Confidence Calibration for Intent Detection via Hyperspherical Space and Rebalanced Accuracy-Uncertainty Loss

Yantao Gong[1,2,3], **Cao Liu**[1], **Fan Yang**[1], **Xunliang Cai**[1],
Guanglu Wan[1], **Jiansong Chen**[1], **Weipeng Zhang**[1], **Houfeng Wang**[3,†]

[1] Meituan
[2] School of Software and Microelectronics, Peking University
[3] MOE Key Lab of Computational Linguistics, Peking University

{gongyt, wanghf}@pku.edu.cn {liucao, yangfan79, caixunliang, wanguanglu, chenjiansong, zhangweipeng02}@meituan.com

Abstract

Data-driven methods have achieved notable performance on intent detection, which is a task to comprehend user queries. Nonetheless, they are controversial for over-confident predictions. In some scenarios, users do not only care about the accuracy but also the confidence of model. Unfortunately, mainstream neural networks are poorly calibrated, with a large gap between accuracy and confidence. To handle this problem defined as confidence calibration, we propose a model using the hyperspherical space and rebalanced accuracy-uncertainty loss. Specifically, we project the label vector onto hyperspherical space uniformly to generate a dense label representation matrix, which mitigates over-confident predictions due to overfitting sparce one-hot label matrix. Besides, we rebalance samples of different accuracy and uncertainty to better guide model training. Experiments on the open datasets verify that our model outperforms the existing calibration methods and achieves a significant improvement on the calibration metric.

Introduction

Intent detection is a crucial portion in comprehending user queries, which generally predicts intent tags by semantic classification (Brenes, Gayo-Avello, and Pérez-González 2009; Qin et al. 2020). Therefore, it is widely used in many NLP applications, such as search, task-based dialogue, and other fields (Zhang and Wang 2016; Larson et al. 2019; Casanueva et al. 2020).

In recent years, data-driven methods develop rapidly and become a primary trend of intent detection. However, they are highly criticized for over-confident predictions (Niculescu-Mizil and Caruana 2005; Nguyen, Yosinski, and Clune 2015; Pereyra et al. 2017; Li, Dasarathy, and Berisha 2020). As shown in Figure 1(a), there is a serious phenomenon that the prediction confidence (i.e. probability associated with the predicted label) of samples is very high even if the samples are misclassified. For example, when the confidence is in [0.9-1], the proportion of misclassified samples reaches 35.67%. Besides, the average confidence (90.39%) is evidently over the accuracy (56.17%).

One of the effective solutions to deal with the aforementioned problem is confidence calibration. A perfectly cali-

†Corresponding author.
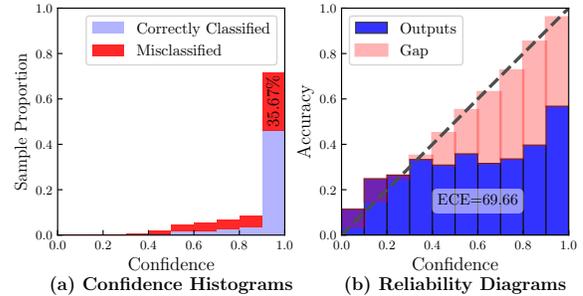
Figure 1: The confidence histograms and reliability diagrams of TNEWS dataset. We employ the fine-tuned BERT model to perform statistical analysis. As the figure demonstrates, when the confidence is between 0.9 and 1, misclassified samples constitute about 35.67%. "Gap" represents the difference between confidence and accuracy. Model is worse calibrated if the "Gap" is larger. Fine-tuned BERT model without calibration tends to make over-confident predictions and possesses a high *expected calibration error* (ECE).

brated model is supposed to output average confidence equal to the accuracy (Kong et al. 2020; Küppers et al. 2020). Unfortunately, due to over-parameterization and overfitting of the conventional methods, mainstream neural networks are poorly calibrated (Krishnan, Tickoo, and Tickoo 2020; Wang et al. 2020b; Schwaiger et al. 2021; Enomoto and Eda 2021). As demonstrated in Figure 1, "Gap" means the discrepancy between the average confidence and accuracy. The larger the "Gap" as, the worse the model is calibrated. Model without calibration, indicated in Figure 1(b), easily faces under-estimation problem when confidence is less than 0.2 and over-estimation problem when confidence is more than 0.4. Therefore, it owns a higher *expected calibration error* (ECE, calibration metric, more details in Experiments Section) than perfectly calibrated model.

To handle the confidence calibration problem, researchers have proposed numerous works (Nguyen and O'Connor 2015; Szegedy et al. 2016a; Müller, Kornblith, and Hinton 2019). One primary calibration approach acts on the post-processing stage. Guo et al. (2017) provide temperature scaling, which learns a single parameter from the development dataset to rescale all the logit before transmitting to softmax.

label$_i$ [0, 1, ..., 0]$_{1 \times K}$ label$_i$ [-0.56, ..., 0.88, 0.20]$_{1 \times H}$

(a) One-hot Label   (b) Uniform Label
in Euclidean Space   in Hyperspherical Space
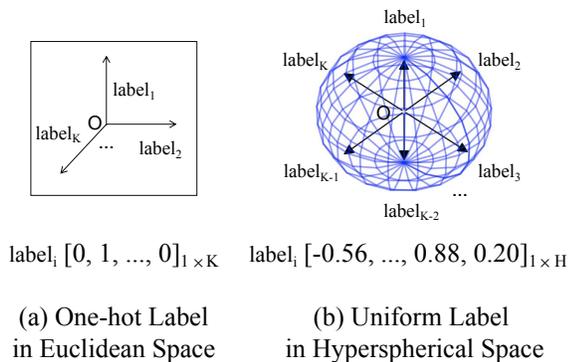
Figure 2: Representation of label vectors in euclidean space and hyperspherical space. One-hot label vectors are in the form of a sparse matrix and only use the positive portion. Additionally, one-hot label vectors require at least as many dimensions as the size of label set like K ($K \ll H$ in most cases). In contrast, label vectors in hyperspherical space are shaped into a dense matrix and employ the entire portion without dimension limitation.

