Measuring Label Ambiguity in Subjective Tasks using Predictive Uncertainty Estimation

Richard Alies[1], Elena Merdjanovska[1,2] and Alan Akbik[1,2]
[1]Humboldt-Universität zu Berlin
[2]Science of Intelligence
rarichn@gmail.com, {elena.merdjanovska, alan.akbik}@hu-berlin.de

Abstract

Human annotations in natural language corpora vary due to differing human perspectives. This is especially prevalent in subjective tasks. In these datasets, certain data samples, i.e. annotatable instances, are more prone to label variation and can be indicated as ambiguous. This paper investigates methodologies for quantifying such label ambiguity by leveraging uncertainty estimation techniques when fine-tuning transformer-based models. We conducted experiments on three tasks characterized by subjective content and inherent label ambiguity: classifying sentiment, emotions and hate speech. The selected datasets include multi-annotator labels, which we use to derive a label ambiguity score for each data sample. This score is the entropy of the empirical probability distribution of annotator labels. The results indicate that uncertainty estimation techniques can measure label ambiguity to some extent. Deep Ensembles consistently outperform other techniques, increasing the correlation coefficients between model uncertainty and annotator disagreement, but the observed correlations are low. When comparing the annotator label distributions with the predicted class distributions, we see that Label Smoothing is able to notably reduce this difference, however a discrepancy still exists. This suggests that uncertainty estimation techniques improve the quantification of label ambiguity, however their ability remains limited, highlighting the need for further research [1].

1 Introduction

Natural language processing often relies on annotated corpora. Due to the subjective nature of language (Mohammad, 2016), annotation tasks often involve subjective judgments, where the meaning of text can be open to multiple interpretations due to personal perceptions, cultural backgrounds or contextual nuances. This subjectivity leads to label ambiguity, a phenomenon where different annotators assign different labels to the same piece of text, reflecting the inherent uncertainty in human language understanding (Mostafazadeh Davani et al., 2022; Khurana et al., 2025). This issue is particularly pronounced in applications requiring nuanced understanding of human emotions or opinions. For example, consider a movie review stating:

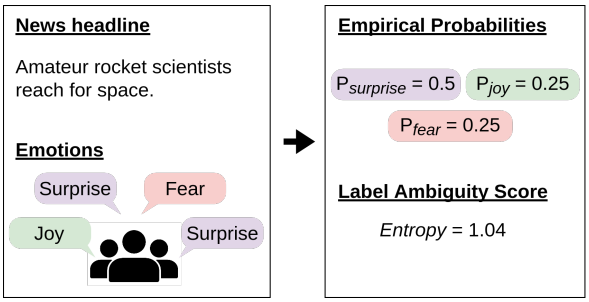


Figure 1: Example text snippet for emotion classification, showing the diverse emotion labels assigned by a group of annotators. Given these labels, we calculate the empirical probability distribution over classes. We use the characteristics of this distribution to define the *label ambiguity score* for the given text snippet.

> *"The film was surprisingly unconventional and thought-provoking."*

Some annotators might label this as *positive* due to its praise of originality, while others might perceive it as *negative* if they prefer traditional narratives. Such discrepancies highlight the difficulty in assigning definitive labels to subjective content (Plank et al., 2014b).

Current models excel in well-defined tasks with clear, objective labels, such as spam detection, where the distinction between spam and not-spam is relatively straightforward. However, they often underperform in subjective tasks due to their inability to account for label ambiguity (Pavlick and Kwiatkowski, 2019). These models tend to pro-

[1]Code available at: https://github.com/halra/raala

vide overconfident predictions even on inherently ambiguous samples, lacking mechanisms to reflect uncertainty in their outputs (Guo et al., 2017). This overconfidence can lead to misguided trust in the model's predictions and obscure the identification of samples, i.e. annotatable items, that require further human review or special attention (Zhang and Yang, 2021).

Furthermore, traditional evaluation metrics and training methodologies do not address the challenges posed by label ambiguity sufficiently (Beigman and Klebanov, 2009). Models are usually trained to minimize error, based on the assumption that there is a single correct label for each sample, which is not always the case in subjective tasks (Uma et al., 2021). This can result in models that are ill-equipped to handle the variability present in real-world data (Aroyo and Welty, 2015).

The core problem addressed in this paper is the lack of effective methodologies for detecting and quantifying label ambiguity in text classification models. Without proper identification and handling of ambiguous samples, models cannot differentiate between confidently correct predictions and those that are uncertain due to inherent ambiguity in the data. This limitation may hinder the development of reliable NLP systems capable of managing the complexities of human language interpretation, particularly in applications where understanding nuance and subjectivity is crucial.

To address this problem, the paper investigates whether techniques for estimating uncertainty in model predictions can serve as a means to measure label ambiguity.

Label ambiguity is often demonstrated in datasets with crowd-sourced annotations, which exhibiti varying degrees of annotator agreement. For instance, in the GoEmotions dataset — a corpus for fine-grained emotion classification (Demszky et al., 2020) — some text samples receive unanimous labels, while others have annotations spread across multiple emotion categories. The variance in annotations indicates the level of ambiguity for each sample. Traditional models might still assign high confidence to a single label, disregarding the underlying uncertainty reflected in the annotators' disagreement (Mostafazadeh Davani et al., 2022).

Given many annotators for each sample, we frame the *empirical probability distribution* over classes as a ground truth measure for sample-level ambiguity, as shown in Figure 1. This allows us to evaluate how well the sample-level uncertainty scores from various techniques align with ambiguity, by comparing them against the empirical probability distribution. In an additional ambiguity detection experiment, we define a threshold and have the models, equipped with stated uncertainty estimation techniques, predict which samples are ambiguous; samples with uncertainty scores within the threshold are marked as ambiguous.

Our contributions can be summarized as follows:

- We propose an empirical label ambiguity measure. This includes framing the annotator label distribution over classes as a ground truth measure for sample-level ambiguity.
- We evaluate uncertainty estimation techniques for measuring label ambiguity. These techniques are trained using a single label, and not a distribution, and we evaluate how well their output class distributions capture the inherent label ambiguity. We see that the techniques successfully improve over the Baseline Softmax in quantifying label ambiguity, but their performance is limited.
- We present an ambiguity detection task and evaluate the methods. We conduct experiments that classify samples as ambiguous based on defined uncertainty thresholds, demonstrating modest improvements over standard fine-tuning and random baselines.

2 Evaluation Data for Label Ambiguity

In this section, we outline the evaluation data and metrics employed to investigate label ambiguity in subjective tasks. We utilize publicly available datasets with inherent annotation ambiguity, each annotated with multi-annotator labels, described in Section 2.1. We define the *label ambiguity score* as the entropy of the empirical probability distribution over annotator labels, explained in Section 2.2.

