TOWARDS BETTER CONFIDENCE ESTIMATION FOR NEURAL MODELS

*Vishal Thanvantri Vasudevan**

University of California, San Diego
Department of Computer Science and Engineering
San Diego, California, USA
vthanvan@eng.ucsd.edu

Abhinav Sethy, Alireza Roshan Ghias

Alexa AI
Amazon
Seattle, Washington, USA

ABSTRACT

In this work we focus on confidence modeling for neural network based text classification and sequence to sequence models in the context of Natural Language Understanding (NLU) tasks. For most applications, the confidence of a neural network model in it's output is computed as a function of the posterior probability, determined via a softmax layer. In this work, we show that such scores can be poorly calibrated [1]. We propose new ensemble and gradient based features that predict model uncertainty and confidence. We evaluate the impact of these features through a gradient boosted decision tree (GBDT) framework to produce calibrated confidence scores. We demonstrate that the performance of our proposed approach surpasses the baseline across multiple tasks. Moreover, we show that this method produces confidence scores which are better suited for Out-Of-Distribution(OOD) classification when compared to the baseline.

Index Terms— Confidence modeling, uncertainty estimation, out-of-distribution classification

1. INTRODUCTION

In most intelligent personal digital assistant systems like Alexa, Google Assistant and Siri, there are multiple Natural Language Understanding (NLU) domains, which can provide a response to the user query [2]. In order to evaluate which of these responses is best suited, the system needs to have a measure of how confident each of these answer providers is about their response. Having a better confidence estimation for the competing models enables reliable decision making in terms of which response to choose. This is especially important in the case of third party response providers (skills) competing with internal components [3].

Neural networks are central to many of the component NLU systems, and are used as classifier and sequence prediction models. Typically the models use softmax to assign scores to output labels. However, most of these models tend to misclassify examples with high softmax probability, which is undesirable.

Our main contributions in this paper are new features based on ensemble diversity and gradient based measures that correlate with model confidence and an algorithm to combine these features with the posterior probability baseline using a regression model with instance-level $\{0/1\}$ accuracy as target. We show that the proposed features and confidence prediction model produce a more calibrated confidence score. We apply our algorithm to various utterance classification tasks such as intent classification, domain classification, third party (3P) skill identification[3] and measure the performance of the model by evaluating metrics such as probability alignment score, soft F1 score [1], reliability diagrams [4] and correlation coefficient with respect to instance-level accuracy. We compare these metrics to the baseline which is the softmax probability of the prediction. In addition, we evaluate the performance of the confidence score by applying it to OOD classification for first and third party skills [5] and show that it surpasses the baseline.

The rest of the paper is structured as follows. In section 2, we explore the related work in this domain. We introduce the uncertainty features (gradient, ensemble) we use in section 3. In section 4, we elaborate on our proposed training algorithm for the confidence model and the experimental setup along with the datasets used. We also provide background on the various metrics used to evaluate confidence models. We compare the performance of our model with that of the baseline and also it's ability to classify OOD data in section 5. We conclude with a summary of our findings in Section 6

2. RELATED WORK

Calibrated confidences are known to be critical for applied NLU systems [6] [7]. [1] Presents a baseline for confidence scores in the context of neural models and shows that the baseline can be surpassed by using an auxiliary decoder while training. [8] Explores uncertainty estimation for machine translation by analyzing the model distribution. As an alternative to training multiple models for producing more calibrated confidence scores, [9] Proposes the use of network at the end of each epoch as a different model. [4] Provides an algorithm to calibrate the predictions by extending platt scaling. [10] Proposes an approach to measure uncertainty of Convolutional Neural Networks (CNN) by using gradient features during test time. [11] Proposes a non-Bayesian approach similar to the ones above by using deep ensembles to measure uncertainty. [12] Proposes a regression based model framework for confidence estimation.

Various metrics to evaluate confidence scores have been proposed previously. [1] Introduces two metrics namely, Probability Alignment Score (PAS) and soft F1 score which are representations of accuracy and F1 score of the model, weighted by the confidence of the model's prediction. [4] Propose reliability diagrams as a visual representation of a models calibration. The diagrams plot accuracy as a function of the confidence score. Apart from these metrics, one

*The first author performed the work while interning at Alexa AI.

