoothing reduce JSD and improve accuracy when it obtains small ECE?

We apply the unigram label smoothing (Xie et al., 2016). The hyperparameter $\alpha \in (0, 1)$ controls the soften strength, meanwhile reflecting the correctness probability of the target label.

$$y_i = \begin{cases} \alpha & i = target \\ \frac{1-\alpha}{K-1} & i \neq target \end{cases}$$

For example, α=0.8, **y**=$[0.8, 0.1, 0.1]$ when target is the first class, contrasting with ambiguous labels.

How to search an optimal α used in ChaosNLI evaluation? What relates to a proper α given a dataset? In prior work, α is generally tuned by a validation set. It tends to be set as 0.9 over many corpus (Pereyra et al., 2017; Müller et al., 2019), regarded as introduced label noise to help the model

[3] Limited by our GPU memory, 64 is used instead of 128.
[4] The results are not exactly same perhaps due to different training batch size.

to be more robust, by moving the decision boundary closer to a class (Lukasik et al., 2020).

Li et al. shows that the inductive bias of the label smoothing is dependent on the statistical structure of the data. They concretely cluster data by Bayes error rate (BER) bias $\mathcal{R}(\mathbf{x})$, and then learn cluster-dependent smoothing strength $\alpha(\mathbf{x})$, where $P(y = k|\mathbf{x})$ is the conditional posterior probability.

$$\mathcal{R}(\mathbf{x}) = 1 - \max_{k \in [K]} P(y = k|\mathbf{x})$$

In our experiment, we simplify the learnable $\alpha(\mathbf{x})$ to conventional tuning manner, but maintain the cluster-dependent. We first predict label probability $\hat{P}$ for AmbiSNLI and AmbiMNLI using the base-NLI model, and then extract validation sets for AmbiSNLI and AmbiMNLI respectively, by the condition of $\hat{p_i} \in (a, b)$, $\hat{p_i} = \max(\hat{P})$ is the predictive confidence ("conf" in short), remaining partition of both datasets are combined as the training set. Then we search the best α depending on the validation set.

From the perspective of calibration, label smoothing aims to align accuracy with the predictive probability, when α is the target probability in our setting, we expect that $\alpha \to Accuracy$, obtains the smallest ECE. During investigation, conditioned conf interval (a,b) $\in$ {(0.3,0.4), (0.4,0.6), (0.6,0.8), (0.8,0.9), (0.9,1.0)}. [5] We evaluate a set of α=[1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3] for each validation set filtered by (a,b), α=1.0 is the setting using one-hot labels.

Label smoothing can effectively calibrate NLI model, and decrease ECE. In Figure 1, training on soft labels either of gold ambiguity labels collected by crowdsourcing or label smoothing are useful to calibrate the base-NLI model (red line), i.e. yellow and green lines are both closer to the black diagonal line that represents the perfect calibration. And continuous fine-tuning on one-hot hard labels gets the blue line more deviate from the diagonal line than the base-NLI model, leading to a more over-confident one. This can also be observed in Figure 2 — ECE bar chart of 0.8-0.9 conf interval. Fine-tuning using ambiguous soft labels and label smoothed labels can obtain lower ECE than base-NLI, while hard labels leads to higher ECE. However, the interval of 0.9-1.0 demonstrates the opposite results. This indicates using ambiguous labels is not always better than hard labels.

[5] NLI is a three-class classification task, $\hat{p_i} > 0.33$, so start from (0.3,0.4) rather than (0,0.1)

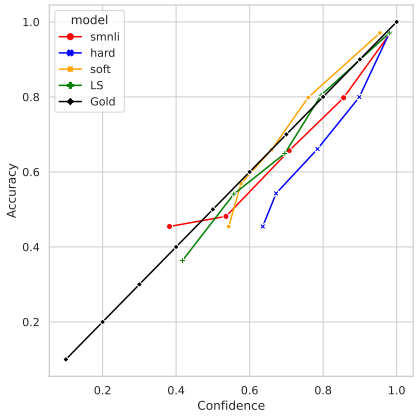

Figure 1: Reliability diagram of AmbiSNLI validation set under different models. smnli=base-NLI model, hard=fine-tune smnli on one-hot labels of remained AmbiSNLI+AmbiMNLI training data, soft=gold ambiguous labels, LS=soft labels from label smoothing with the optimal α. Gold=diagonal line representing the perfect calibration, under this line means over-confidence, over the line is under-confidence.