Another way to calibrate the model is by designing a particular loss function to minimize the discrepancy between accuracy and confidence. Krishnan, Tickoo, and Tickoo (2020) lately propose the accuracy versus uncertainty calibration loss (AVUC loss), which leverages the relationship between accuracy and uncertainty as an anchor for calibration, and it obtains a significant improvement.

Nevertheless, the aforementioned methods have some important issues. 1) As demonstrated in Figure 2, one of the problems lies in that the above methods project the labels in the form of a one-hot matrix in Euclidean space, which is sparse and merely uses the positive portion of the output space. During the training process, such a sparse matrix is easy to bring about the network to make over-confident predictions, as proved by Szegedy et al. (2016b) and Müller, Kornblith, and Hinton (2019). 2) Another issue is that although Krishnan, Tickoo, and Tickoo (2020) divide samples into several groups according to their accuracy and uncertainty, it treats accurate and inaccurate samples equally. In fact, there exists a large number of misclassified samples with high confidence (low uncertainty), displayed in Figure 1(a), which suggests that the model is misleading by the wrong signal during training.

In order to deal with the above issues, we propose a model employing the **H**yperspherical **S**pace and **R**ebalanced **A**ccuracy-**U**ncertainty loss (HS-RAU) to process confidence calibration for the intent detection task. Specifically, 1) We project the label vector onto the hyperspherical space uniformly, as vividly shown in Figure 2. Hyperspherical space uses a dense matrix to represent labels and employs the entire portion of the output space rather than one-hot labels. In this way, we mitigate the overfitting problem of model to the sparse one-hot matrix. 2) We propose a rebalanced accuracy-uncertainty loss to capitalize on the properties of distinct samples. Through RAU loss, we optimize the accu-

rate samples with high uncertainty and the inaccurate samples with low uncertainty respectively, which contributes to better guide model training.

To validate the effectiveness of our model, we conduct abundant experiments on the three open datasets. Empirical results demonstrate that our model achieves evident improvements compared with the SOTA. Specifically, F1 increases on all the datasets with the calibration metric (ECE) drops down 10.50% on average. On the TNEWS dataset, the ECE achieves an obvious amelioration of 29.67% and the F1 obtains 1.21% promotion. Furthermore, our model acquires better performance among the existing methods on noisy data and low-frequency labels.

To sum up, our contributions are as follows:

(1) We uniformly project the label vectors onto the hyperspherical space to obtain a denser representation matrix, which mitigates the model to overfit the sparce one-hot label matrix and generate over-confident predictions.

(2) We rebalance the accuracy and uncertainty of samples and optimize the accurate samples with low uncertainty and inaccurate samples with high uncertainty separately by RAU loss to provide better guidance in the training process.

(3) The experimental results demonstrate that our model gains an advantage over the SOTA, not only in the F1 but also in the confidence calibration metric. Moreover, we obtain noteworthy performance on noisy data and low-frequency labels.

Related Work

Intent Detection. Intent is the sematic purpose of a query, which is generated by users (Xu and Sarikaya 2013; Wang, Tang, and He 2018). As a matter of fact, the essence of intent detection is text classification (Brenes, Gayo-Avello, and Pérez-González 2009; Mehri, Eric, and Hakkani-Tur 2020; Chatterjee and Sengupta 2020). After training on the dataset with ground-truth labels, the model attempts to predict the intent of query within the existing intent set. There have been plenty of researches on conventional neural network methods in the last few decades (Xu and Sarikaya 2013; Liu and Lane 2016; Zhang et al. 2019; Haihong et al. 2019; Wang et al. 2020a; Gerz et al. 2021). During recent years, with the rapid development of computing power, pre-trained models such as BERT (Devlin et al. 2018) are employed for intent detection frequently (Castellucci et al. 2019; He et al. 2019; Zhang, Zhang, and Chen 2019; Athiwaratkun et al. 2020; Gong et al. 2021).