2.1 Datasets

We employ publicly available datasets with multi-annotator labels, which demonstrate annotator disagreements. In our experiments, we utilize GoEmotions (Demszky et al., 2020), Rotten Tomatoes Reviews (Pang and Lee, 2005), and the GAB Hate Speech Corpus (Kennedy et al., 2020). For each dataset we used 70% for training, 15% as validation and 15% as a holdout test set.

Table 1 summarizes the dataset characteristics. This includes the original characteristics of each

		Samples	Classes	Annotators
GoEmotions	*orig.*	58,009	28	4.3
	modif.	23,990	9	2.8
Rotten Tomatoes	*orig.*	4,999	2	5.55
	modif.	4,999	2	5.55
GAB Hate Speech	*orig.*	27,665	13[1]	3+
	modif.	4,674	2	3.12

Table 1: Overview of the three datasets. The columns show the total number of samples, number of classes and average number of annotators per sample.

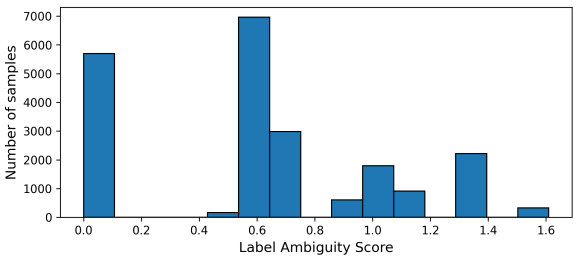
dataset, as well as the modified ones used in this paper. Following are the modifications we applied:
GoEmotions: We reduced the label set to 9 primary emotions: sadness, neutral, love, gratitude, disapproval, amusement, adminration, annoyance, approaval. We also removed examples with only one annotator vote, and balanced the dataset across classes.
GAB Hate Speech: We consolidated the multiple hate categories into a binary hate label and balanced the resulting subset. Merging all hate categories into one class brings more variety into the hate class, which induces more disagreements than according to the original label set.

2.2 Label Ambiguity Score

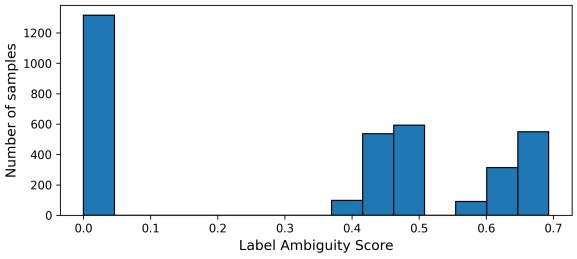
We define the label ambiguity score using empirical probability distributions. These distributions consist of empirical probabilities for each class computed using labels from multiple human annotators. The empirical probabilities are computed as the proportion of annotators who choose that class relative to the total number of annotators. This distribution reflects annotator consensus and allows us to compute the label ambiguity score, given that ambiguous samples exhibit higher disagreement among annotators.

We use the entropy of this distribution as a *label ambiguity score*, calculated for each dataset example. Higher entropy indicates greater disagreement among annotators and ambiguity, whereas lower entropy corresponds to stronger consensus.

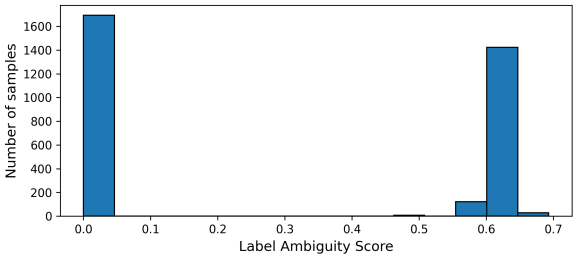
We analyse the distribution of label ambiguity scores for each dataset in Figure 2. We can see that the GoEmotions and Rotten Tomatoes datasets have wide distributions, with the data samples exhibiting either total agreement (label ambiguity score close to zero), or different levels of ambiguity. The high label ambiguity scores in GoEmotions overall, larger than 1, are due to the larger number of classes, whereas Rotten Tomatoes and GAB Hate Speech have only two classes. For GAB Hate Speech, we see a bimodal histogram with two very narrow peaks, indicating two very distinct groups of samples - low ambiguity around 0 or high ambiguity around 0.6.



(a) GoEmotions



(b) Rotten Tomatoes



(c) GAB Hate Speech

Figure 2: Distribution of label ambiguity scores

3 Methods

We describe our methodology for uncertainty estimation to assess label ambiguity. Our goal is to use uncertainty estimation either to directly predict the label ambiguity score or to approximate the full label distribution across classes. We detail the uncertainty estimation techniques employed in Section 3.2, and explain how we derive an uncertainty score from the model outputs in Section 3.3.

3.1 Baseline Softmax and Oracle Softmax Distribution

First, we will briefly explain the standard fine-tuning approach for classification, used as a base-

[2]Total number including various types of hate speech.

line in our paper.

Baseline Softmax. In this approach, the target labels used are the *majority vote* of the multi-annotator labels. This means that the model is trained on one-hot encoded labels where each sample is assigned exactly one class - the most frequent one of the crowd annotations. The model outputs a softmax distribution (Bridle, 1990) over the classes, which can be interpreted as a probability distribution. This predicted distribution is used to later calculate the uncertainty score.

Additionally, we include another standard approach, that is common when dealing with multi-annotator datasets (Plank et al., 2014a).

Oracle Softmax. Instead of the majority vote, this approach uses soft training labels, obtained from the *full distribution of annotations*. The frequency of annotator votes for each class is used as a corresponding soft label. This represents an ideal scenario where the distribution of human annotator labels for the training samples is known. Again, the softmax distribution is used to calculate the uncertainty score.

The goal of this paper is to measure label ambiguity when annotator distributions are in fact not available and all of our evaluated approaches train with a single label for each sample. This makes the *Oracle Softmax* approach infeasible, however we include it as an upper performance bound, because it could inform us on the potential of ambiguity quantification when richer labels are available.

3.2 Uncertainty Estimation Techniques

We focus on three techniques: *Monte Carlo Dropout*, *Deep Ensembles* and *Label Smoothing*. These techniques all involve fine-tuning models for classification, using the majority vote of the multi-annotator labels and no additional information about the annotator distribution.

Deep Ensemble (DE) involves training multiple neural networks independently, each initialized differently (Lakshminarayanan et al., 2017). In our case, we use multiple instances of the same model architecture, which are just multiple instances of the previously explained *Baseline Softmax*. Each of these models outputs a predicted distribution over classes. We use the average of these distributions to calculate the uncertainty score.

