Towards More Accurate Uncertainty Estimation In Text Classification -Appendix

Jianfeng He†**, Xuchao Zhang**†**, Shuo Lei**$^{\dagger *}$**, Zhiqian Chen**$^+$**, Fanglan Chen**†**,**
Abdulaziz Alhamadani†**, Bei Xiao**‡**, Chang-Tien Lu**†
†Discovery Analytics Center, Virginia Tech, Falls Church, VA, USA
$^+$ Computer Science and Engineering, Mississippi State University, Starkville, Mississippi, USA
‡Department of Computer Science, American University, Washington, DC, USA
†{jianfenghe, xuczhang, slei, fanglanc, hamdani, ctlu}@vt.edu,
$^+$zchen@cse.msstate.edu, ‡bei.xiao@american.edu

1 Organization of Appendix

In this appendix, we show the supplementary material of this paper "Towards More Accurate Uncertainty Estimation In Text Classification". Firstly, we present more experiment results and more experiment settings. The experiment results are obtained from Transformer model and RNN model, which are also important in machine learning and different from CNN model. The experiment settings are further details about our experiments, such as computing infrastructure. Secondly, we show more Natural Language Processing (NLP) works related to uncertainty, and different overconfidence.

2 Appendix of Experiments

2.1 Results of transformer and RNN model

Concretely, we apply XLnet (Yang et al., 2019) as an example of the Transformer on Amazon dataset. The XLnet is pretrained from (Wolf et al., 2019) where it has 12 layers, 12 heads and dimension of hidden state as 768. We apply the feature from the last hidden state of the XLnet, followed by two trainable layers, which is added by us. The two trainable layers are FC1 layer (768→ 768) and FC2 layer (768→ 5), because of 5 sentimental labels in Amazon. Only for this task, we apply its own word embedding rather Glove embedding with dimension of 200. The Micro F1 of XLnet on Amazon is shown as Table 1. To show the flexibility of MSD, we try MSD2-b, which has two components as mix-up and distinctiveness score, and apply MSD2-a to represent default MSD2 setting, where it only has mix-up and self-ensembling.

Besides, Bidirectional Gated Recurrent Units (BiGRU) (Jabreel et al., 2018) is applied as an example of RNN model with two hidden layers. The Micro F1 of BiGRU on Amazon is shown as Table 2.

From Tables 1 and 2, we can conclude:

Higher performance in micro F1 by MSD2(-a). From Tables 1 and 2, we find MSD2 and MSD2-a always achieve the highest results with little increase compared with the MSD1. This still shows the effect of mix-up and self-ensembling. Though MSD3 performs not competitively compared with MSD1 and MSD2, it still outperforms DE+Metric obviously in both F1 scores and improved ratios of F1 scores in BiGRU. The poor performance of MSD3 compared with MSD1 and MSD3 also shows the offsets in calculating uncertainty scores by summing two parts in the testing process. Thus, how to avoid offsets in reducing epistemic uncertainty in the training process will be our future work.

2.2 More experiment settings

Computing infrastructure. We do all experiments on two GPUs, which both are GTX 1080Ti. The RAM in our machine is 64 GB.

Running time. For the experiments on 20News, IMDb, Amazon with saved hidden states from pretrained XLnet, the training can be completed in 2 hours, and testing can be completed in 30 minutes. For the experiments on Amazon (CNN) and Amazon (BiGRU), the training can be finished in 6 hours, while testing is finished in 1 hour and 3 hours respectively. All training epoches are set as 4 and testing will repeat $k = 100$ times tryouts with the same dropout rate.

Number of parameters. We have 7 parameters totally. Concretely, the mix-up has 1 parameter: Ω, which is a lower boundary of the mix-up random ratio α; the self-ensembling has 2 parameters λ_1 and λ_2 which are the coefficients of the losses L_{KL_2} and L_{SE} respectively; the distinctiveness score has 4 parameters, where β_1 and β_1 are the coefficients of the penalty and Mahalanobis distance respec-

---

[*] Corresponding author.