Confidence Calibration. Confidence calibration has a long history of research in statistical machine learning (Brier 1950; Griffin and Tversky 1992; Gneiting and Raftery 2007). In the past several years, one major calibration methods fall into the post-processing stage (Platt 1999; Zadrozny and Elkan 2001; Kumar, Liang, and Ma 2019; Zhang, Kailkhura, and Han 2020; Rahimi et al. 2020). For example, Guo et al. (2017) propose the temperature scaling. The trained model learns a single calibration scale from the development set. Another main calibration approaches try to optimize a function that represents the difference of average confidence and accuracy (Kumar, Sarawagi, and Jain 2018;
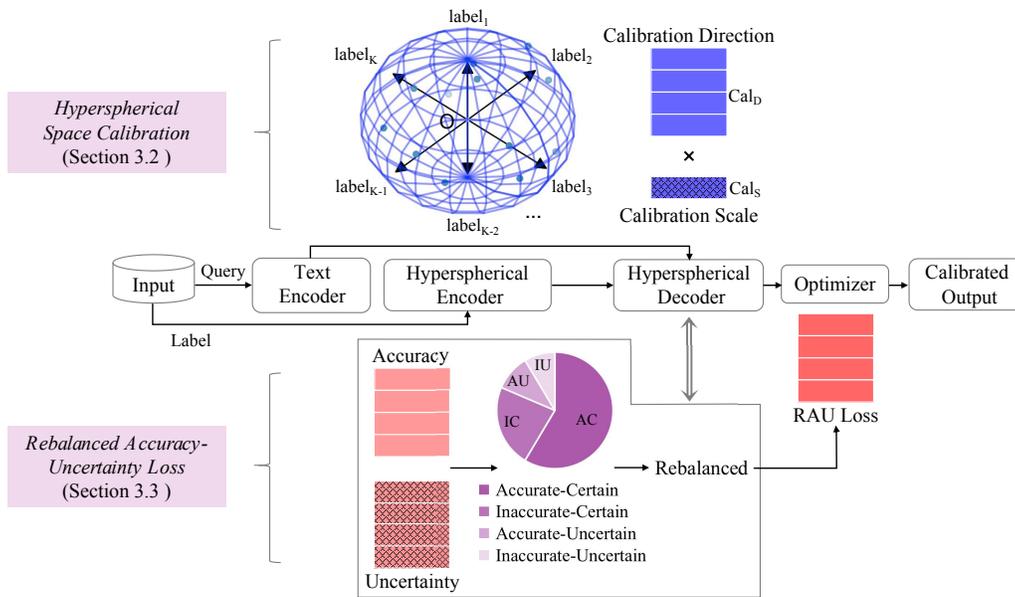
Figure 3: The illustration of confidence calibration via Hyperspherical Space and Rebalanced Accuracy-Uncertainty loss (HS-RAU) framework. After getting the encoded vector of input query by the text encoder, we project the label vector onto the hyperspherical space uniformly, and encode the input label by the hyperspherical encoder to obtain a dense label matrix. Then, we compute the calibration direction matrix as well as the calibration scale. Next, we partition the samples into four sets according to their accuracy and uncertainty, rebalance the samples' accuracy and uncertainty by the RAU loss. Through the above process, we acquire the output with calibration.

Kull et al. 2019; Mukhoti et al. 2020; Gupta et al. 2020). For instance, Krishnan, Tickoo, and Tickoo (2020) devise a loss function to combine accuracy and uncertainty. Jung et al. (2020) come up with a method to minimize the distribution between predicted probability and empirical probability.

Method

Model Overview

As shown in Figure 3, we employ the *hyperspherical space and rebalanced accuracy-uncertainty loss* to process confidence calibration. First, we use a text encoder such as BERT to acquire the encoded vector of the input query. Next, through the hyperspherical encoder, we obtain the dense encoded matrix of the input labels, which alleviates the overconfident predictions caused by the overfitting of sparse one-hot label matrix. After that, we utilize the hyperspherical decoder to calculate the calibration direction matrix and calibration scale. Furthermore, we separate the samples according to their accuracy and uncertainty, and design the rebalanced accuracy-uncertainty loss to optimize accurate and inaccurate samples respectively. In the end, we obtain the output with calibration.

Hyperspherical Space Calibration

In this submodule, we introduce how to separate the hyperspherical space homogeneously and project label vectors onto the hyperspherical space to obtain a dense label matrix.

Text Encoder For N queries $\{Q_1, ..., Q_i, ..., Q_N\}$, the corresponding labels are $\{T_1, ..., T_i, ..., T_N\}$, where $T_i \in C$. $C = \{1, ..., K\}$ indicates the set of K label tags. We exploit the text encoder like BERT to extract the encoded vector E_i (H dimension) such as [CLS] of each input query Q_i. The encoded vector matrix E of all the queries is calibrated in the hyperspherical decoder.

Hyperspherical Encoder Before the learning process, we separate the H-dimensional output space $\mathbb{S}^H$ into K subspaces uniformly, which has the same size as the label set C. Then, we define the vector of the hyperspherical label as $\{h_1, ..., h_i, ..., h_K\}$, corresponding to the K subspaces. In addition, the norm of each vector satisfies $\|h_i\| = 1$. The dimension of hyperspherical label vector is H, which equals the dimension of encoded vector. The hyperspherical encoder encodes each input label to a dense hyperspherical label vector, which is utilized in the hyperspherical decoder for calibration.

Here comes the detail of uniformly projecting the label vectors onto hyperspherical space. For each label vector h_i in the hyperspherical space, it has $K - 1$ cosine distances between all the $K - 1$ label vectors except itself, and the max cosine distance among them is D_i, defined as below:

$$D_i = \max(d_{ij}) \tag{1}$$

where $i, j \in C$ and $i \neq j$. d_{ij} is the cosine distance between label vector h_i and h_j. As our goal is to make the label vector uniformly distributed in the hyperspherical space, therefore, it is equivalent to the optimization problem that minimizes

the sum of the maximum cosine distance D_i of each label vector, as the following modality:

$$\mathcal{L}_h = \min \frac{1}{K} \sum_{i=1}^{K} D_i \qquad (2)$$

Furthermore, due to all the label vectors are unit vectors, the above formula can be converted to matrix multiplication, which speeds up the calculation, by the following equations:

$$\mathcal{L}_h = \min \frac{1}{K} \sum_{i=1}^{K} \max(Z_i),$$
$$Z = X \cdot X^T - 2I \qquad (3)$$

where $X = [h_1, ..., h_i, ..., h_K]$ is the matrix of hyperspherical label vector. I is the identity matrix. Z_i is the ith row of Z. In order to avoid self-selection, Z subtracts identity matrix I twice.