Monte Carlo Dropout (MCD) is a method used for estimating uncertainty in neural network predictions (Gal and Ghahramani, 2016). By randomly disabling neurons during inference, it provides multiple stochastic predictions that help measure model uncertainty. We use the average of these predicted distributions to calculate the uncertainty score.

Label Smoothing (LS) is a technique that modifies the target labels to reduce model overconfidence by assigning soft probabilities to non-target labels (Szegedy et al., 2015). Instead of using hard one-hot encoded labels, we uniformly distribute a fraction of the label probability mass across other classes which helps mitigate overfitting. Similar to the other methods, the output softmax distribution is used to calculate the uncertainty score.

3.3 Uncertainty Score

Each uncertainty estimation technique outputs a predicted probability distribution over the classes. Given this probability distribution, we calculate its entropy as an *uncertainty score*. Entropy quantifies the amount of uncertainty or randomness in a probability distribution (Namdari and Li, 2019).

In addition to the entropy, we can calculate other uncertainty metrics, such as variance and the Jensen-Shannon divergence (JSD). We initially experimented with all three of them, however our results showed that they perform very similarly. The comparison of the three uncertainty metrics for the task of ambiguity detection can be found in Appendix A. Due to this, we only use entropy in the remainder of this paper.

4 Experiment: Measuring Label Ambiguity

In the first experiment, we evaluate the effectiveness of the uncertainty estimation techniques in measuring label ambiguity. Here, we compare how correlated the ambiguity and uncertainty scores are, as well as how close the empirical and predicted distributions are.

4.1 Experimental Setup

We compare the three uncertainty estimation techniques (Section 3.2) with the *Baseline Softmax* and *Oracle Softmax* fine-tuning. We perform the experiment using three datasets, listed in Section 2.1.

We selected well-known models that have consistently demonstrated robust performance across natural language processing tasks. Namely BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2020). Table 2 provides a high-level overview of the key specifications for BERT, RoBERTa, and XLNet. Despite differences

in training strategies and data volumes, all three models share a transformer-based architecture. By employing three different models we can verify the generalizability of our findings.

	BERT	RoBERTa	XLNet
Vocab. size	30,522	50,265	32,000
Max. seq. length	512	512	512
Training data	16GB	160GB+	158GB+
Pre-train object.	MLM, NSP	MLM	Permut. LM

Table 2: Comparison of Architectural Specifications

All experiments were ran for 3 random seeds and tables show the mean scores and standard deviations. Further implementation details can be found in Appendix C.

We calculate multiple metrics to evaluate how well the techniques measure ambiguity. To compare the scores themselves, we calculate the Pearson correlation coefficient between the predictive entropies (uncertainty scores) and the empirical entropies (label ambiguity scores). A high correlation indicates that the model's uncertainty estimates align with human perceptions of ambiguity, suggesting that the model can effectively identify ambiguous samples.

To compare the empirical and predicted distributions directly, we calculate the Jensen-Shannon divergence (JSD), Kullback–Leibler divergence (KLD) and mean squared error - averaged over all classes (MSE). With this, for each sample, we evaluate how close the distribution of predicted class probabilities is to the empirical distribution of the annotator labels. These metrics are calculated for each sample independently, and then averaged over samples.

4.2 Results - Baseline Softmax

The classification metrics for the *Baseline Softmax* model can be found in Appendix B. We see that the three tasks have different difficulty levels. The F1 score for sentiment classification (Rotten Tomatoes) is the highest - 0.87, followed by hate speech classification (GAB Hate Speech) with 0.77 and emotion classification (GoEmotions) with 0.64. Additionally, we see that the scores on each dataset are consistent across the three transformer models.

Additionally, we compare the most common cases of disagreements in the models' predictions and the human annotations. On the GoEmotions dataset we compare the classifier's confusion matrix with human annotation co-occurence counts. Half of the ten most frequent pairs *neutral* ↔ *approval*, *neutral* ↔ *disapproval*, *neutral* ↔ *sadness*, and *annoyance* ↔ *disapproval* appear in *both* rankings, giving a 50% overlap. This shows that the models often make prediction mistakes exactly where annotators tend to attribute multiple emotions, which means these mistakes can be attributed to annotator disagreement and label variation. On another hand, the remaining pairs in Table 3a) are class distinctions genuinely difficult for the model.

The complete confusion and co-occurrence heatmaps are shown in Figure 6 in Appendix E.

4.3 Results - Measuring Label Ambiguity

Table 4 shows our aggregated results—averaged over the three model architectures.

As expected, *Oracle Softmax* has the highest correlation and lowest JSD, KLD and MSE out of all the methods. The average correlations for *Oracle Softmax* are in the range 0.290 - 0.375 across all datasets and models, indicating moderate correlation (Hopkins, 2000). This is expected, since it incorporates annotator distribution information during training, while the other techniques do not. A minor exception is the GoEmotions dataset, where even though the *Oracle Softmax* method achieves the lowest MSE and highest correlation, its relatively higher JSD and KLD suggest that, while it minimizes squared differences, it does not fully capture the distribution. One reason for this could be the larger number of classes in GoEmotions, compared to the other two datasets.

In all cases, all uncertainty estimation techniques improve over *Baseline Softmax*. The *Deep Ensemble* technique achieves the highest mean correlation coefficients of 0.218 and 0.212 for GoEmotions and Rotten Tomatoes. *Monte Carlo Dropout* also shows substantial improvement, with average correlations of 0.216 and 0.167 for GoEmotions and Rotten Tomatoes.

On the GAB Hate Corpus, we generally observe much lower correlations than for the other two datasets. One potential reason for this could be the very narrow peaks in the histogram of this dataset (see Figure 2) when compared to the other two, which means that this dataset includes a very limited variety of label ambiguity scores. Additionally, for this dataset we applied the most significant modification, which was changing the target into binary classification (hate or no hate), by merging all various hate classes into one.

Overall, our results suggest that using uncer-

Rank	Pair	Count
1	neutral ↔ approval	74
2	annoyance ↔ disapproval	62
3	approval ↔ neutral	56
4	neutral ↔ disapproval	55
5	annoyance ↔ neutral	47
6	neutral ↔ annoyance	47
7	disapproval ↔ neutral	46
8	approval ↔ admiration	45
9	neutral ↔ sadness	41
10	disapproval ↔ annoyance	38

(a) Classifier confusion pairs.

Rank	Pair	Count
1	neutral ↔ approval	226
2	approval ↔ neutral	226
3	sadness ↔ neutral	159
4	neutral ↔ sadness	159
5	neutral ↔ disapproval	151
6	disapproval ↔ neutral	151
7	annoyance ↔ neutral	143
8	neutral ↔ annoyance	143
9	annoyance ↔ disapproval	116
10	disapproval ↔ annoyance	116

(b) Human co-occurrence pairs.