Table 1: Accuracy of uncertainty scores shown by improvement of micro F1 scores for the Amazon (XLnet)

Methods(Ω, λ_2, γ_1, γ_2)	**Uncertainty Ratio(Micro F1, Improved Ratio)**				
	0%	**10%**	**20%**	**30%**	**40%**
DE	0.682	0.711(4.18%)	0.737(8.10%)	0.762(11.76%)	0.783(14.88%)
DE+Metric	0.686	0.717(4.47%)	0.744(8.49%)	0.771(12.46%)	0.796(16.12%)
MSD1 (1, 0, 1, 0)	0.683	0.722(5.63%)	0.754(10.43%)	0.789(15.46%)	0.821(20.19%)
MSD2-a (1, 0.01, 1, 0)	0.684	**0.723**(**5.72%**)	**0.758**(**10.92%**)	**0.792**(**15.88%**)	**0.824**(**20.57%**)
MSD2-b (1, 0, 1, 1)	0.683	0.717(5.12%)	0.740(8.39%)	0.755(10.59%)	0.783(14.76%)
MSD3 (1, 0.01, 1, 1)	0.684	0.722(5.51%)	0.749(9.37%)	0.767(12.12%)	0.791(15.56%)

Table 2: Accuracy of uncertainty scores shown by improvement of micro F1 scores for the Amazon (BiGRU)

Methods(Ω, λ_2, γ_1, γ_2)	**Uncertainty Ratio(Micro F1, Improved Ratio)**				
	0%	**10%**	**20%**	**30%**	**40%**
DE	0.712	0.747(4.97%)	0.780(9.60%)	0.783(9.96%)	0.783(9.97%)
DE+Metric	0.709	0.745(5.06%)	0.755(6.48%)	0.754(6.33%)	0.753(6.20%)
MSD1 (1, 0, 1, 0)	0.706	0.745(**5.60%**)	0.781(**10.63%**)	0.817(**15.82%**)	0.849(**20.30%**)
MSD2 (1, 0.1, 1, 0)	0.708	**0.748**(5.58%)	**0.783**(10.54%)	**0.819**(15.58%)	**0.850**(20.02%)
MSD3 (1, 0.1, 1, 0.1)	0.709	0.746(5.32%)	0.778(9.83%)	0.808(13.97%)	0.848(19.66%)

tively, and γ_1 and γ_2 are the coefficient of the reciprocal of winning scores and distinctiveness scores respectively.

Evaluation metrics link. Though we have explained evaluation metrics in the experiment section, we provide our metrics code in ”emnlp_eval.py” in our submitted zip file (downloading form “SupMat__Software ” of EMNLP website), which is revised from (Zhang et al., 2019b) metrics code “drop_entropy_eval.py” [1]. We revise their metrics, entropy calculated by bin count, into ours, reciprocal of winning score but still with dropout mechanism.

Hyperparameter configurations for best-performing models. We give the concrete parameter setting, which obtains best performance in certain DNN and component combinations, after MSDs in the first column of each Table, the remaining parameters are constants, which have been introduced in the model section.

3 Appendix of Related Work.

Comparison with previous NLP works related to uncertainty. We compare MSD with some recent NLP works in terms of uncertainty categories. The comparison is listed in Table. 3. From the table, we can conclude that few works in NLP consider uncertainty in a comprehensive way (at least three categories of uncertainty are considered). Two works are noteworthy. Firstly, though we summarize (Wang et al., 2020) considers the structural uncertainty by two models with different layer designs, compared with NAS, which tries hundreds of different neural architectures, it is not a good way to solve the structural uncertainty. Secondly, although (Dong et al., 2018) considers the same three categories of uncertainty in semantic parsing, the main differences between it and MSD are: (a) (Dong et al., 2018) only scales the uncertainty score in the tesing process, without interfering with the training process. While we apply both the training process and testing process for uncertainty score, because we assume the training process is more flexible and has more abundant information compared with the testing process. For examples, we apply the training process to boost the negative correlation between the training samples and their uncertainty by the mix-up. (b) Though we consider the three same categories of uncertainty, we have obvious difference in solving them. For examples, we solve the aleatoric uncertainty by the mix-up, while (Dong et al., 2018) applys Gaussian noise; we consider both the dropout and self-ensembling to solve pamametric uncertainty, while only dropout is considered in (Dong et al., 2018). (c) The uncertainty scores between two works are calculated differently. We apply the reciprocal of winning scores as our uncertainty

---

[1] https://github.com/xuczhang/UncertainDC/blob/master/

Table 3: Comparison between MSD and recent NLP works in terms of applied categories of uncertainty, where "A", "E", "P", and "S" represent aleatoric uncertainty, epistemic uncertainty, parametric uncertainty, and structural uncertainty respectively; "1" represents that the model considers respective uncertainty, while "0" represents that the model ignores the respective one.