Hyperspherical Decoder After acquiring the encoded query vector and the dense encoded hyperspherical label vector through hyperspherical encoder, we utilize the hyperspherical decoder to get the calibration direction and the calibration scale.

We perform the dot product of the encoded vector with each hyperspherical label vector to get the calibration direction matrix $Cali_D$, formulated as below:

$$Cali_D = E \cdot X^T \qquad (4)$$

where $E \in \mathbb{R}^{N \times H}$ denotes the encoded vector matrix of all the queries. $X^T \in \mathbb{R}^{H \times K}$ is the transpose matrix of dense hyperspherical label vector. Then, we calculate the norm of label vector matrix as the calibration scale $Cali_S$, which is the scale parameter during the overall process, by using the following equation:

$$Cali_S = \|X\| \qquad (5)$$

Finally, we compute the calibrated new logit L as below:

$$L = Cali_S \times Cali_D \qquad (6)$$

where the calibration scale $Cali_S$ is an unidimensional variable and the calibration direction $Cali_D \in \mathbb{R}^{N \times K}$.

Rebalanced Accuracy-Uncertainty Loss

In this submodule, we design the rebalanced accuracy-uncertainty loss to optimize accurate and inaccurate samples separately. Whether a sample is considered as accurate depends on whether the predicted label of the sample T_i' equals to the exact sample's label T_i, so we define the confidence (probability of predicted label) of a single sample as a_i in the following:

$$a_i = \begin{cases} \max(p_i), & \text{if } T_i' = T_i \\ 1 - \max(p_i), & \text{otherwise.} \end{cases} \qquad (7)$$

where p_i is the predicted probability after transmitting to softmax. Therefore, when the predictions are accurate the a_i is close to 1, while it is close to 0 when inaccurate. As there is no ground truth evaluation of the uncertainty, we utilize

the calculation method described in Krishnan, Tickoo, and Tickoo (2020) to get the uncertainty u_i as follows:

$$u_i = -p_i \log p_i \qquad (8)$$

Then, we set the uncertainty threshold as $u_\theta \in [0, 1]$, which is a heuristic setting obtained through the average uncertainty of training samples from initial epochs. A sample is defined as certain when the uncertainty of it is lower than u_θ. Otherwise, it's defined as uncertain. Then, we divide the training samples into four sets $\{AC, AU, IC, IU\}$ separately, where AC means Accurate-Certain, AU means Accurate-Uncertain, IC means Inaccurate-Certain, and IU means Inaccurate-Uncertain.

Based on the assumption mentioned in Krishnan, Tickoo, and Tickoo (2020), a well-calibrated model provides a low uncertainty for accurate predictions while it provides a high uncertainty for inaccurate predictions. Therefore, the model with calibration is supposed to produce a higher $AVU \in [0, 1]$ measure. AVU is computed by summing the number of $\{AC, IU\}$ two sets, and then divide the total number of $\{AC, AU, IC, IU\}$ four sets.

To make the AVU function differentiable for neural network parameters, we devise the calculation methods like:

$$n_{AC} = \sum_{i \in \{T_{i'}=T_i \text{ and } u_i \leq u_\theta\}} a_i \odot (1 - \tan(u_i)),$$
$$n_{AU} = \sum_{i \in \{T_{i'}=T_i \text{ and } u_i > u_\theta\}} a_i \odot \tan(u_i),$$
$$n_{IC} = \sum_{i \in \{T_{i'} \neq T_i \text{ and } u_i \leq u_\theta\}} a_i \odot (1 - \tan(u_i)),$$
$$n_{IU} = \sum_{i \in \{T_{i'} \neq T_i \text{ and } u_i > u_\theta\}} a_i \odot \tan(u_i) \qquad (9)$$

where $\odot$ is hadamard project. In addition, we step further on and rebalance the accuracy-uncertainty, which prompts the model to respectively optimize accurate samples with low uncertainty and inaccurate samples with high uncertainty during training. To be specific, we define the RAU loss as:

$$\mathcal{L}_{RAU} = \log\left(1 + \frac{n_{AU}}{n_{AC} + n_{AU}} + \frac{n_{IC}}{n_{IC} + n_{IU}}\right) \qquad (10)$$

When n_{AU} and n_{IC} are optimized close to zero, the RAU loss is close to zero, which means the model is certain about the predictions of accurate samples, while there are no overconfident predictions of the inaccurate samples.

Experiments

Experimental Setup

Experimental Datasets We mainly experiment on three open datasets described below. The download links are displayed in Appendix A.

TNEWS, a Chinese dataset proposed by Xu et al. (2020), has identical essence with intent detection. It includes 53360 samples in 15 categories. The provided test set are without gold labels. So we regard validation set as test set and randomly divide 5000 samples from training set for validation.

HWU64, proposed by Liu et al. (2019) to reflects human-home robot interaction, which owns 15726 samples spanning 64 intents. We use one fold train-test split with 9960 training samples and 1076 testing samples.

BANKING77, proposed by Casanueva et al. (2020), which has 13083 samples, 9002 for training and 3080 for testing. This dataset consists of 77 intents in a single domain of online banking inquiry.