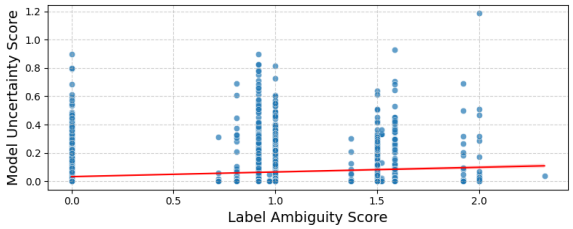
Table 3: Most frequent emotion pairs in the misclassifications of the baseline classifier (left) and in the human co-annotations (right) on the 9-class GoEmotions dataset.

tainty scores derived from uncertainty estimation techniques, particularly *Deep Ensembles* and *MC Dropout*, enhance the model's ability to detect ambiguous samples. However, it is important to note that the correlation coefficients between the uncertainty and ambiguity scores are low, with values close to 0.2, indicating that while there is a positive relationship, it is small (Hopkins, 2000). This suggests that the techniques' ability to detect ambiguity is limited and there is room for improvement.

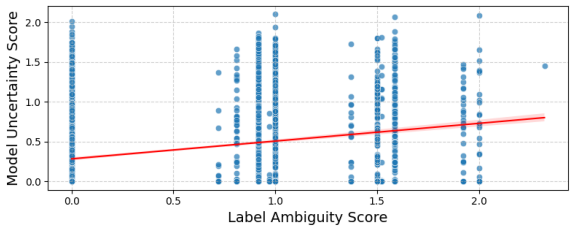
When comparing the distributions, *Label Smoothing* significantly reduces the discrepancy between the predicted and annotator distributions, much better than *Deep Ensemble* and *Monte Carlo Dropout*. This is opposite from the correlation analysis, where in terms of overall correlation of entropies, *Label Smoothing* scores much lower than the other methods. With this, we see that training with soft labels significantly improves the predicted class distributions and makes them more ambiguity-aware, even when the soft labels are only in the form of a uniform smoothing factor.

Figure 3 showcases the improvement the *Deep Ensemble* brings over the *Baseline Softmax*, by visualizing the correlation across all data samples on the GoEmotions dataset. The scatter plots show that the *Deep Ensemble* technique results in a stronger positive correlation, with data points more closely following an upward trend compared to the baseline. This highlights the finding that the uncertainty score derived from ensembles of models improves the measuring of label ambiguity, as opposed to using a single model.

As an additional insight, for BERT on Rotten Tomatoes we selected the top-100 most-uncertain sentences for MC Dropout, Deep Ensemble, and Label-Smoothing. Eighteen sentences (18 %) occur



(a) *Baseline*: Correlation 0.095



(b) *Deep Ensemble*: Correlation 0.226

Figure 3: Correlation between label ambiguity scores and uncertainty scores across all data samples. Results for the GoEmotions dataset using XLNet.

in *all* three lists, and the pair-wise Jaccard overlaps average 0.24 ± 0.01. Across the entire score vectors the mean Spearman correlation is 0.50 ± 0.20 (after aligning on common IDs). Each estimator nonetheless brings novel evidence: 39%, 43%, and 40% of their respective top-100 sentences are unique to MC, Smoothing, and DE.

5 Experiment: Detecting Ambiguous Samples

This experiment demonstrates our methodology for detecting ambiguous samples in text classification using model uncertainty estimates. We apply percentile-based thresholds and flag samples that exceed these thresholds. With this, we assess the overlap between model-identified and annotator-identified ambiguous samples and evaluate how

Dataset	Technique	Distribution Mean JSD ↓	Distribution Mean KLD ↓	Distribution Mean MSE ↓	Ambiguity Score Correlation ↑	Ambiguity Score % Improv. ↑
GoEmotions	Baseline Softmax	0.342 ± 0.005	5.303 ± 0.440	0.0608 ± 0.0009	0.084 ± 0.007	-
	Deep Ensemble	**0.285 ± 0.002**	3.271 ± 0.050	0.0443 ± 0.0003	0.218 ± 0.007	163%
	MC Dropout	0.294 ± 0.002	2.799 ± 0.039	0.0478 ± 0.0003	0.216 ± 0.003	161%
	Label Smoothing	0.340 ± 0.002	**1.115 ± 0.007**	0.0407 ± 0.0004	0.155 ± 0.012	87%
	Oracle Softmax	0.382 ± 0.006	1.489 ± 0.042	**0.0125 ± 0.0003**	**0.375 ± 0.009**	**354%**
Rotten Tomatoes	Baseline Softmax	0.150 ± 0.002	2.662 ± 0.102	0.1174 ± 0.0027	0.081 ± 0.015	-
	Deep Ensemble	0.115 ± 0.002	1.788 ± 0.051	0.0880 ± 0.0017	0.212 ± 0.009	174%
	MC Dropout	0.125 ± 0.005	1.754 ± 0.093	0.0989 ± 0.0045	0.167 ± 0.020	122%
	Label Smoothing	0.084 ± 0.003	0.245 ± 0.009	0.0745 ± 0.0033	0.135 ± 0.010	78%
	Oracle Softmax	**0.070 ± 0.003**	**0.208 ± 0.013**	**0.0543 ± 0.0024**	**0.290 ± 0.020**	**279%**
GAB Hate Speech	Baseline Softmax	0.208 ± 0.003	3.262 ± 0.224	0.1794 ± 0.0032	0.036 ± 0.043	-
	Deep Ensemble	0.165 ± 0.002	1.922 ± 0.078	0.1390 ± 0.0019	0.073 ± 0.013	185%
	MC Dropout	0.176 ± 0.004	1.970 ± 0.107	0.1536 ± 0.0036	0.084 ± 0.033	173%
	Label Smoothing	0.132 ± 0.003	0.381 ± 0.009	0.1205 ± 0.0039	0.046 ± 0.033	65%
	Oracle Softmax	**0.104 ± 0.010**	**0.355 ± 0.048**	**0.0916 ± 0.0109**	**0.375 ± 0.031**	**1075%**

Table 4: Evaluation of the experiment of measuring label ambiguity. Three distribution metrics: Jensen-Shannon divergence (JSD), Kullback–Leibler divergence (KLD) and mean squared error (MSE) are shown. The Pearson correlation coefficients of the uncertainty and ambiguity scores are also shown, together with percentage improvement over the *Baseline Softmax* (%Improv.), in terms of the correlations. The scores are averaged over all test set samples, and then averaged over the three models. The table shows mean ± std., where the standard deviation is calculated over the models. In each column, the best scores are **bolded**, and the second-best are underlined.