Model	Task	A	E	P	S
(Onan et al., 2016)	Keyword extraction	0	0	1	0
(Nadeem et al., 2019)	Multi-modal classification	0	0	1	0
(Hama et al., 2019)	Image-caption retrieval	0	0	1	0
(Jagannatha and Yu, 2020)	Entities of interest	0	1	1	0
(Shen et al., 2019)	Document quality assessment	0	0	1	0
(Wang et al., 2020)	Machine Translation	0	1	1	1
(Wang et al., 2019)	Machine Translation	0	0	1	0
(Dong et al., 2018)	Semantic Parsing	1	1	1	0
(Zhang et al., 2019a)	Semantic Parsing	0	0	1	0
(Ebrahimi et al., 2017)	Text classification	1	0	1	0
(Liang et al., 2017)	Text classification	1	0	0	0
(Papadopoulos et al., 2019)	Text classification	0	1	0	0
(Vasudevan et al., 2019)	Text classification	0	1	1	0
(Xiao and Wang, 2019)	Text classification	0	1	1	0
(Zhang et al., 2019b)	Text classification	0	0	1	0
MSD	Text classification	1	1	1	0

scores, while (Dong et al., 2018) calculates the uncertainty score by inputting confidence metrics to gradient tree boosting model (Chen and Guestrin, 2016).

Different overconfidence. Though (Thulasidasan et al., 2019) also mentions overconfidence, their overconfidence is different from ours. Their overconfidence is that the model accuracy is prone to be lower than what is indicated by the predictive score. While our overconfidence means that we cannot guarantee negative correlation between the winning scores and sample uncertainty, because the winning scores of training samples are all set as 1 by one-hot labels. We name it also as overconfidence because the negative correlation is missing by the same winning scores, which should be designed differently. Though the winning scores of 1 means the highest confidence, we can make sample confidence different by mix-up, which will decrease the winning scores. Hence, compared with different but decreased winning scores, the original winning scores are overconfident. In other word, their overconfidence focuses on the two metrics of model performance, while ours focuses on the correlation between the winning scores and sample uncertainty, which also shows bias in the latent assumption.

References

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. *arXiv preprint arXiv:1805.04604*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Kenta Hama, Takashi Matsubara, Kuniaki Uehara, and Jianfei Cai. 2019. Exploring uncertainty measures for image-caption embedding-and-retrieval task. *arXiv preprint arXiv:1904.08504*.

Mohammed Jabreel, Fadi Hassan, and Antonio Moreno. 2018. Target-dependent sentiment analysis of tweets using bidirectional gated recurrent neural networks. In *Advances in Hybridization of Intelligent Methods*, pages 39–55. Springer.

Abhyuday Jagannatha and Hong Yu. 2020. Calibrating structured output predictors for natural language processing. *arXiv preprint arXiv:2004.04361*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

Uzair Nadeem, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri. 2019. Learning-based confidence estimation for multi-modal classifier fusion. In *International Conference on Neural Information Processing*, pages 299–312. Springer.

Aytuğ Onan, Serdar Korukoğlu, and Hasan Bulut. 2016. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, 57:232–247.

Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. 2019. Outlier exposure with confidence control for out-of-distribution detection. *arXiv preprint arXiv:1906.03509*.

Aili Shen, Daniel Beck, Bahar Salehi, Jianzhong Qi, and Timothy Baldwin. 2019. Modelling uncertainty in collaborative document quality assessment. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 191–201.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 13888–13899.

Vishal Thanvantri Vasudevan, Abhinav Sethy, and Alireza Roshan Ghias. 2019. Towards better confidence estimation for neural models. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7335–7339. IEEE.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving back-translation with uncertainty-based confidence estimation. 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 791–802.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. *arXiv preprint arXiv:2005.00963*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7322–7329.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Xiang Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2019a. Adansp: Uncertainty-driven adaptive decoding in neural semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4265–4270.

Xuchao Zhang, Fanglan Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2019b. Mitigating uncertainty in document classification. In *Proceedings of NAACL-HLT*, pages 3126–3136.