Model	TNEWS		HWU64		BANKING77		Average	
	F1	ECE	F1	ECE	F1	ECE	F1	ECE
BERT (Devlin et al. 2018)	54.81	69.66	91.85	17.18	93.61	11.98	80.09	32.94
TS (Guo et al. 2017)	54.81	49.88	91.85	15.86	93.61	11.87	80.09	25.87
LS (Müller, Kornblith, and Hinton 2019)	55.29	53.99	92.06	16.51	93.86	11.40	80.40	27.30
PosCal (Jung et al. 2020)	54.98	68.05	92.03	16.14	93.66	11.93	80.30	32.05
AVUC (Krishnan, Tickoo, and Tickoo 2020)	55.41	67.98	92.02	**15.71**	93.83	11.79	80.42	31.83
HS-RAU (Ours)	**56.02**	**39.99**	**92.52**	16.12	**93.89**	**11.21**	**80.81**	**22.44**

Table 1: Overall comparison with different calibration methods on three open datasets.

Comparison Methods We compare with the methods as listed below:

BERT (Devlin et al. 2018): represents the pre-trained base BERT model.

Temperature Scaling (TS) (Guo et al. 2017): is the classical post-processing method learning a single parameter from the dev dataset to rescale the logit after the model is trained.

Label Smoothing (LS) (Müller, Kornblith, and Hinton 2019): smoothes some part of the one-hot label' probability to a weighted mixture probability of the none ground-truth labels, which is set to compare our hyperspherical labels.

Posterior Calibrated (PosCal) (Jung et al. 2020): minimizes the difference between the predicted and empirical posterior probabilities, which is a competitive recent research.

Accuracy Versus Uncertainty Calibration (AVUC) (Krishnan, Tickoo, and Tickoo 2020): proposes an optimization method that utilizes the relevance of accuracy and uncertainty as an anchor for calibration.

Implementation Details All experiments are taken on BERT (with or without confidence calibration) unless otherwise specified. We employ Adam (Kingma and Ba 2015) as the optimizer and search learning rate in {4e-5, 5e-5} with the training epochs in {19, 23} and about 40s per epoch. To make full use of the GPU memory, we set the batch size to 256. The type of GPU is Tesla V100. Besides, the KL loss between predicted probability and empirical probability is added optionally in PosCal, AVUC, and our model. More implementation details are shown in Appendix A.

Confidence Calibration Metric We follow the previous researches and utilize the *expected calibration error* (ECE) (Naeini, Cooper, and Hauskrecht 2015), which is a common evaluation metric to calculate the calibration error in confidence calibration. ECE separates the predictions of samples into M bins according to the predicted probability called confidence. Then, accumulating the weighted differences between accuracy and confidence in each bin:

$$ECE = \frac{1}{K} \sum_{i=1}^{K} \sum_{j=1}^{M} \frac{|B_{ij}|}{N} |Acc_{ij} - Con_{ij}| \qquad (11)$$

where $|B_{ij}|$ is the size of bin j in label i, N is the number of total prediction samples, Acc_{ij} is the empirical probability and Con_{ij} is the average predicted probability for label i in bin j respectively.

Comparison with State-of-the-arts

Comparison Settings. We reproduce the baselines, and the results are almost equal to the published metrics. Based on that, we conduct extensive experiments on the above three datasets to validate the effectiveness of our model. F1 and ECE are considered as the main evaluation metrics.

Comparison Results. As Table 1 illustrated:

(1) Regardless of which dataset we choose, our model achieves the best F1 performance among all the calibration methods. Specially, we obtain a significant reduction in ECE, which drops down 10.50% on average compared with the baseline model. This proves the effectiveness of our model to project label vector onto hyperspherical space uniformly and utilize the rebalanced accuracy-uncertainty loss in confidence calibration.

(2) All the calibration methods have limited amelioration in the dataset possessing better performance. It makes sense for the reason that the model is well studied on these datasets. Hence the main distribution of its confidence is as high as the accuracy, like 90+%, which results in more credible predictions. So we majorly analyze the TNEWS dataset in the subsequent experiments.

(3) In the case of TNEWS dataset, the F1 gains 1.21% over BERT while the ECE decreases remarkably. Furthermore, more information on the rest datasets can be inquired in Appendix B.

Observing Miscalibration

Comparison Settings. We use the reliability diagrams for observation, which is a visual representation of model calibration (Niculescu-Mizil and Caruana 2005). The average confidence within each bin is defined as "Outputs", while the absolute difference between confidence and accuracy in each bin is defined as "Gap". The ECE is proportional to "Gap" in some degree, described in Sec. .