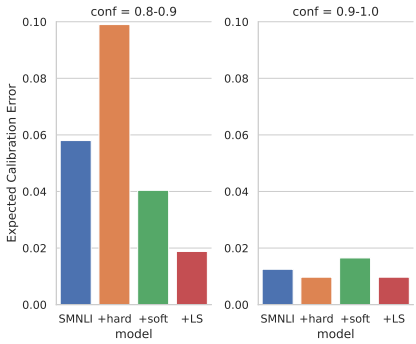

Figure 2: ECE of AmbiSNLI validation set over conf intervals (0.8,0.9) and (0.9,1.0).

The optimal α is not only dependent on the statistical structure of the data, but also the state of the model. In the conf interval of 0.9-1.0, the base-NLI model has almost reached the perfect calibration, ambiguity labels cuts the legitimate confidence. So we can see the yellow line is mostly above the diagonal line (Figure 1), it's the indication of being under-confident. Table 3 shows the best choice of α for each interval. It relates to the varying predictive confidence and the empirical accuracy, but not being equal to either of them, while empirically higher than the accuracy.

Label smoothing can improve accuracy and reduce JSD, being comparable to using gold ambiguity labels. We draw the accuracy (left) and JSD (right) of three edge intervals in Figure 3. On low-confidence interval 0.3-0.4, label smoothing

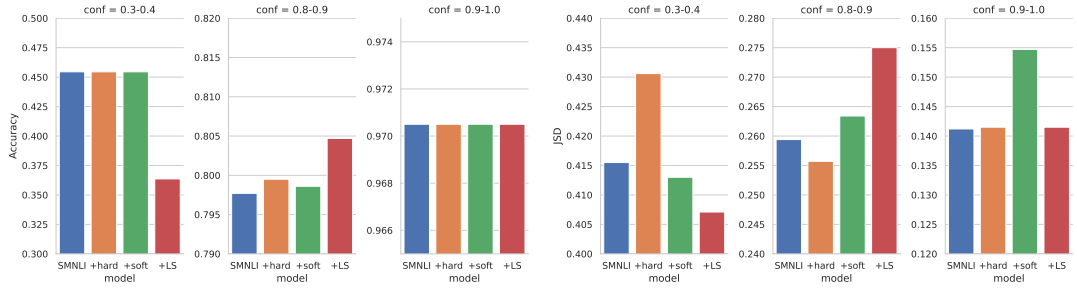

Figure 3: Accuracy and JSD of AmbiSNLI validation sets over three edge intervals. +hard=continuous fine-tuning on one-hot labels of AmbiSNLI+AmbiMNLI training set, +soft=gold ambiguous labels, +LS=label smoothing.

(a,b)	(0.3,0.4)	(0.4,0.6)	(0.6,0.8)	(0.8,0.9)	(0.9,1.0)
num_val	11	538	1080	1152	15371
Acc_{smnli}	0.45	0.48	0.65	0.80	0.97
$\alpha_{optimal}$	0.60	0.80	0.90	0.90	1.00

Table 3: The size and empirical accuracy of AmbiSNLI validation sets under varying conf interval condition, and the corresponding optimal soften strength α.

(LS) reduces JSD by a larger margin than using gold ambiguous labels. But one may argue that applying LS is harmful, since it declines the accuracy. It's not a rigorous conclusion in the context of ambiguity. As accuracy is less meaningful in low-confidence cluster. Accuracy is strictly applicable in deterministic setups, counting the percentage of exact matching pairs, based on the assumption that there is only one correct score. So JSD is favored over accuracy in low-confident cluster.