Metric	Value
Common to all three	18 / 100 (18%)
Mean Jaccard	0.24 ± 0.01
Mean Spearman ρ	0.50 ± 0.20
Unique to MC Dropout	39 %
Unique to Label Smoothing	43 %
Unique to Deep Ensemble	40 %

Table 5: Overlap statistics for the top–100 most-uncertain Rotten-Tomatoes items.

well our model-derived uncertainty works for detecting human ambiguity.

The first experiment, gives us correlation coefficients which are positive, but low. This does not tell us what these values imply for the practical use of these methods. With this second experiment, we hope to get better insights into whether these correlation values are sufficient to guide downstream filtering of ambiguous samples.

5.1 Task Setup

With this experiment, we transform the task into a binary classification task, where the two classes are *ambiguous* and *non-ambiguous*. We refer to this setup as ambiguity detection. We assign ground truth labels based on the label ambiguity scores. A sample is labeled as *ambiguous* if its label ambiguity score exceeds a pre-defined threshold.

We set this threshold dynamically, to always match the 60th percentile of the ambiguity scores. We chose this threshold as it has been adopted in some prior works with limited backing (Dumitrache et al., 2015). Intuitively, in Figure 2, we see that applying a dataset-specific threshold using the 60th percentile, would result in a large number of samples flagged as ambiguous. This is confirmed in Table 6, where we see that the shares of ambiguous samples are close to 50%[3]. In other words, we flag as ambiguous almost all samples that do not have perfect agreement among the annotators.

This is one way to separate samples into two classes according to their annotator agreement scores. In reality, determining this threshold and defining the difference between ambiguous and non-ambiguous samples is a very significant question, but also challenging to answer and out of the scope of this paper.

During inference, we apply the same type of thresholding using the 60th percentile to the model-

[3]The 60th percentile threshold implies that 40% of the samples will be flagged. However, with 2–5 annotators per item, ambiguity scores are limited to a few possible values. For some datasets, like GAB Hate Speech, this includes a lot of ties, which raises the ambiguous shares to over 40%, but avoids arbitrarily splitting items with identical agreement.

derived uncertainty scores. This determines the predicted label for each sample: if the uncertainty score is above the threshold the sample is predicted as ambiguous.

5.2 Random Baseline

For this task, we also include a random baseline in the evaluations. Here, instead of calculating an uncertainty score, we randomly generate a number between 0 and 1 for each sample. Then, on these random scores we apply the same threshold as explained in the previous section: if the random score is above the threshold the sample is predicted as ambiguous. This helps us assess the practical effectiveness of the uncertainty techniques in detecting ambiguous samples.[4]

5.3 Results

The main results of this experiment, in terms of error rates, are shown in Table 6. We can see that all methods consistently outperform the *Random baseline*, which has error rates of around 50%. This indicates that all methods are helpful in flagging ambiguous samples.

Out of the techniques, and consistent with our previous experiments, *Deep Ensemble* achieved the lowest error rates, with average of 41.19%. Notably, these rates are promising when compared to a *Random Baseline*, indicating that our techniques capture meaningful predictive information. We obtained comparable scores across the three datasets. On the GoEmotions dataset, all three techniques outperformed the *Baseline Softmax*, whereas on the Rotten Tomatoes and GAB Hate Speech datasets, *Label Smoothing* and *Monte Carlo Dropout* performed worse than the *Baseline Softmax*. The *Oracle Softmax* approach again provided an advantage by reducing the average error rate to around 37%.

In Figure 4, we present the ROC curves of the ambiguity detection task. The ROC curves illustrate the trade-off between the true positive rate and the false positive rate at various threshold settings.

Out of the methods, the *Deep Ensemble* exhibits the highest area under the curve (AUC) of 0.61, indicating the best overall performance where *Monte Carlo Dropout* performs slightly below Deep Ensemble but still surpasses the *Baseline Softmax* and Label Smoothing techniques. All four methods outperform the Random baseline.

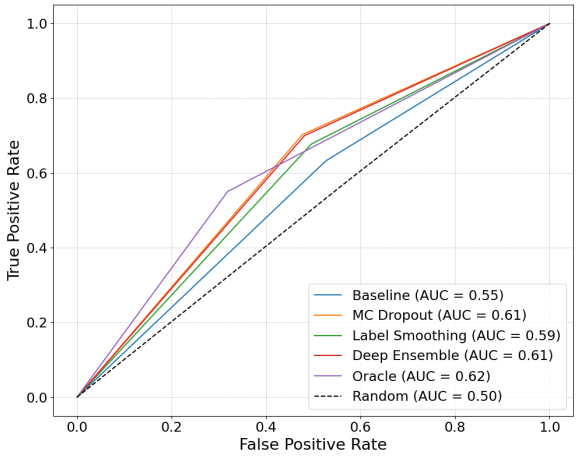


Figure 4: ROC curves for ambiguity detection, on the GoEmotions dataset with RoBERTa. Each sample is annotated as ambiguous if the empirical entropy (label ambiguity score) is over 60% of the maximum value.

These results are consistent with our previous analysis, reinforcing the conclusion that the *Deep Ensemble* technique is more adept at capturing label ambiguity.

6 Related Work

There have been numerous studies addressing human label variation and label ambiguity. Snow et al. (2008) highlighted the variability in annotations obtained from non-expert annotators and the impact of this variability on NLP tasks. They demonstrated that aggregating multiple annotations can improve the quality of labels.

Another study proposed leveraging annotator disagreement instead of resolving it, suggesting that disagreement can provide valuable information.They advocated for models that learn from soft labels reflecting annotator probabilities rather than hard labels (Plank et al., 2014a). We include this as our Oracle Softmax approach.

Uncertainty estimation techniques have gained attention as a means to quantify model confidence (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017). In the context of deep learning, methods such as Monte Carlo Dropout (Gal and Ghahramani, 2016) approximate Bayesian inference by performing dropout at inference time, enabling models to estimate predictive uncertainty. Similarly, Deep Ensembles (Gal and Ghahramani, 2016) improve uncertainty estimation by training multiple models with different initializations and aggregat-

[4] An alternative random baseline is to always output the majority class (non-ambiguous). This will result in error rates equal to the share of ambiguous samples, which are sometimes better than the random baseline we use. However, this would also give us a zero precision and recall scores of the class of interest, making it unusable for this task.