Measure	Intent Classification	Domain Classification	Skills Classification	Query-Rewriting
Train-Set Examples	160,000	160,000	89,000	12,000,000
Dev-Set Examples	24,000	24,000	5,000	42,000
Test-Set Examples	27,000	27,000	5,000	42,000
Number of Output Classes	1,300	800	1,400	-

Table 1: Dataset statistics

other very important metric for evaluating confidence scores used is the correlation of the confidence score to instance-level accuracy, similar to [12].

3. FEATURES FOR MODELING UNCERTAINTY

In this section, we introduce our ensemble uncertainty features based on softmax output probabilities as well as gradient based features.

3.1. Ensemble Uncertainty Features

Neural network models are high variance learners. Neural models with the same architecture trained on the same training data with different initializations and data sampling order can be viewed as multiple experts, each with a different view of the data [11]. The degree of agreement between multiple experts on a prediction, can be a good predictor of the correctness of the prediction. To measure the agreement of these experts for each prediction, we first predict the probability distribution over all the outputs for each model. The mean of the output distribution is computed and the Kullback-Leibler(KL) divergence of each model's output distribution with the mean output distribution is computed. With this, we obtain a KL divergence value for each model and data-point. Taking the mean and variance of these KL values, we get the proposed MeanKL and VarKL features (Algorithm 1).

Algorithm 1 Ensemble Uncertainty Features

Input: $P_{\Theta,x} = P_{\theta_1,x}, P_{\theta_2,x}, \ldots, P_{\theta_n,x}$, where $P_{\theta_i,x}$ is the probability distribution over output classes for model with parameters θ_i for input x
Output: $MeanKL_x, VarKL_x$
Procedure:
$meanPD_x \longleftarrow$ mean($P_{\Theta,x}$)
$KLValues_x \longleftarrow \emptyset$
for i *in* $1, 2, \ldots n$ **do**
  $KLValues_x[i] \longleftarrow$ KLDivergence($meanPD_x$, $P_{\theta_i,x}$)
end
$MeanKL_x \longleftarrow$ mean($KLValues_x$)
$VarKL_x \longleftarrow$ variance($KLValues_x$)
Return: $MeanKL_x, VarKL_x$

3.2. Gradient Uncertainty Features

As proposed by [10], we employ gradient based features as a sign of 're-learning-stress' in addition to the ensemble features. These features can seen as a measure of the model's uncertainty. Model parameter gradients are computed with respect to the loss given the output distribution and the predicted class as the target. The gradients of the embedding matrix can provide information about input uncertainty or how sensitive the prediction is to the input text. The statistical measures such as mean, variance, minimum, maximum and absolute sum are used to represent the 're-learning-stress'. These statistics computed with respect to all parameters except the embedding layer parameters are used as measures of model uncertainty. A input uncertainty feature is represented by computing the same statistics with only the embedding layer parameters taken into consideration.

Algorithm 2 Gradient Uncertainty Features

Input: M_θ, x where M_θ is Model with parameters θ and x is the data-point
Output: $GradStats_{\theta,x}$
Procedure:
$outputPred_{\theta,x} \longleftarrow M_\theta(x)$
$predClass_{\theta,x} \longleftarrow$ argmax($outputPred_{\theta,x}$)
$target_{\theta,x} \longleftarrow$ OneHotEnc($outputPred_{\theta,x}.size, predClass_{\theta,x}$)
$loss \longleftarrow$ CrossEntropy($target_{\theta,x}, outputPred_{\theta,x}$)
$Grad_{\theta,x} \longleftarrow$ Gradient($\theta, loss$)
for $pool$ *in* $(max, min, mean, var, sum)$ **do**
  $GradStats_{\theta,x}[pool] \longleftarrow pool(Grad_{\theta,x})$
end
Return: $GradStats_{\theta,x}$

In the next section we describe how we use the ensemble and gradient based features along with posterior probabilities to train a confidence prediction model on various NLU tasks (Algorithm 2).