On middle-confidence interval 0.8-0.9, both improve accuracy, but increase divergence distance. On high-confidence interval of 0.9-1.0, accuracy remains steady in a high level, but ambiguity labels results in large JSD. AmbiMNLI validation sets exhibit similar consequences as these three findings of AmbiSNLI, see Appendix for details.

5.2 Soft Labels VS Hard Labels

Is training using soft labels always better than using one-hot labels? and vice versa? From Figure 3, we know the answer is *NOT*. In high accuracy cluster with high confidence, hard labels is needed to force model to be certain for the predictions. That's why when conf=0.8-1.0, hard labels can reduce JSD while soft labels are unable to do so. Therefore, targeting soft labels are not always superior to one-hot labels. It tends to show positive effect in the highly-uncertain cluster, such as conf=0.3-0.4, in which both gold ambiguity labels and LS decrease JSD, while hard labels make it rise significantly.

5.3 Evaluation on ChaosNLI

To confirm findings induced above and make direct comparison with AmbiNLI original results, we evaluate on ChaosNLI. Based on the finding that soften strength α should be set higher than the accuracy, we set α=0.8 and 0.6 for Chaos-SNLI and Chaos-MNLI respectively in label smoothing.

As shown in Table 4, compared with the baseline model, continuous fine-tuning on AmbiSM hard improves the accuracy, but predictive probability deviates more from the true distribution, resulting in larger JSD. While fine-tuning over cheap soft labels generated by label smoothing is as effective as ambiguity distribution collected by expensive crowdsourcing to narrow JSD, improve accuracy simultaneously. Moreover, they demonstrate sufficient strength to calibrate the model, forcing the predictive probability closer to the true correctness likelihood. reflecting in small ECE and matched sharpness value with the empirical accuracy.

We also experiment with other α settings, results verify that α=0.8 is the best on Chaos-SNLI and 0.6 on Chaos-MNLI.

5.4 Re-calibrate by Temperature Scaling

We posit once the model is calibrated, it naturally has captured the human agreement distribution. In this section, we investigate whether other re-calibration method like temperature scaling (TS) can reduce JSD, capturing linguistic ambiguity.

We search the optimal temperature T based on AmbiSM as validation set, and evaluate on Chaos-SNLI and Chaos-MNLI. As models tend to be over-confident, we search T>1.0. Table 5 shows, on validation set, T=1.2 and 1.5 can obtain the smallest ECE for AmbiSNLI and AmbiMNLI resp. We find that temperature scaling can reduce ECE and make the sharpness be more matched with empirical accuracy, but seemingly cannot decrease JSD. We

Dataset	Chaos-SNLI						Chaos-MNLI					
Metrics	JSD ↓	Accuracy ↑	F1-macro ↑	NLL ↓	ECE ↓	sharpness	JSD ↓	Accuracy ↑	F1-macro ↑	NLL ↓	ECE ↓	sharpness
smnli baseline	0.2443	0.7477	0.7365	0.7650	0.1338	0.8808	0.3432	0.5585	0.5566	1.4739	0.2971	0.8568
+ AmbiSM hard	0.2606	**0.7596**	**0.7462**	0.7947	0.1512	0.9084	0.3476	0.5829	0.5756	1.4405	0.2921	0.8748
+ AmbiSM (0.9)	0.2506	0.7517	0.7375	0.7370	0.1178	0.8661	0.3191	0.5829	0.5738	1.1826	0.2674	0.8534
+ AmbiSM (0.8)	0.2334	0.7576	0.7442	0.6498	**0.0286**	**0.7500**	0.2642	0.5822	0.5734	0.9814	0.1538	0.7353
+ AmbiSM (0.7)	0.2619	0.7517	0.7386	0.7116	0.0752	0.6746	0.2580	0.5816	0.5726	0.9570	0.0817	0.6640
+ AmbiSM (0.6)	0.2860	0.7517	0.7387	0.7682	0.1686	0.5811	0.2553	0.5804	0.5721	0.9467	**0.0179**	**0.5715**
+ AmbiSM (0.5)	0.3171	0.7490	0.7358	0.8579	0.2570	0.4900	0.2648	**0.5847**	0.5751	0.9742	0.1038	0.4841
+ AmbiSM soft	**0.1918**	0.7543	0.7420	**0.5905**	0.0513	0.8036	0.2758	0.5816	0.5755	1.0306	0.2037	0.7863

Table 4: Results on Chaos-SNLI/MNLI, fine-tuning on AmbiSM with different labels. AmbiSM (α) applies LS.

