Towards Improving Selective Prediction Ability of NLP Systems

Neeraj Varshney, Swaroop Mishra, Chitta Baral
Arizona State University
{nvarshn2, srmishr1, cbaral}@asu.edu

Abstract

It's better to say "I can't answer" than to answer incorrectly. This selective prediction ability is crucial for NLP systems to be reliably deployed in real-world applications. Prior work has shown that existing selective prediction techniques fail to perform well, especially in the out-of-domain setting. In this work, we propose a method that improves probability estimates of models by calibrating them using prediction confidence and difficulty score of instances. Using these two signals, we first annotate held-out instances and then train a calibrator to predict the likelihood of correctness of the model's prediction. We instantiate our method with Natural Language Inference (NLI) and Duplicate Detection (DD) tasks and evaluate it in both In-Domain (IID) and Out-of-Domain (OOD) settings. In (IID, OOD) settings, we show that the representations learned by our calibrator result in an improvement of $(15.81\%, 5.64\%)$ and $(6.19\%, 13.9\%)$ over *MaxProb* –a selective prediction baseline– on NLI and DD tasks respectively.

1 Introduction

In real-world applications, AI systems often encounter novel inputs that differ from their training data distribution. Prior work has shown that even state-of-the-art models tend to make incorrect predictions on such inputs (Elsahar and Gallé, 2019; Miller et al., 2020; Koh et al., 2021; Hendrycks et al., 2021). This raises reliability concerns and hinders their adoption in real-world safety-critical domains like biomedical and autonomous robots. *Selective prediction* addresses these concerns by enabling systems to abstain from making predictions when they are likely to be incorrect. Avoiding incorrect predictions allows them to maintain high task accuracy and thus makes them more reliable.

Hendrycks and Gimpel (2017) proposed '*MaxProb*' that uses the maximum softmax probability across all answer candidates as the confidence estimate to selectively make predictions. While performing reasonably well in the *in-domain* setting, *MaxProb* and other existing selective prediction techniques fail to translate that performance in the *out-of-domain* setting (Varshney et al., 2022b; Kamath et al., 2020).

In this work, we propose a selective prediction method that improves probability estimates of models in both in-domain and out-of-domain settings by learning strong representations via calibration. Specifically, we calibrate models' outputs using a held-out dataset and use the calibrator as confidence estimator for selective prediction. To this end, we first argue that "*all instances are not equally difficult and the model is not equally confident in all its predictions*" and then through extensive experiments, we show that prediction confidence is positively correlated with correctness while difficulty score is negatively correlated (5.2). We leverage the above finding to calibrate models' outputs using these two signals.

For computing the difficulty scores, we use a *model-based* technique (3.1) because human perception of difficulty may not always correlate well with machine interpretation. To calibrate a model, we annotate instances of a held-out dataset conditioned on the model's predictive correctness (computed using difficulty score and prediction confidence) and then train a calibrator using these instances. This annotation score represents the likelihood of correctness of the model's prediction. Finally, the trained calibrator predicts this likelihood value for test instances and is used as the confidence estimator for selective prediction.

To evaluate the efficacy of our method, we conduct comprehensive experiments in In-Domain (IID) and Out-of-Domain (OOD) settings for Natural Language Inference (NLI) and Duplicate Detection (DD) tasks. We also compare its performance with existing calibration techniques. On the NLI task, our method achieves 15.81% and 5.64% im-

Proceedings of the 7th Workshop on Representation Learning for NLP, pages 221 - 226
May 26, 2022 ©2022 Association for Computational Linguistics

provement on AUC of *risk-coverage* curve over *MaxProb* in IID and OOD setting respectively. Furthermore, on the DD task, it achieves 6.19% and 13.9% improvement in IID and OOD setting respectively. Finally, we hope that our work will facilitate development of more robust and reliable AI systems making their wide adoption in real-world applications possible.

2 Selective Prediction

Selective prediction enables a system to abstain on instances where it is likely to be incorrect i.e it consists of a *selector* (g) that determines if the system should output the prediction. Usually, g comprises of a prediction confidence estimator $\tilde{g}$ and a threshold th that controls the abstention level:

$$g(x) = \mathbb{1}[\tilde{g}(x)) > th]$$

A selective prediction system makes trade-offs between $coverage$ and $risk$. For a dataset D, coverage at a threshold th corresponds to the fraction of answered instances (where $\tilde{g} > th$) and risk is the error on those answered instances.