4. EXPERIMENTAL SETUP AND DATASET DESCRIPTION

We tested our approach on three sentence classification tasks and a query rewriting task on subsets of Alexa NLU datasets collected from random users. The tasks were created from a subset of Alexa data and were chosen to do a controlled evaluation of different approaches for confidence modeling and not as a means to improve the accuracy of the production system. For classification tasks, the datasets used were for intent (first party skills) classification, domain classification and skill (third party skills) classification. Each of these datasets have a sentence as input and corresponding target (intent, domain, skill) as output. The query rewriting task is a sequence prediction task where we model consecutive friction utterances from Alexa users where the first utterance was unsuccessful and the second one was successful. In this task, we want to learn from users how to fix an unsuccessful utterance to a successful one. The dataset included one month of friction utterances, where we used the last day as a validation set, and the rest for training purpose. Table 1

Task	Correlation with instance-level accuracy		Probability Alignment Score		Soft F1 Score	
	Baseline	GBDT	Baseline	GBDT	Baseline	GBDT
Intent Classification	0.6500	**0.7782**	0.8271	**0.8626**	0.4966	**0.6318**
Domain Classification	0.6910	**0.7752**	0.8253	**0.8772**	0.3842	**0.6385**
Skill Classification	0.6013	**0.6616**	**0.7023**	0.6938	0.4481	**0.5390**
Query Rewriting	0.4277	**0.5425**	**0.4006**	0.3967	**0.7954**	0.7854

Table 2: Confidence calibration metrics evaluated on the posterior probability baseline and the proposed features incorporated in a GBDT model.

describes the size of our dataset for these tasks.

Next we look at the design of our classification and query rewrite models, followed by our approach to use the features presented in Section 3 for confidence modeling.

4.1. Classification and query rewrite models

The classification models have similar architectures. The input is transformed into vector representations by an embedding layer. The sequential input is fed into an LSTM[13] unit. The hidden state representation of the last time-step is fed into a linear layer which transforms it into the output vector space. Softmax activation is applied to this vector to produce a probability distribution over possible labels.

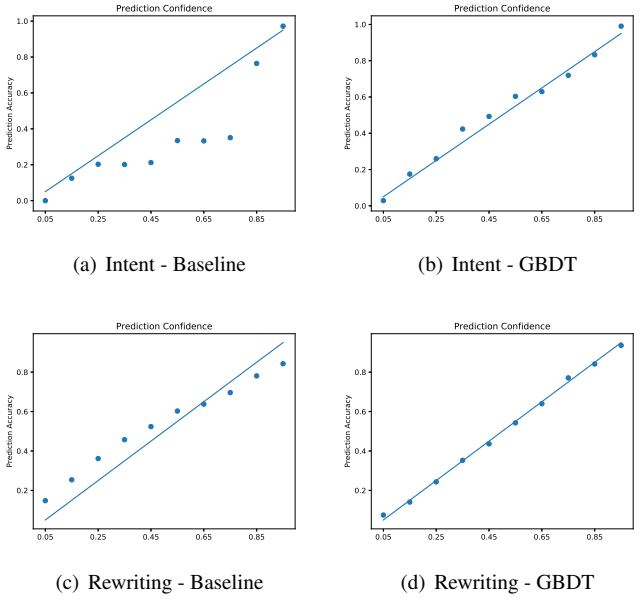
The rewriting model is a seq2seq model with a Recurrent Neural Network (RNN)-based encoder and decoder with attention [14] and copy [15] mechanisms.

4.2. Confidence Model

The confidence model for each task is trained on the the dev-set of the corresponding task, so that the model scores and other features are more representative of test or evaluation condition. We generate ensemble and gradient feature representations (Section 3) for each dev-set instance, along with the posterior probability of the predicted class. A gradient boosting decision tree (GBDT) regressor [16] model is trained with these features as inputs and the instance-level {0,1} prediction error as the target. For the query rewriting model, the instance-level accuracy is calculated by comparing if both the ground truth and rewritten query are mapped to the same intent and slots. The models are evaluated on the test-set of the corresponding tasks.

As an ensemble of multiple models is used, the prediction is computed by pooling the output probability distributions through summation of each models softmax output and choosing the class with the maximum value as the predicted class. Instance-level prediction error labels are computed by matching this prediction with the ground-truth. The GBDT model is trained with the default settings in the Scikit-Learn library and the output of the regression model is clipped to the range [0,1]. Moreover, the gradient features are computed with respect to each model in the ensemble and the mean of each statistical measure is taken across the models.