Model	JSD ↓	Accuracy ↑	F1-macro ↑	NLL ↓	ECE ↓	sharpness
AmbiSNLI						
smnli	0.1645	0.9261	0.9254	0.2216	0.0189	0.9449
+ T=1.2	0.1738	0.9261	0.9254	0.2157	**0.0043**	**0.9273**
AmbiMNLI						
smnli	0.1899	0.8683	0.8670	0.3780	0.0545	0.9228
+ T=1.2	0.2011	0.8683	0.8670	0.3553	0.0337	0.9020
+ T=1.4	0.2132	0.8683	0.8670	0.3481	0.0155	0.8796
+ T=1.5	0.2195	0.8683	0.8670	0.3486	**0.0142**	**0.8680**

Table 5: Results using TS on AmbiSNLI (upper) and AmbiMNLI (bottom). The bold is the smallest ECE, and sharpness matches empirical accuracy the most closely.

Model	JSD ↓	Accuracy ↑	F1-macro ↑	NLL ↓	ECE ↓	sharpness
Chaos-SNLI						
smnli	0.2443	0.7477	0.7365	0.7650	0.1338	0.8808
+T=1.2	0.2274	0.7477	0.7365	0.6926	0.1092	0.8554
+T=1.4	0.2147	0.7477	0.7365	0.6504	0.0830	0.8298
+T=2.0	**0.2004**	0.7477	0.7365	**0.6114**	**0.0242**	**0.7565**
Chaos-MultiNLI						
smnli	0.3432	0.5585	0.5566	1.4739	0.2971	0.8568
+T=1.5	0.2899	0.5585	0.5566	1.1360	0.2261	0.7850
+T=2.0	0.2567	0.5585	0.5566	1.0042	0.1614	0.7211
+T=4.0	**0.2328**	0.5585	0.5566	**0.9215**	**0.0301**	**0.5607**

Table 6: Results with varying temperature T on Chaos-SNLI (upper) and Chaos-MNLI (bottom).

speculate it's due to the inaccurate "gold" ambiguity distribution of AmbiSM (5 annotations) used in calculation of JSD between predictive probability.

As shown in Table 6, applying temperature scaling on both Chaos-SNLI (T=1.2) and Chaos-MNLI (T=1.5) are effective to decrease ECE, forcing the predictive probability closer to the empirical accuracy, and we observed JSD and NLL decline at the same time, compared with baseline model trained using smnli without temperature scaling. This indicates that in addition to label smoothing, temperature scaling is also useful to capture human linguistic ambiguity as long as they are calibrated with small calibration error.

We find after temperature scaling using T=1.2 and 1.5, models are still over-confident with large ECE, we wonder: can proper T calibrate them furthermore? Will JSD decrease with the decline of ECE? Note that it's not correct to choose the optimal T according to the held-out test set, we do so here just for case study. [6] Therefore, we increase T=1.2 to 2.0 for Chaos-SNLI, it reaches the smallest ECE, JSD is observed to decline as the same time, as demonstrated in Table 6. On Chaos-MNLI, T increases from 1.5 to 4.0, JSD consecutively drops from 0.29 to 0.23.

[6]Note that this does not invalidate other empirical observations.

6 Domain Transfer

It's well-known that accuracy will drop in domain transfer. But how about JSD and ECE? Is calibration an intrinsic property of the model which is independent of evaluation benchmark? Can knowledge of linguistic ambiguity transferred across domains? Can the property of calibration learned from general domain transfer to the biomedical?