With the decrease in th, coverage will increase, but the risk will usually also increase. The overall selective prediction performance across all thresholds is measured by the area under *risk-coverage curve* (El-Yaniv et al., 2010). **Lower the AUC, the better the system** as it represents lower average risk across all thresholds.

3 Method

We propose to train a confidence estimator that can assign higher scores to correctly predicted instances than incorrectly predicted ones. To this end, we leverage a held-out dataset and annotate it's instances conditioned on the model's predictive correctness. Specifically, we infer the model on the held-out dataset and annotate instances with a score such that correctly predicted instances get assigned a higher score than incorrectly predicted instances. This annotation score models the likelihood of the prediction being correct and is computed using the model's prediction confidence and difficulty level of the instance. Finally, a calibrator (regression model) is trained using this annotated held-out dataset and used as the confidence estimator for selective prediction.

We detail each component of our method and the intuition behind it in the following subsections.

3.1 Difficulty Score Computation

To compute difficulty score of an instance, we evaluate it after every training epoch and subtract the aggregated softmax probability assigned to the ground-truth answer from 1 i.e. for an instance i, difficulty score d_i is calculated as:

$$s_i = \frac{\sum_{j=1}^{E} c_{ji}}{E}$$

$$d_i = 1 - s_i$$

where the model is trained till E epochs and c_{ji} is prediction confidence of the correct answer given by the model after j^{th} training epoch. Note that c_{ji} is probability assigned to the correct answer not the maximum probability across all answer candidates. The intuition behind this procedure is that the *instances that can be consistently answered correctly from the early stages of training are inherently easy and should receive lower difficulty score than the ones that require a large number of training steps.* A similar method has been explored in Swayamdipta et al. (2020) for analyzing "training dynamics" but here we use it to quantify difficulty of the held-out instances.

3.2 Annotation Score Computation

We define annotation score for the held-out instances as a function of *softmax probability* outputted by the model and the *difficulty score*. We show that softmax score is positively correlated while difficulty score is negatively correlated with the predictive correctness i.e the system is more likely to be correct if the softmax score is high and difficulty score is low. Furthermore, in order to justifiably separate the scores for correct and incorrect prediction scenarios in the range 0 to 1, we push the scores above 0.5 in case of correct and below 0.5 in case of incorrect scenarios. Concretely, we use the following functions to compute this:

$$AS_1 = \begin{cases} 0.5 + \frac{maxProb}{2}, & \text{if correct} \\ 0.5 - \frac{maxProb}{2}, & \text{otherwise} \end{cases}$$

$$AS_2 = \begin{cases} 0.5 + \frac{s_i}{2}, & \text{if correct} \\ 0.5 - \frac{s_i}{2}, & \text{otherwise} \end{cases}$$

$$AS_3 = \begin{cases} 0.5 + \frac{max(s_i, maxProb)}{2}, & \text{if correct} \\ 0.5 - \frac{min(s_i, maxProb)}{2}, & \text{otherwise} \end{cases}$$

AS_1 uses only softmax, AS_2 uses only difficulty score and AS_3 uses a combination of both. These

annotation strategies assign a relatively higher score when the model's prediction is correct and a lower score when it is incorrect. This gold score ranges from 0 to 1 as both s_i and $maxProb$ lie in the same range and better captures the likelihood of correctness unlike the categorical labels (1 for correct and 0 for incorrect) used in typical calibration approaches. **Note that this annotation computation is only required for training the calibrator and not at test time.** Therefore, difficulty score of the test instances need not be computed.

Both difficulty score and annotation score computation procedures are generic and are widely applicable since NLP systems usually make probabilistic predictions for all kinds of tasks ranging from Classification to Question Answering.

3.3 Calibration

Equipped with annotation scores, we extract syntactic features, namely, lengths, Semantic Textual Similarity (STS) value, number of common words between given sentences, and presence of negation words / numbers from the held-out instances to train the calibrator model. These features along with maxProb and prediction outputted by the model serve as inputs for the calibrator. Finally, we use a simple random forest implementation of Scikit-learn (Pedregosa et al., 2011) to train our calibrator that learns strong representations for the inputs. We note that these syntactic features are general and applicable for all language understanding tasks and any regression model can be used as the calibrator. We compare our method with other calibration techniques described in Section 4.1.