	GoEmotions	Rotten Tomatoes	GAB Hate Speech	Average
%Ambiguous	53.81	42.80	45.93	-
Error Rate (%)				
Random	51.52 ± 0.61	52.34 ± 0.77	50.21 ± 0.25	51.36
Baseline Softmax	45.01 ± 1.75	41.64 ± 1.23	44.13 ± 2.84	43.59
Deep Ensemble	40.90 ± 0.29	39.75 ± 0.57	42.91 ± 3.57	41.19
Monte Carlo Dropout	40.73 ± 0.37	42.79 ± 0.68	45.76 ± 2.91	43.09
Label Smoothing	42.83 ± 0.68	45.73 ± 1.78	47.99 ± 2.95	46.18
Oracle Softmax	**37.62 ± 0.49**	**37.13 ± 1.10**	**37.39 ± 1.09**	**37.38**

Table 6: Ambiguity rates and error rates (mean ± std) for ambiguity detection. The results are averaged over the three models. In each column, the best scores are **bolded**, and the second-best are underlined.

ing their predictions.

These techniques have shown effectiveness in improving model calibration and detecting out-of-distribution samples. Bley et al. (2024) evaluated various uncertainty estimation methods under dataset shift and found that ensembles generally provide better calibration and uncertainty estimates compared to single models.

Malinin and Gales (2018) introduced Prior Networks to model predictive uncertainty, distinguishing between data uncertainty and model uncertainty in text classification tasks.

Recent research has begun to explore the relationship between model uncertainty and label ambiguity. Braiek and Khomh (2024) studied how incorporating human-like uncertainty into models can improve robustness in image classification tasks. They showed that models trained with uncertain labels can better handle ambiguous inputs.

Despite these advancements, there is limited work specifically focusing on leveraging uncertainty estimation techniques to detect label ambiguity arising from annotator disagreement in subjective text classification.

7 Conclusion

In this paper, we focused on three subjective tasks of great interest: sentiment, emotion, and hate speech classification. For each task, we used public datasets with published multi-annotator labels. For every sample in these datasets, we defined a label ambiguity score as the entropy of the annotator label distribution, which measures the inherent randomness in the labeling process.

We assessed the effectiveness of uncertainty estimation in quantifying label ambiguity. Our evaluation included three techniques—Deep Ensemble, Monte Carlo Dropout, and Label Smoothing—which we compared with both a Baseline Softmax model and an Oracle Softmax approach, the latter serving as an upper performance bound. For each method, we computed an uncertainty score defined as the entropy of the predicted label distribution.

First, we evaluated whether predictive uncertainty techniques could effectively capture label ambiguity by calculating the correlation between uncertainty scores and label ambiguity scores. Our findings indicate that these techniques—most notably Deep Ensembles—outperform the Baseline Softmax approach, with both Deep Ensembles and Monte Carlo Dropout showing a low positive correlation with label ambiguity. Additionally, we assessed the alignment between predicted class distributions and annotator class distributions. Here, the Label Smoothing approach was successful in reducing the discrepancy between the distributions, making the predictions more ambiguity-aware.

Next, we applied the uncertainty estimation techniques to an ambiguity detection task, classifying each sample as either ambiguous or non-ambiguous using a fixed threshold. Under these conditions, the Deep Ensemble approach achieved an error rate of about 40%, reducing it when compared to the Baseline Softmax approach.

Our results indicate that when fully leveraging annotator labels, as in the Oracle Softmax fine-tuning, the models' ability to quantify ambiguity improves, but the performance improvements remain modest. Although the current uncertainty estimation techniques do not perfectly capture all aspects of label ambiguity, the findings are promising and indicate further research in this direction is needed. We believe this paper can provide a foundation for future research into more robust and effective methods for quantifying label ambiguity.

Limitations

Several limitations of our study should be acknowledged. First, our experiments were primarily conducted on the GoEmotions, Rotten Tomatoes and GAB Hate Corpus datasets, which, while extensive and diverse, may not capture all nuances of subjective expressions across different cultures, languages or contexts.

Second, uncertainty estimation techniques like Deep Ensembles require training multiple models, increasing computational complexity and resource requirements. This may limit their practicality in environments with constrained resources or real-time processing needs. While uncertainty estimation techniques provide valuable information about model confidence, interpreting these estimates in a meaningful way for end-users remains a challenge.

And third, we focus on single-label classification which has inherent limitations as opposed to multi-label classification and may not be the most suitable for tasks such as emotion classification.

Acknowledgments

Elena Merdjanovska and Alan Akbik are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135. Alan Akbik is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Emmy Noether grant "Eidetic Representations of Natural Language" (project number 448414230).

References

Lora Aroyo and Chris Welty. 2015. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24.

Eyal Beigman and Beata Beigman Klebanov. 2009. Learning with annotation noise. In *ACL-IJCNLP*.

Florian Bley, Sebastian Lapuschkin, Wojciech Samek, and Grégoire Montavon. 2024. Explaining predictive uncertainty by exposing second-order effects. *Preprint*, arXiv:2401.17441.

Houssem Ben Braiek and Foutse Khomh. 2024. Machine learning robustness: A primer. *Preprint*, arXiv:2404.00897.

John S Bridle. 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, architectures and applications*, pages 227–236. Springer.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, 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. *Preprint*, arXiv:1810.04805.

Anca Dumitrache, Lora Aroyo, and Chris Welty. 2015. Crowdtruth measures for language ambiguity: The case of medical relation extraction. In *LD4IE@ISWC*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Preprint*, arXiv:1506.02142.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. *Preprint*, arXiv:1706.04599.

W.G. Hopkins. 2000. *A New View of Statistics*. Internet Society for Sport Science.

Brendan Kennedy, Mohammad Atari, Aida Mostafazadeh Davani, Leigh Yeh, Ali Omrani, Yehsong Kim, Kris Koombs, Shreya Havaldar, G J Portillo-Wightman, Elaine Gonzalez, Joe Hoover, Aida Azatian, Alyzeh Hussain, Austin Lara, Gabriel Olmos, Adam Omary, Christina Park, Clarisa Wang, Xin Wang, and Morteza Dehghani. 2020. The gab hate corpus: A collection of 27k posts annotated for hate speech.

Urja Khurana, Eric Nalisnick, and Antske Fokkens. 2025. DefVerify: Do hate speech models reflect their dataset's definition? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4341–4358, Abu Dhabi, UAE. Association for Computational Linguistics.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Preprint*, arXiv:1612.01474.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Andrey Malinin and Mark Gales. 2018. Predictive uncertainty estimation via prior networks. *Preprint*, arXiv:1802.10501.