To observe how accuracy, JSD and ECE varies in domain transfer, we first evaluate AmbiSNLI, AmbiMNLI and MedNLI test set, under three NLI models: snli, mnli and smnli trained using SNLI, MNLI training set and their combinations resp. Evaluation of snli model on AmbiMNLI is across textual genres, same as mnli model on AmbiSNLI. All models on MedNLI is across-domain, contrasting with in-domain evaluation — snli and smnli on AmbiSNLI, mnli and smnli on AmbiMNLI.

Metrics of JSD and ECE are data-dependent as accuracy. Table 7, 8 show that they become larger in textual genre and domain transfer. In other words, the model is perfectly-calibrated on benchmark A, but it may poorly-calibrated in other benchmarks that are distantly-distributed from its training data. Calibration is not an intrinsic property of the model, but varies according to data.

The knowledge of linguistic ambiguity learned from general-purpose domain can be transferred to the medical. In middle of Table 8,

Dataset	AmbiSNLI			AmbiMNLI		
Model	Accuracy ↑	JSD ↓	ECE ↓	Accuracy ↑	JSD ↓	ECE ↓
snli	0.9227	0.1639	0.0276	0.7567	0.2693	0.1207
mnli	0.8158	0.2373	0.1069	0.8669	0.1903	0.0639
smnli	0.9261	0.1645	0.0189	0.8683	0.1899	0.0545

Table 7: Accuracy, JSD and ECE on AmbiSNLI/MNLI under three NLI models: snli, mnli and smnli trained using SNLI, MultiNLI train and their combinations.

Dataset	MedNLI validation			MedNLI test		
Model	Accuracy ↑	NLL ↓	ECE ↓	Accuracy ↑	NLL ↓	ECE ↓
snli	0.6179	1.2681	0.2241	0.5985	1.3442	0.2445
mnli	0.6201	1.1780	0.2325	0.6013	1.2939	0.2506
smnli	0.6301	1.0212	0.1904	0.6125	1.1232	0.2104
AmbiSM$_{hard}$	0.6358	1.1126	0.2183	0.6139	1.2289	0.2409
+ AmbiSM$_{soft}$	0.6373	0.8783	0.1279	0.6048	0.9569	0.1542
+TS$_{T=1.6}$	0.6373	0.8168	0.0468	0.6048	0.8663	0.0697
+LS$_{\alpha=0.8}$	0.6444	0.8216	0.0385	0.6188	0.8730	0.0687
MedNLI	0.8057	0.4827	0.0486	0.7771	0.5656	0.0655
+TS$_{T=1.2}$	0.8057	0.4722	0.0219	0.7771	0.5444	0.0333
+LS$_{\alpha=0.9}$	0.8022	0.4940	0.0320	0.7771	0.5514	0.0210

Table 8: Results on the MedNLI validation and test sets.

training on general NLI dataset: AmbiSM gold ambiguous labels, label smoothing and temperature scaling can improve accuracy and reduce NLL of MedNLI, compared with hard labels. [7] This suggest linguistic ambiguity information can be transferred, though not remarkable. Continuous fine-tuning over MedNLI training set improves performance by a large margin over all metrics. Two commonly-used re-calibration methods are effective in biomedical domain as well.

7 Conclusion

In this paper, we explore *how to capture the human linguistic ambiguity* from the perspective of model calibration. We empirically verify our hypothesis that NLI models have naturally captured the linguistic ambiguity as long as it's well-calibrated. In such case, model predictions can truly reflect the correct human subjective distribution. Moreover, we find it's not always better to train with soft labels than hard ones, particularly in highly-certain data cluster. And the knowledge of linguistic ambiguity can be transferred across domains, benefiting low-resource setups. These takeaways are significant for future work in ambiguous NLI.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. This work was supported by The university of Melbourne, China Scholarship Council (CSC) and Huawei Translation Services Center.

[7] NLL is used for MedNLI because gold ambiguous labels is not available to calculate JSD. NLL varies with JSD in same trends empirically in all experiment results above.

References

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

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2020. Uncertain natural language inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8772–8779, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.