Comparison Results. As Figure 4 depicts, although distinct calibration methods still have a miscalibration phenomenon on TNEWS dataset, BERT with calibration can acquire a lower ECE. Especially, our model decreases ECE prominently and less inclined to make overconfident predictions for samples with higher accuracy compared with the AVUC, which manifests the validity of our RAU loss that rebalances accuracy-uncertainty and optimizes accurate as well as inaccurate samples respectively.
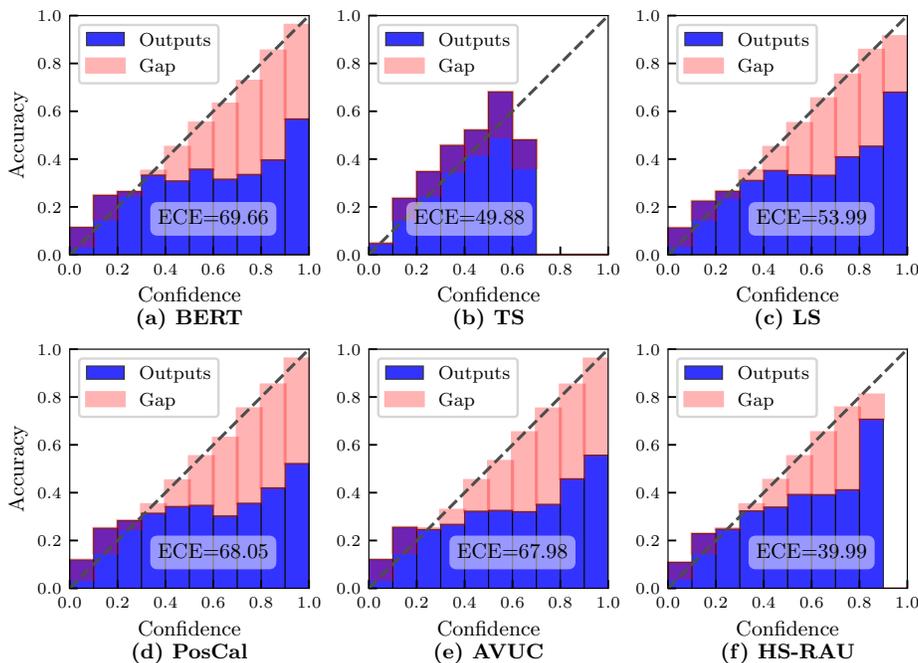
Figure 4: Reliability diagrams for TNEWS dataset, before calibration (a) and after calibration (b)-(f).

Model	ACC	P	R	F1	ECE
HS-RAU	**56.39**	**56.31**	**55.82**	**56.02**	**39.99**
w/o HS	56.32	55.82	55.26	55.52	67.97
w/o RAU	55.90	56.05	54.75	55.28	42.53
w/o Both	56.37	56.13	54.81	55.31	68.50
RAU⇒AVUC	56.21	55.72	55.45	55.55	41.82

Table 2: Ablation study on TNEWS dataset by removing the main components, where "w/o" means without, "HS" represents hyperspherical space calibration, and "RAU" indicates rebalanced accuracy-uncertainty loss.

Ablation Study

Comparison Settings. To validate the effectiveness of our model components, we gradually get rid of some components, including hyperspherical space and rebalanced accuracy-uncertainty loss. In practice, if hyperspherical space is not employed in the model, we use the typical one-hot vector in Euclidean space to represent the labels.

Comparison Results. As described in Table 2:

(1) Taking out any components of our model results in performance reduction, which certifies the validity of all components.

(2) Specifically, replacing hyperspherical space leads to conspicuous performance degradation. It shows that projecting labels onto hyperspherical space to obtain a dense label matrix and setting the calibration scale throughout the process can effectively draw down the ECE.

(3) Substituting RAU for AVUC causes a minor decline

in F1, while ECE gets 1.83% worse. This proves that RAU, which respectively optimizing accurate and inaccurate samples, is beneficial to improve performance. See more results on other datasets in Appendix C.

Discussion

Effectiveness on Noisy Data

Comparison Settings. Model performance under noisy data is an important indicator to measure robustness, as it's a frequent phenomenon for data to have noise. We randomly mislabel 5%, 10%, 30%, and 50% part of samples' labels on TNEWS dataset to simulate the noisy environment.

Comparison Results. The experimental results of Table 3 support the statements as below:

(1) The experimental results indicate that our model still obtains significant improvement irrespective of how much noise label is in the dataset.

(2) Take TNEWS dataset with 30% error labels as an example, F1 increases 3.27%. In the meantime, ECE decreases by 45.73%. During the experiment, we find that although temperature scaling obtains a comparable ECE, the scale parameter turns extremely large, and the output probability is only distributed into one bin. The reason may be that temperature scaling is not suitable for the situation where training set and other sets are labeled differently, as it learns the scale parameter from the development set after the model is trained on the training set.

(3) Above results verify the effectiveness of our model on the noisy data, which projects the label vector uniformly onto the hyperspherical space and makes better use of the dense label representations. Hyperspherical label vector is

Model	5% Noisy Labels		10% Noisy Labels		30% Noisy Labels		50% Noisy Labels	
	F1	ECE	F1	ECE	F1	ECE	F1	ECE
BERT	53.46	72.19	51.44	72.38	43.76	77.98	32.22	83.35
LS	53.95	46.46	52.59	43.81	44.83	46.92	32.91	65.94
PosCal	54.11	52.77	52.14	61.92	45.40	56.83	33.93	71.37
AVUC	54.04	64.47	51.94	61.78	45.74	56.44	33.74	71.14
HS-RAU	**54.43**	**34.37**	**53.66**	**35.48**	**47.03**	**32.25**	**35.20**	**41.06**

Table 3: Performance on TNEWS dataset with noise.

Model	L-F_1	L-F_2	L-F_3	Average	
	F1	F1	F1	F1	ECE
BERT	40.54	44.50	57.54	47.53	1.61
TS	40.54	44.50	57.54	47.53	3.05
LS	40.45	46.56	59.50	48.84	1.14
PosCal	35.44	46.15	60.00	47.20	1.50
AVUC	43.37	**48.31**	**60.88**	50.85	1.46
HS-RAU	**54.12**	47.22	59.06	**53.47**	**1.08**

Table 4: Performance of the low-frequency labels on TNEWS dataset, where L-F_1 means the Lowest Frequency label and "Average" means the average performance of the three lowest frequency labels.