4 Experimental Setup

4.1 Calibration Baselines

Kamath et al. (2020) study a calibration-based selective prediction technique for Question Answering datasets where they annotate a held-out dataset such that correctly predicted instances are assigned class label '1' and incorrect ones are assigned label '0'. Then, a calibrator is trained using this annotated binary classification dataset using features such as input length and probabilities of top 5 predictions. The softmax probability assigned to class '1' by this calibrator is used as the confidence estimator for selective prediction. We refer to this approach as **Calib C**. We also train a transformer-based model for calibration (**Calib T**) that leverages the entire input text for this classification task instead of the syntactic features (Garg and Moschitti, 2021).

Our proposed calibration method differs from these approaches as we quantify the correctness on a continuous scale (instead of categorical labels '1' and '0') using prediction confidence and difficulty of the instances and use explicitly provided general syntactic features described in Section 3.3 for training. Our annotation procedure provides more flexibility for the calibrator to look for fine-grained features distinguishing various annotation scores. We note that our simplest annotation strategy (AS_1) that does not incorporate difficulty score is similar to Calib R method described in Varshney et al. (2022b) but our calibration method uses more general syntactic features.

Note that for fair estimation of abilities of the proposed method, we compare it with other calibration-based techniques only. Other techniques such as Monte-Carlo dropout (Gal and Ghahramani, 2016) and Error Regularization (Xin et al., 2021) are complementary and can further improve our performance.

4.2 Datasets

We conduct experiments with Natural Language Inference and Duplicate Detection datasets and compare the performance of various calibration techniques in in-domain and out-of-domain settings.

NLI Datasets: SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018) (Matched and Mismatched), and Stress Test (Naik et al., 2018) (Competence, Distraction, and Noise).

Duplicate Detection Datasets: QQP (Iyer et al., 2017) and MRPC (Dolan and Brockett, 2005).

For NLI task, we train 3-way classification model (NLI has three labels) on SNLI and evaluate the selective prediction performance on SNLI (IID) and MNLI, Stress Test (OOD) datasets. For the DD task, we train model on MRPC and evaluate on MRPC (IID) and QQP (OOD) datasets. We use BERT-BASE model (Devlin et al., 2019) with a linear layer on top of [CLS] token representation for training the model for these tasks. We train these models with the default learning rate of $5e-5$ for 3 epochs.[1] We use the same experimental setup as (Varshney et al., 2022b) for calibration methods.

[1] See Appendix for details

Method	SNLI	MNLI			Stress Test			
		Matched	Mismatched	Avg	Competence	Distraction	Noise	Avg
MaxProb (AUC)	2.78	14.00	14.44	14.22	47.87	26.49	20.34	31.57
Calib T (%)	-181.2	-129.55	-127.86	-128.69	-48.65	-81.3	-91.17	-68.93
Calib C (%)	+8.97	+2.15	-1.36	+0.40	-3.75	**+8.27**	-0.80	+0.55
Proposed (%)	**+15.81**	**+2.35**	**+2.04**	**+2.19**	**+8.01**	+6.60	**+0.22**	**+5.64**

Table 1: Comparing percentage improvement of various calibration approaches on AUC of risk-coverage curve (over MaxProb) in in-domain (SNLI) and out-of-domain settings (MNLI, Stress Test) for NLI task.

Method	MRPC	QQP
MaxProb (AUC)	6.13	40.46
Calib T (%)	-148.87	+2.21
Calib C (%)	-0.82	+2.0
Proposed (%)	**+6.19**	**+13.9**

Table 2: Comparing % improvement of various calibration approaches on AUC of risk-coverage curve in IID (MRPC) and OOD (QQP) settings for DD task.

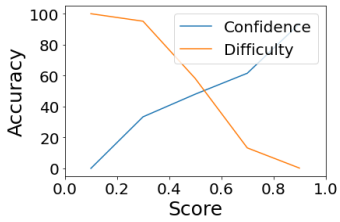

Figure 1: Trend of Model Accuracy with Confidence and Difficulty score for the NLI task.

5 Results and Analysis

5.1 MaxProb Struggles in OOD Setting

First rows in Table 1 and 2 show the AUC values achieved by MaxProb in NLI and DD tasks respectively. Note that in selective prediction, low AUC values of risk-coverage curves are preferred. We find that MaxProb performs well in the IID setting as it achieves low AUC values (2.78 on SNLI and 6.13 on MRPC). However, it fails to translate that in the OOD setting (AUC of 14.22 on MNLI, 31.57 on Stress Test, and 40.46 on QQP). This implies that the model makes a significant number of incorrect predictions with relatively high MaxProb and thus needs to be calibrated.