Dominik Maria Endres and Johannes E. Schindelin. 2003. A new metric for probability distributions. *IEEE Trans. Inf. Theory*, 49(7):1858–1860.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Weizhi Li, Gautam Dasarathy, and Visar Berisha. 2020. Regularization via structural label smoothing. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1453–1463. PMLR.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458. PMLR.

Johannes Mario Meissner, Napat Thumwanit, Saku Sugawara, and Akiko Aizawa. 2021. Embracing ambiguity: Shifting the training target of nli models. *arXiv preprint arXiv:2106.03020*.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. 2019. When does label smoothing help? In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

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, Online. Association for Computational Linguistics.

Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. 2016. Loss factorization, weakly supervised learning and label noise robustness. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 708–717. JMLR.org.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Trans. Assoc. Comput. Linguistics*, 7:677–694.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *The International Conference on Learning Representations (ICLR 2017)*.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1586–1596. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, and Qi Tian. 2016. Disturblabel: Regularizing CNN on the loss layer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4753–4762. IEEE Computer Society.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021. Capturing label distribution: A case study in NLI. *CoRR*, abs/2102.06859.

Xiang Zhou, Yixin Nie, and Mohit Bansal. 2021. Distributed NLI: learning to predict human opinion distributions for language reasoning. *CoRR*, abs/2104.08676.

Dataset	Split	Used By	#Samples	#Labels
SNLI	Train	UNLI	55,517	1r
	Dev.	UNLI	3,040	1r
		ChaosNLI	1,514	100
		AmbiS	9,842	5
	Test	UNLI	3,040	1r
		AmbiS	9,824	5
MNLI	Dev. M.	ChaosNLI	1,599	100
		AmbiM	9,815	5
	Dev. Mism.	AmbiM	9,832	5

Table 9: Datasets of ambiguous labels. "1r" denotes a regression label in the [0,1], refer to AmbiNLI original paper.

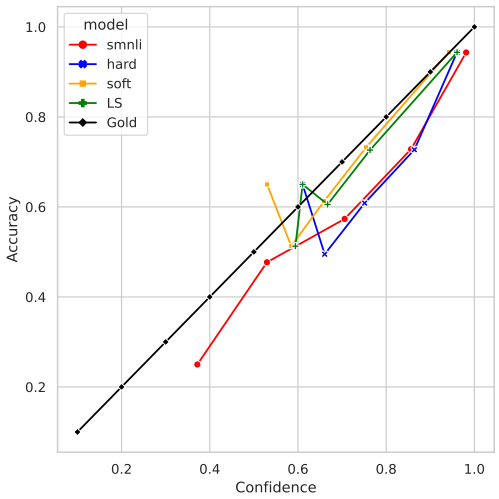

Figure 4: Reliability diagram for AmbiMNLI validation set.

A Appendices

A.1 Ambiguity Dataset Details

In the case of UNLI, AmbiNLI had taken only the 55,517 samples from the training set, so there is no overlap with ChaosNLI. They apply a simple linear approach to convert the UNLI regression value p into a probability distribution z_{NLI}, as described in the following composed function:

$$z_{\text{NLI}} = \begin{cases} (0, 2p, 1-2p) & p < 0.5 \\ (2p-1, 2-2p, 0) & p \geq 0.5. \end{cases}$$

The resulting AmbiNLI dataset has 18,152 SNLI examples, 18,048 MNLI examples, and 55,517 UNLI examples, for a total of 91,717 premise-hypothesis pairs with an ambiguity distribution as the target label.