The features used in the confidence model for the query rewriting task are extensions of the features used in the classification tasks. Due to the sequential nature of the outputs generated, pooling operations (minimum, maximum, mean, variance) are performed on the probability distributions across time-steps to remove the dependency on the output sequence length. Additionally, the MeanKL and VarKL features are pooled across time-steps as well. The Seq2Seq model produced a probability of generation(p_gen) at each time-step during prediction. These probabilities were used as features as well.



(a) Intent - Baseline (b) Intent - GBDT

(c) Rewriting - Baseline (d) Rewriting - GBDT

Fig. 1: Reliability diagrams for intent classification and query rewriting. The solid line represents the reliability plot for a perfectly calibrated model.

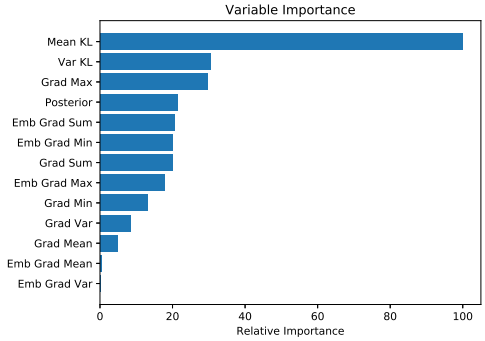
5. RESULTS & ANALYSIS

We compare our proposed approach to the baseline which is the posterior probability. Table 2 shows the Pearson correlation coefficients for the confidence scores generated by our confidence models and the accuracy of the predictions. It can be clearly seen that our model outperforms the baseline in all four cases. For the probability alignment metric, our confidence model performs better than the baseline in two of the four cases, however the difference is minimal in the other two cases. An explanation for this observation could be that the confidence models are unable to quantify uncertainty due to the poor performance of the task specific models they were trained on. This is observed in the case of the soft F1 score as well.

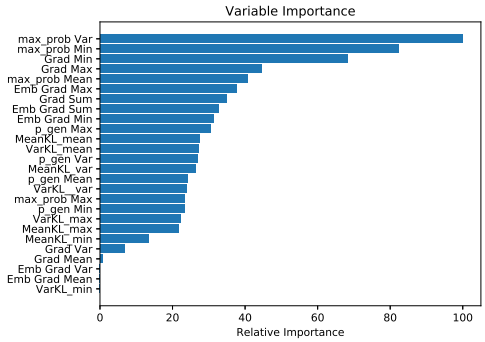
Reliability diagrams are plotted by computing accuracy as a function of the confidence score. This is done by binning the confidence scores into specific intervals and computing accuracy of examples in each interval. As per the definition of confidence, a

Experiment	Metric	Threshold				
		0.01	0.025	0.05	0.1	0.2
First Party Skills Data	Baseline	71.87%	68.09%	64.16%	58.73%	49.06%
	GBDT Model	**76.78%**	**75.29%**	**72.65%**	**67.54%**	**61.31%**
Third Party Skills (Intent) Data	Baseline	85.20%	51.82%	49.30%	44.35%	38.32%
	GBDT Model	**94.77%**	**94.58%**	**94.19%**	**92.28%**	**90.6%**

Table 3: Percentage of samples having the difference in the corresponding score predicted by the two models greater than the threshold. The difference for the skills data was computed between the metric predicted by the skills model and the intent model and vice-versa for intent data. The predicted confidence score outperforms the baseline for all thresholds.



(a) Intent classification



(b) Query rewriting

Fig. 2: Variable importance as per the regression model for each task.

model whose predictions have a confidence of x should have expected value of accuracy as x. Hence an ideal reliability diagram is an identity function or the solid line in Figure 1. As can be seen from Figure 1, our confidence model is highly calibrated for both sentence classification and query rewriting tasks. Note that we have only included the intent classification plots because of space constraints. The posterior probability in comparison, is not calibrated. The relative importance of the features used in the regression model can be observed in Figure 2. Features such as 'Emb Grad Sum', 'Grad Sum' represent the gradient features and 'Mean KL', 'Var KL' represent the ensemble features. For the query rewriting task, the features such as 'max_prob_var', 'max_prob_min' are features obtained by pooling the posterior probabilities of all time steps and features such as 'MeanKL_mean', 'VarKL_mean are obtained by pooling the Mean KL and Var KL features of all time steps. The gradient features and ensemble features play a significant role in the confidence score prediction. Confidence models that were trained using the same framework but without ensemble or gradient features do not surpass the baseline, proving the effectiveness of our proposed features in quantifying uncertainty.