Model	ACC	P	R	F1	ECE
Albert-tiny	52.00	48.67	48.41	48.49	28.00
+ HS-RAU	**53.03**	**50.24**	**49.18**	**49.58**	**23.91**
XLNet	56.84	**56.10**	55.72	55.81	40.56
+ HS-RAU	**57.06**	**56.10**	**56.14**	**56.04**	**35.79**
BERT-large	57.84	**57.59**	56.45	56.91	69.26
+ HS-RAU	**58.09**	57.33	**56.65**	**56.93**	**39.02**

Table 5: Performance of distinct encoders on TNEWS.

not utterly orthogonal like the one-hot label, so mislabeled samples are more likely to be calibrated. More information on other metrics can be found in Appendix D.

Effectiveness on Low-Frequency Labels

Comparison Settings. Class-imbalanced datasets commonly face the long tail problem. To examine the performance of minority labels, we experiment on the three lowest frequency labels on TNEWS dataset, which contains fifteen classes in total. The sum of fifteen labels' ECE equals the gross ECE. Besides, L-F_1 means the Lowest Frequency label, L-F_2 means the second lowest, and so on. "Average" means the average performance of the three lowest frequency labels.

Comparison Results. As demonstrated in Table 4, our model reaches the highest average F1 of the low-frequency labels, with a 5.94% absolute increase. Apart from that, it works better on the average ECE too. According to the consequences, we can infer that it's no picnic to learn sparse label features (like one-hot) with low-frequency samples. In contrast, the label features of our model are dense, as we separate the label vector into the hyperspherical space horizontally and use more portion of the output space. More details on the comparison results are shown in Appendix E.

Effectiveness on Different Encoders

Comparison Settings. Different encoders have distinct output spaces. To assess the performance of different encoders,

we also horizontally compare with other encoders such as Albert-tiny (Lan et al. 2020), XLNet (Yang et al. 2019), and BERT-large (Devlin et al. 2018) on TNEWS dataset.

Comparison Results. The consequences of different encoders on TNEWS dataset are shown in Table 5. Though different encoders behave diversely from each other, they all acquire a comparable enhancement with the help of our confidence calibration strategy. Results indicate that projecting the output space onto hyperspherical space by our strategy possesses a certain universality, which is not limited to the BERT model.

Conclusion

In this work, we propose a confidence calibration model for intent detection via **H**yperspherical **S**pace and **R**ebalanced **A**ccuracy-**U**ncertainty loss (HS-RAU). With the help of projecting label vectors onto hyperspherical space uniformly, we make better use of the dense label representation matrix to mitigate the over-confident predictions as well as the whole portion of output space. Through the rebalanced accuracy-uncertainty loss, we better guide the model to respectively optimize the accurate and inaccurate samples. Experimental results indicate that our model obtains a decent rise over SOTA. Especially, we achieve a significant improvement in the confidence calibration metric (ECE) among the calibration methods.

Acknowledgments

References

Athiwaratkun, B.; Santos, C. D.; Krone, J.; and Xiang, B. 2020. Augmented Natural Language for Generative Sequence Labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Brenes, D. J.; Gayo-Avello, D.; and Pérez-González, K. 2009. Survey and evaluation of query intent detection methods. In *Proceedings of the 2009 Workshop on Web Search Click Data*, 1–7.

Brier, G. W. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1): 1–3.

Casanueva, I.; Temčinas, T.; Gerz, D.; Henderson, M.; and Vulić, I. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Castellucci, G.; Bellomaria, V.; Favalli, A.; and Romagnoli, R. 2019. Multi-lingual Intent Detection and Slot Filling in a Joint BERT-based Model. *ArXiv*, abs/1907.02884.

Chatterjee, A.; and Sengupta, S. 2020. Intent Mining from past conversations for Conversational Agent. In *COLING*.

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

Enomoto, S.; and Eda, T. 2021. Learning to Cascade: Confidence Calibration for Improving the Accuracy and Computational Cost of Cascade Inference Systems. In *AAAI*.

Gerz, D.; hao Su, P.; Kusztos, R.; Mondal, A.; Lis, M.; Singhal, E.; Mrksic, N.; Wen, T.-H.; and Vuli'c, I. 2021. Multilingual and Cross-Lingual Intent Detection from Spoken Data. *ArXiv*, abs/2104.08524.

Gneiting, T.; and Raftery, A. 2007. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102: 359 – 378.

Gong, Y.; Liu, C.; Yuan, J.; Yang, F.; Cai, X.; Wan, G.; Chen, J.; Niu, R.; and Wang, H. 2021. Density-Based Dynamic Curriculum Learning for Intent Detection. *ArXiv*, abs/2108.10674.

Griffin, D.; and Tversky, A. 1992. The weighing of evidence and the determinants of confidence. *Cognitive Psychology*, 24: 411–435.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, 1321–1330. PMLR.

Gupta, K.; Rahimi, A. M.; Ajanthan, T.; Mensink, T.; Sminchisescu, C.; and Hartley, R. 2020. Calibration of Neural Networks using Splines. *ArXiv*, abs/2006.12800.

Haihong, E.; Niu, P.; Chen, Z.; and Song, M. 2019. A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling. *ArXiv*, abs/1907.00390.

He, C.; Chen, S.; Huang, S.; Zhang, J.; and Song, X. 2019. Using Convolutional Neural Network with BERT for Intent Determination. *2019 International Conference on Asian Language Processing (IALP)*, 65–70.