Three datasets: ChaosNLI, UNLI and AmbiNLI are all derived from existing SNLI and MNLI, their

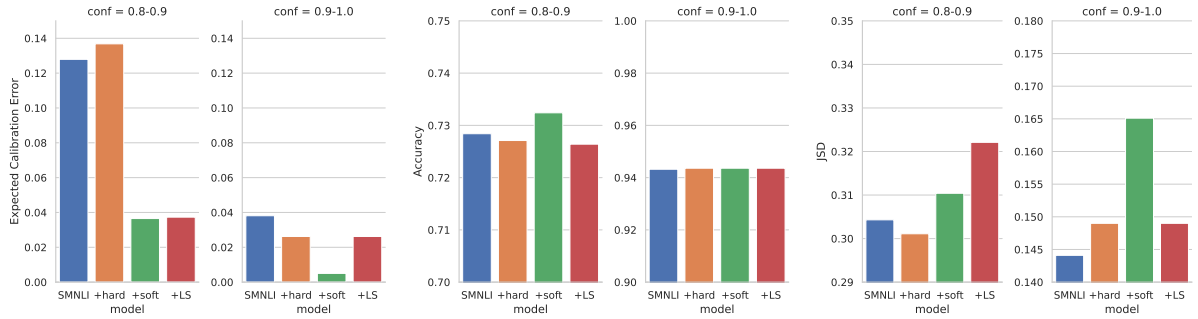

Figure 5: Expected calibration error (ECE), Accuracy and JSD of AmbiMNLI validation set.

Dataset	Chaos-SNLI						Chaos-MultiNLI					
Metrics	JSD ↓	Accuracy ↑	F1-macro ↑	NLL ↓	ECE ↓	sharpness	JSD ↓	Accuracy ↑	F1-macro ↑	NLL ↓	ECE ↓	sharpness
S/MNLI baseline	0.2443	0.7477	0.7365	0.7650	0.1338	0.8808	0.3432	0.5585	0.5566	1.4739	0.2971	0.8568
+ AmbiSM Gold	0.2606	**0.7596**	**0.7462**	0.7947	0.1512	0.9084	0.3476	0.5829	0.5756	1.4405	0.2921	0.8748
+ AmbiSM	**0.1918**	0.7543	0.7420	**0.5905**	0.0513	0.8036	0.2758	0.5816	0.5755	1.0306	0.2037	0.7863
+ AmbiSM (0.8)	0.2334	0.7576	0.7442	0.6498	**0.0286**	**0.7500**	0.2642	0.5822	0.5734	0.9814	0.1538	0.7353
+ AmbiSM (0.6)	0.2860	0.7517	0.7387	0.7682	0.1686	0.5811	0.2553	0.5804	0.5721	0.9467	**0.0179**	**0.5715**
+ AmbiSM (>0.6)	0.2421	0.7550	0.7410	0.6735	0.0421	0.7622	0.2734	0.5760	0.5676	1.0119	0.1689	0.7468
+ AmbiU Gold	0.3071	0.5859	0.5871	1.0871	0.2154	0.8025	0.3078	0.5266	0.5110	1.2413	0.2327	0.7618
+ AmbiU	0.2851	0.6017	0.6029	0.9989	0.1946	0.7432	0.2843	0.5253	0.5105	1.1718	0.1922	0.7188
+ AmbiU Filt.(p<0.05||p>0.97)	0.2311	0.6717	0.6608	0.7371	0.0493	0.6559	**0.2222**	0.5822	0.5706	**0.9070**	0.0600	0.6516
+ AmbiU soft.(p<0.05||p>0.97)	0.2674	0.6063	0.6075	0.8279	0.0861	0.6679	0.2415	0.5485	0.5361	0.9486	0.1046	0.6465
+ AmbiU soft.(p<0.10||p>0.90)	0.2707	0.6110	0.6121	0.8347	0.0864	0.6627	0.2425	0.5472	0.5330	0.9485	0.0973	0.6418
+ AmbiSMU Gold	0.2863	0.6281	0.6295	0.9415	0.1765	0.7932	0.3389	0.5779	0.5707	1.3831	0.2817	0.8609
+ AmbiSMU	0.2588	0.6301	0.6317	0.8389	0.1360	0.7334	0.2735	0.5785	0.5715	1.0246	0.2010	0.7785
+ AmbiSMU Filt.	0.2185	0.7087	0.7023	0.6823	0.0623	0.7078	0.2738	0.5841	**0.5770**	1.0296	0.1949	0.7803
+ AmbiSMU RS32924	0.2390	0.6796	0.6674	0.7501	0.0947	0.7360	0.2750	0.5810	0.5751	1.0338	0.1989	0.7808
+ AmbiSMU RS18000	0.2539	0.6374	0.6379	0.8134	0.1180	0.7353	0.2753	0.5804	0.5738	1.0348	0.1973	0.7796
+ AmbiSMU RS9000	0.2244	0.7107	0.7043	0.7015	0.0668	0.7470	0.2758	0.5822	0.5766	1.0374	0.1990	0.7829