Saif Mohammad. 2016. A practical guide to sentiment annotation: Challenges and solutions. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 174–179, San Diego, California. Association for Computational Linguistics.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Alireza Namdari and Zhaojun (Steven) Li. 2019. A review of entropy measures for uncertainty quantification of stochastic processes. *Advances in Mechanical Engineering*, 11(6):1687814019857350.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

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

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014a. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 742–751, Gothenburg, Sweden. Association for Computational Linguistics.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014b. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511, Baltimore, Maryland. Association for Computational Linguistics.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. *Preprint*, arXiv:1512.00567.

Alexandra Uma, Tommaso Fornaciari, Anca Dumitrache, Tristan Miller, Jon Chamberlain, Barbara Plank, Edwin Simpson, and Massimo Poesio. 2021. SemEval-2021 task 12: Learning with disagreements. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 338–347, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding. *Preprint*, arXiv:1906.08237.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *Preprint*, arXiv:1707.08114.

A Comparison of Uncertainty Metrics - Ambiguity Detection

In this paper we use entropy as an uncertainty score, however, we experimented with variance and JSD (Jensen-Shannon divergence between uniform and a given distribution). Figure 5 shows ROC curves for different techniques, for the ambiguity detection task. For MC Dropout and Deep Ensemble, we also see different variants of the uncertainty score. We see that all three variants (JSD, variance and entropy) behave similarly across all thresholds, which is why we chose to use one of them throughout the paper.

B Baseline Softmax Results

Table 7 shows the classification metrics of the Baseline Softmax fine-tuning runs.

C Implementation Details

We fine-tuned three transformer-based models: BERT (`bert-base-uncased`) (Devlin et al., 2019), RoBERTa (`roberta-base`) (Liu et al., 2019) and XLNet (`xlnet-base-cased`) (Yang et al., 2020).

Consistent hyperparameters were used across all experiments to ensure fair comparisons and isolate the effects of the uncertainty estimation techniques:

- Seeds: [42, 13, 815]
- Seeds for Deep Ensamble: [[42, 13, 815, 142, 113], [142, 113, 1815, 1142, 1113], [242, 213, 2815, 2142, 2113]]
- Optimizer: AdamW (Loshchilov and Hutter, 2019)
- Learning Rate: 5×10^{-5}
- Batch Size: 8
- Number of Epochs:
  - 14 epochs for Baseline Softmax, MC Dropout and Label Smoothing experiments
  - And [10, 11, 13, 14, 15] epochs for Deep Ensembles to introduce diversity among ensemble members

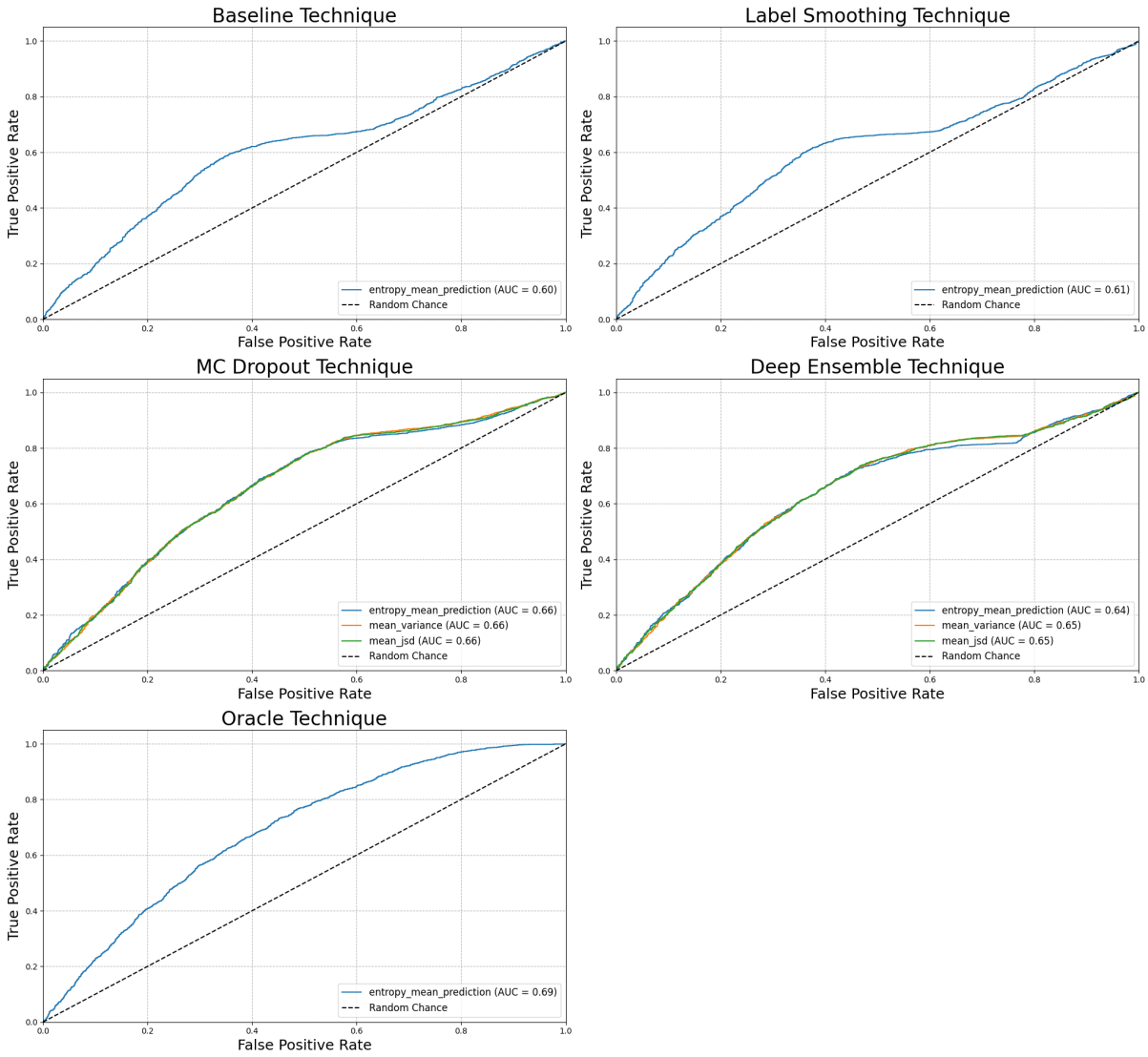


Figure 5: XLNet with GoEmotions ROC/AUC

Model	Dataset	Precision	Recall	F1 Score	Accuracy
RoBERTa	Rotten Tomatoes	0.87	0.87	0.87	0.87
	GoEmotions	0.64	0.64	0.64	0.64
	GAB Hate Corpus	0.78	0.78	0.78	0.78
BERT	Rotten Tomatoes	0.85	0.85	0.85	0.85
	GoEmotions	0.64	0.65	0.64	0.65
	GAB Hate Corpus	0.77	0.77	0.77	0.77
XLNet	Rotten Tomatoes	0.88	0.87	0.87	0.87
	GoEmotions	0.64	0.64	0.64	0.64
	GAB Hate Corpus	0.77	0.77	0.77	0.77