Jung, T.; Kang, D.; Cheng, H.; Mentch, L.; and Schaaf, T. 2020. Posterior Calibrated Training on Sentence Classification Tasks. *ArXiv*, abs/2004.14500.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Kong, L.; Jiang, H.; Zhuang, Y.; Lyu, J.; Zhao, T.; and Zhang, C. 2020. Calibrated Language Model Fine-Tuning for In- and Out-of-Distribution Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1326–1340. Online: Association for Computational Linguistics.

Krishnan, R.; Tickoo, O.; and Tickoo, O. 2020. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33.

Kull, M.; Perelló-Nieto, M.; Kängsepp, M.; de Menezes e Silva Filho, T.; Song, H.; and Flach, P. A. 2019. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration. In *Conference and Workshop on Neural Information Processing Systems*.

Kumar, A.; Liang, P.; and Ma, T. 2019. Verified Uncertainty Calibration. In *Conference and Workshop on Neural Information Processing Systems*.

Kumar, A.; Sarawagi, S.; and Jain, U. 2018. Trainable Calibration Measures For Neural Networks From Kernel Mean Embeddings. In *International Conference on Machine Learning*.

Küppers, F.; Kronenberger, J.; Shantia, A.; and Haselhoff, A. 2020. Multivariate Confidence Calibration for Object Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1322–1330.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *ArXiv*, abs/1909.11942.

Larson, S.; Mahendran, A.; Peper, J. J.; Clarke, C.; Lee, A.; Hill, P.; Kummerfeld, J. K.; Leach, K.; Laurenzano, M. A.; Tang, L.; et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Li, W.; Dasarathy, G.; and Berisha, V. 2020. Regularization via Structural Label Smoothing. In *AISTATS*.

Liu, B.; and Lane, I. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *Conference of the International Speech Communication Association*.

Liu, X.; Eshghi, A.; Swietojanski, P.; and Rieser, V. 2019. Benchmarking Natural Language Understanding Services for building Conversational Agents. In *International Workshop on Spoken Dialog System Technology*.

Mehri, S.; Eric, M.; and Hakkani-Tur, D. 2020. Example-Driven Intent Prediction with Observers. *ArXiv*, abs/2010.08684.

Mukhoti, J.; Kulharia, V.; Sanyal, A.; Golodetz, S.; Torr, P.; and Dokania, P. 2020. Calibrating Deep Neural Networks using Focal Loss. *ArXiv*, abs/2002.09437.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When Does Label Smoothing Help? In *Conference and Workshop on Neural Information Processing Systems*.

Naeini, M.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining Well Calibrated Probabilities Using Bayesian Binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015: 2901–2907.

Nguyen, A. M.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 427–436.

Nguyen, K.; and O'Connor, B. T. 2015. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. *Proceedings of the 22nd international conference on Machine learning*.

Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, L.; and Hinton, G. E. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. *ArXiv*, abs/1701.06548.

Platt, J. 1999. Probabilistic Outputs for Support vector Machines and Comparisons to Regularized Likelihood Methods.

Qin, L.; Xu, X.; Che, W.; and Liu, T. 2020. AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling. *arXiv: Computation and Language*.

Rahimi, A.; Shaban, A.; Cheng, C.-A.; Hartley, R.; and Boots, B. 2020. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 33.

Schwaiger, F.; Henne, M.; Küppers, F.; Roza, F. S.; Roscher, K.; and Haselhoff, A. 2021. From Black-box to White-box: Examining Confidence Calibration under different Conditions. *ArXiv*, abs/2101.02971.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016a. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016b. Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826.

Wang, P.; Xu, J.; Liu, C.; Feng, H.; Li, Z.; and Ye, J. 2020a. Masked-field Pre-training for User Intent Prediction. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Wang, S.; Tu, Z.; Shi, S.; and Liu, Y. 2020b. On the Inference Calibration of Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3070–3079.

Wang, Y.; Tang, L.; and He, T. 2018. Attention-Based CNN-BLSTM Networks for Joint Intent Detection and Slot Filling. In *China National Conference on ComputationalLinguistics*.

Xu, L.; Zhang, X.; Li, L.; Hu, H.; Cao, C.; Liu, W.; Li, J.; Li, Y.; Sun, K.; Xu, Y.; et al. 2020. Clue: A chinese language understanding evaluation benchmark. *arXiv preprint arXiv:2004.05986*.

Xu, P.; and Sarikaya, R. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 78–83.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Conference and Workshop on Neural Information Processing Systems*.

Zadrozny, B.; and Elkan, C. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *International Conference on Machine Learning*.

Zhang, C.; Li, Y.; Du, N.; Fan, W.; and Yu, P. S. 2019. Joint Slot Filling and Intent Detection via Capsule Neural Networks. *ArXiv*, abs/1812.09471.

Zhang, J.; Kailkhura, B.; and Han, T. Y. 2020. Mix-n-Match: Ensemble and Compositional Methods for Uncertainty Calibration in Deep Learning. In *International Conference on Machine Learning*.

Zhang, X.; and Wang, H. 2016. A Joint Model of Intent Determination and Slot Filling for Spoken Language Understanding. In *International Joint Conference on Artificial Intelligence*.

Zhang, Z.; Zhang, Z.; and Chen, H. 2019. A Joint Learning Framework With BERT for Spoken Language Understanding. *IEEE Access*, 7: 168849–168858.