Table 10: Results of AmbiNLI. Gold means that gold one-hot labels. Filt. indicates that extreme examples in UNLI have been filtered out. soft. means apply label smoothing (α=0.9) to examples satisfied the condition in brackets. RS-N=random selecting N examples from UNLI to include into training data.

relation can be see in Table 9, which is completely referred to (Meissner et al., 2021) Table 1. It can help readers to understand their overlap relation intuitively.

A.2 Evaluation Results of AmbiMNLI

In Figure 4, both gold ambiguity soft labels (soft-yellow line) and label smoothing (LS-green line) drag baseline (red line) much closer to the diagonal black line, while the hard label fine-tuning goes towards the opposite. This indicates label smoothing is as effective as gold ambiguity labels to calibrate models, and decreases ECE as show in left of Figure 5.

In addition, we find training on soft labels does not always lead to accuracy improvement and JSD decline, this particularly is exhibited in high confidence intervals. In conf=0.9-1.0, accuracy over all types of labels remains on a high level, but gold ambiguity labels has much high JSD. This is consistent with the findings in AmbiSNLI.

A.3 Results and Analysis of ChaosNLI

We reproduced results of AmbiNLI Table 2 in Table 10 below, and further did some ablation studies and analysis.

Random sampling of instance-specific soften factor α: In conventional label smoothing, all training examples are softened by a same fixed α, resulting in same probability of the target class, while the maximum probability of the crowdsourcing soft ambiguity label differs from each other. So we simulate a setting at risk of introducing much noise, where for each instance, they have an unique soften factor α which is uniformly sampled from the real interval (0.6. 1.0), i.e. the row "+AmbiSM (>0.6)" α is set greater than 0.6 because we assume gold label is agreed by at least 60% annotators. It's inferior to using gold ambiguous labels.

Discussion Ambi-UNLI: Using AmbiU, looking at the AmbiU and AmbiSMU results in Table 10, apparently UNLI data is not always beneficial. Specifically, it seems to worsen scores in all metrics except for ChaosMNLI accuracy. The origi-

nal paper supposes that UNLI's skewed distribution worsens scores. The distribution of labels in UNLI is drastically different from SNLI and MultiNLI, including textual topic, style and label type, when a model is fine-tuned on it, this distribution shift has a negative influence.

They found a very large number of samples with labels very close to 0 or 1, which translate into very extreme non-ambiguous distributions when converted. They confirm their hypothesis by filtering out all UNLI samples that had a probability label $p < 0.05$ or $p > 0.97$, and ran the "Filtered" experiments. Following this line, if it's due to extreme non-ambiguous distributions, label smoothing (α=0.8) over these samples should have obtained comparable results as filtering, even better because of more training data, but the fact is that it's better than +AmbiU, worse than +AmbiU Filt., enlarging the range of soft samples to $p < 0.1$ or $p > 0.9$ worsens more. This indicates filtering is occasionally a useful remedy in this setting, it does not result from non-ambiguous labels.

We further confirm this by randomly select (RS) subset of AmbiU to AmbiSM, i.e. AmbiSMU RS32924 [8], AmbiSMU RS18000, AmbiSMU RS9000, the number behind SR represents the size of the subset, we can see that the less AmbiU is incorporated, the better scores can be gained. Thus, UNLI data, under the current conversion approach, is somewhat problematic. We only apply AmbiSNLI and AmbiMNLI in the experiments.

[8] 32924 is the number of examples of AmbiU which is remained for after filtering by ($p<0.05$ || $p>0.97$), we randomly sample the same number of cases to make a comparison, result shows Filt. is better than RS.