For further analysis of the effectiveness of the proposed method, we compare the confidence scores predicted by the intent confidence model for the skills data and vice versa. Using two different models to classify the utterance as a match for first or third party skill gives us two predictions for each data-point. By choosing the prediction with higher confidence, we eliminate the need for a classifier that performs first party versus third party skills classification. Table 3 shows the number of examples from each dataset that had a difference in the confidence scores greater than the threshold. By varying the threshold, we modify the confidence gap and choose to proceed with the model with higher confidence or re-query the user for validation.

The proposed model clearly outperforms the baseline in both the cases, as can be seen from Table 3. The confidence model assigns a lower confidence score to utterances that must be serviced by the other set of skills and widens the confidence gap for a higher percentage of examples than the baseline. This shows that the proposed model provides scores better suited for detecting Out-of-Domain samples.

6. CONCLUSION & FUTURE WORK

By using ensemble and gradient features to represent uncertainty and combining the features with posterior probability, we demonstrate that our proposed confidence model outperforms the baseline in almost all cases with respect to the evaluation metrics used. Moreover, the proposed technique provided improvements on a sequence to sequence query rewriting task as well, showing that our approach can be adapted to other tasks by making minor changes to the features used.

The proposed model is computationally demanding due to the computation of gradients features and ensemble features. However the ensemble features can be computed much faster by parallelizing the forward pass of each of the models. A different avenue to explore would be to alter training schedules and architectures with an additional loss that calibrates posterior probabilities implicitly.

7. REFERENCES

[1] Dan Hendrycks and Kevin Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *CoRR*, vol. abs/1610.02136, 2016.

[2] R. Sarikaya, "The technology behind personal digital assistants: An overview of the system architecture and key components," *IEEE Signal Processing Magazine*, vol. 34, no. 1, pp. 67–81, Jan 2017.

[3] Young-Bum Kim, Dongchan Kim, Anjishnu Kumar, and Ruhi Sarikaya, "Efficient large-scale neural domain classification with personalized attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, vol. 1, pp. 2214–2224.

[4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger, "On calibration of modern neural networks," *CoRR*, vol. abs/1706.04599, 2017.

[5] Joo-Kyung Kim and Young-Bum Kim, "Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates," *Proc. Interspeech 2018*, pp. 556–560, 2018.

[6] Dilek Hakkani-Tur, Gokhan Tur, Giuseppe Riccardi, and Hong Kook Kim, "Error prediction in spoken dialog: from signal-to-noise ratio to semantic confidence scores," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*. IEEE, 2005, vol. 1, pp. I–1041.

[7] Marilyn Walker, Irene Langkilde, Jerry Wright, Allen Gorin, and Diane Litman, "Learning to predict problematic situations in a spoken dialogue system: experiments with how may i help you?," in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, 2000, pp. 210–217.

[8] Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato, "Analyzing uncertainty in neural machine translation," in *ICML*, 2018.

[9] Yonatan Geifman, Guy Uziel, and Ran El-Yaniv, "Boosting uncertainty estimation for deep neural classifiers," *CoRR*, vol. abs/1805.08206, 2018.

[10] Philipp Oberdiek, Matthias Rottmann, and Hanno Gottschalk, "Classification uncertainty of deep neural networks based on gradient information," in *ANNPR*, 2018.

[11] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017, pp. 6402–6413.

[12] Li Dong, Chris Quirk, and Mirella Lapata, "Confidence modeling for neural semantic parsing," in *ACL*, 2018.

[13] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[15] Abigail See, Peter J. Liu, and Christopher D. Manning, "Get to the point: Summarization with pointer-generator networks," in *ACL*, 2017.

[16] Tianqi Chen and Carlos Guestrin, "Xgboost: A scalable tree boosting system," in *KDD*, 2016.