Table 7: Classification Metrics for the Baseline Softmax Models

- Dropout Rate: 0.1

Specific parameters for each uncertainty estimation technique were:

- Monte Carlo Dropout:
  - Number of Stochastic Forward Passes during inference: 100
  - Dropout enabled during inference
  - Dropout during inference: 0.5
- Deep Ensembles:
  - Ensemble Size: 5 models
  - Different random seeds and epochs for each ensemble member
- Label Smoothing:
  - Smoothing Factor: $\epsilon = 0.3$

We split each dataset into training, validation and test sets using a 70/15/15 stratified split to maintain class distribution.

D Correlation between Ambiguity and Uncertainty Scores

Table 8 shows the correlation coefficients and percentage improvement over baseline, averaged over all data samples. The rightmost column shows the average correlation over the 3 datasets.

As expected, *Oracle Softmax* has the highest correlation out of all the methods, with average correlations around 0.35 across all datasets and models, indicating moderate correlation (Hopkins, 2000).

In most cases, all uncertainty estimation techniques improve over *Baseline Softmax*. The *Deep Ensemble* technique achieves the highest mean correlation coefficients ranging between 0.204 and 0.226 for GoEmotions and RottenTomatoes, across the three models. *Monte Carlo Dropout* also shows substantial improvement, with correlations ranging between 0.126 and 0.229 for GoEmotions and RottenTomatoes across models.

On the GAB Hate Corpus, especially in combination with XLNet the results do not align with the patterns observed in the other datasets and models. For this dataset, we even see lower correlations than the baseline, when using *Monte Carlo Dropout* and *Label Smoothing*.

E Class-Level Analysis - Heatmaps

Figure 6 shows the heatmaps comparing the disagreements in the model (baseline BERT) and in human annotations.

Model	Method	GoEmotions Corr.	GoEmotions % Improv.	Rotten Tomatoes Corr.	Rotten Tomatoes % Improv.	GAB Hate Speech Corr.	GAB Hate Speech % Improv.	Average Corr.
BERT	Baseli.	0.081 ± 0.002	-	0.101 ± 0.009	-	0.024 ± 0.042	-	0.069
	DE	0.204 ± 0.008	152%	0.207 ± 0.004	105%	0.078 ± 0.016	225%	0.163
	MCD	0.196 ± 0.003	142%	0.126 ± 0.030	25%	0.087 ± 0.031	262%	0.136
	LS	0.141 ± 0.011	74%	0.123 ± 0.013	22%	0.070 ± 0.038	192%	0.111
	Oracle	**0.372 ± 0.013**	**359%**	**0.264 ± 0.012**	**161%**	**0.399 ± 0.014**	**1562%**	**0.345**
RoBERTa	Baseli.	0.075 ± 0.007	-	0.083 ± 0.009	-	0.031 ± 0.037	-	0.063
	DE	0.224 ± 0.005	199%	0.224 ± 0.013	170%	0.076 ± 0.009	145%	0.175
	MCD	0.229 ± 0.006	205%	0.191 ± 0.024	130%	0.112 ± 0.039	261%	0.177
	LS	0.169 ± 0.019	125%	0.131 ± 0.009	58%	0.056 ± 0.041	81%	0.119
	Oracle	**0.383 ± 0.008**	**411%**	**0.303 ± 0.030**	**265%**	**0.379 ± 0.010**	**1123%**	**0.355**
XLNet	Baseli.	0.095 ± 0.011	-	0.059 ± 0.028	-	0.054 ± 0.049	-	0.069
	DE	0.226 ± 0.008	138%	0.204 ± 0.010	246%	0.065 ± 0.013	20%	0.165
	MCD	0.223 ± 0.001	135%	0.183 ± 0.005	210%	0.052 ± 0.029	-4%	0.153
	LS	0.155 ± 0.007	63%	0.150 ± 0.009	154%	0.012 ± 0.020	-78%	0.106
	Oracle	**0.371 ± 0.006**	**291%**	**0.302 ± 0.019**	**412%**	**0.346 ± 0.069**	**541%**	**0.340**

Table 8: Correlation coefficients (mean ± std.) and percentage improvement over *Baseline* for each model. In each column, per model, the best scores are **bolded**, and the second-best are underlined.

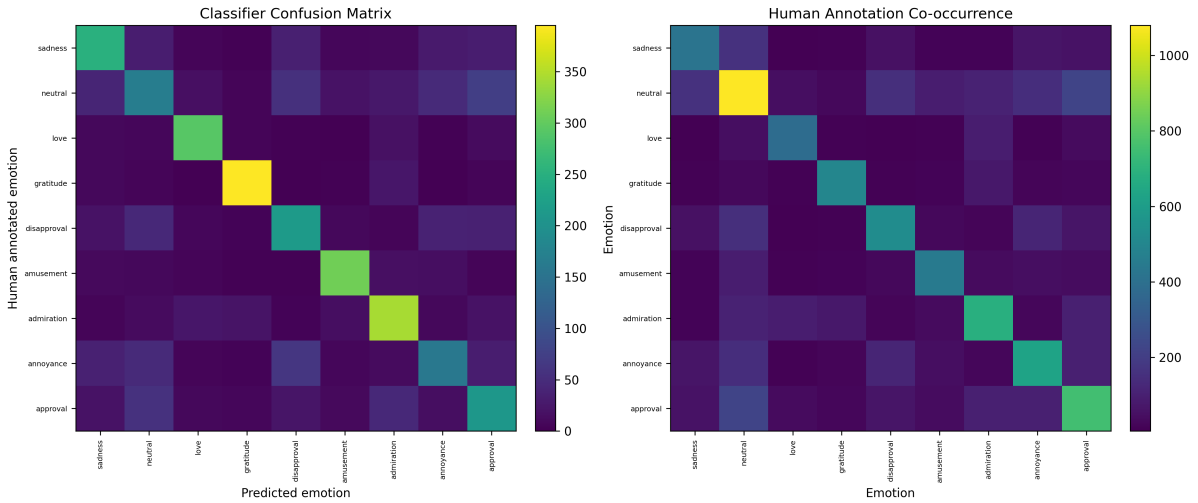


Figure 6: Heat-maps of model confusions (left) and human co-occurrences (right) on GoEmotions.