For calibration methods, we compare the performance improvement achieved over MaxProb w.r.t the minimum possible AUC.

5.2 Proposed Method Outperforms All

Our method shows a clear benefit over existing calibration techniques as it leads to a considerable improvement in all the cases. The proposed method achieves 15.81% and 6.19% improvement in the IID setting on SNLI and MRPC respectively. Furthermore, it achieves 2.19% on MNLI, 5.64% on Stress Test, and 13.9% on QQP in the OOD setting. *Calib T considerably degrades performance in both IID and OOD settings.* However, *Calib C results in a minor improvement in the IID setting* (8.97% for SNLI) *but does not consistently improve in the OOD setting* (especially on MNLI Mismatched and Competence Stress Test). We attribute this to the limited signal that is given to the calibrator by annotating the held-out dataset with categorical labels '1' and '0'. Thus, it learns weak representations.

Comparing Annotation Functions: We find that *the improvement using our method comes from using AS_3 as the annotation score* which outperforms AS_1 and AS_2. This is expected as it leverages useful signals provided by both maxProb and difficulty score for annotation computation.

Relationship With Predictive Correctness: To further analyze our method, we plot the relationship of predictive correctness with prediction confidence and difficulty score in Figure 1. It shows that prediction confidence is positively correlated while the difficulty score is negatively correlated with correctness. This further justifies our annotation score computation procedure.

6 Conclusion and Future Work

We proposed a selective prediction method that calibrates the model outputs using prediction confidence and difficulty level of the instances. Through comprehensive experiments, we demonstrated that it achieves considerable improvement over MaxProb on NLI and Duplicate Detection tasks in both IID and OOD settings. We hope that our work will facilitate development of more robust and reliable AI systems making their wide adoption in real-world applications possible.

Acknowledgements

We thank the anonymous reviewers for their insightful feedback. This research was supported by DARPA SAIL-ON and DARPA CHESS programs.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ran El-Yaniv et al. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5).

Hady Elsahar and Matthias Gallé. 2019. To annotate or not? predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2163–2173, Hong Kong, China. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Siddhant Garg and Alessandro Moschitti. 2021. Will this question be answered? question filtering via answer model distillation for efficient question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7329–7346, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. *data. quora. com.*

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. In *International Conference on Machine Learning*, pages 6905–6916. PMLR.

Swaroop Mishra and Anjana Arunkumar. 2021. How robust are model rankings: A leaderboard customization approach for equitable evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13561–13569.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. 2022a. Ildae: Instance-level difficulty analysis of evaluation data. *arXiv preprint arXiv:2203.03073*.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. 2022b. Investigating selective prediction approaches across several tasks in iid, ood, and adversarial settings. *arXiv preprint arXiv:2203.00211*.

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

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051, Online. Association for Computational Linguistics.

Appendix

A Related Work

Instance-level difficulty analysis has recently received considerable attention. Varshney et al. (2022a) explore five different applications of difficulty analysis of evaluation data such as conducting efficient yet accurate evaluations with fewer instances and estimating OOD performance reliably. Rodriguez et al. (2021) incorporate item response theory based difficulty quantification and analyze ranking reliability of leaderboards. Mishra and Arunkumar (2021) study robustness of model rankings by weighting instances based on their difficulty score. Swayamdipta et al. (2020) analyze the behavior of model on individual instances during training (*training dynamics*) and categorize training instances into three different difficulty regions.

B Experimental Details

We use batch size of 32 on Nvidia V100 16GB GPUs for our experiments. We train these models with the default learning rate of $5e-5$ for 3 epochs. In Calib T approach, we use BERT-BASE model as the calibrator and train it using the annotated held-out dataset. For training this calibrator, we use the default learning rate of $5e-5$. In the proposed approach, we use a simple random forest implementation of Scikit-learn (Pedregosa et al., 2011) to train the calibrator. Note that more advanced regression models could be used to further improve the performance of our approach. However, we leave that for future work as the focus of this paper is to show efficacy of our proposed approach on the selective prediction task.

C Features of Training Calibrator

We extract syntactic features, namely, lengths, Semantic Textual Similarity (STS) value, number of common words between given sentences, and presence of negation words / numbers from the held-out instances to train the calibrator model. These features along with maxProb and prediction outputted by the model serve as inputs for the calibrator.

For the NLI task, we compute these features for premise and hypothesis sentences i.e. STS value, number of common words, etc. between premise and hypothesis sentences.

Similarly, for the DD task, we compute these features for the given two sentences.