2021 International Conference on Computational Science and Computational Intelligence (CSCI) | 978-1-6654-5841-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSCI54926.2021.00024

Uncertainty Estimation for Twitter Inference

Meng Hsiu Tsai
Department of Computer Science and Engineering
University of Tennessee at Chattanooga
Chattanooga, United States
e-mail: wmf223@mocs.utc.edu

Nicole Marie Ely
Department of Data Science and Business Analytics
Florida Polytechnic University
Lakeland, United States
e-mail: nely7539@floridapoly.edu

Yingfeng Wang*
Department of Computer Science and Engineering
University of Tennessee at Chattanooga
Chattanooga, United States
e-mail: yingfeng-wang@utc.edu
*Corresponding author

***Abstract*—Twitter provides a platform for users to express their opinions in the form of Twitter messages called tweets. Analyzing tweets from a specific city or demographical group requires geographical or demographical information. Unfortunately, most Twitter users do not provide these details in their profiles. To overcome this challenge, tools have been developed for inferring users' geographical and demographical information from Twitter data. Using inference results is risky due to the lack of uncertainty estimation of these results. Here, we present a framework to estimate uncertainties of Twitter inference results. The effectiveness of this framework is verified in experiments.**

Keywords-Uncertainty Estimation; Bayesian Neural Network; Knowledge Distillation; Twitter Inference

I. INTRODUCTION

Twitter inference aims to infer latent attributes of Twitter data. Geolocation inference focuses on identifying the locations of Twitter users [1], while demographic inference addresses the demographical information of Twitter users [2]. These inferred attributes provide necessary details for Twitter data analysis. For instance, geolocation data can be applied in the analysis of local event impacts. Twitter inference has received increasing attention. However, the uncertainties of these inferences threaten the reliability of the downstream analysis. To address this threat, it is necessary to estimate the uncertainties of inference results.

Estimating uncertainties of Twitter inference results is crucial but challenging. When inferring latent information, both input data and inference methods may bring uncertainties into the results. All uncertainties will threaten the reliability of Twitter data analysis using inference results. For example, an inference may mislabel a tweet post by an Alabama-based user as a tweet post by a Tennessee-based user. The analysis results based on this inference could be misleading. Therefore, it is critical to estimate the uncertainties of inference results. Unfortunately, uncertainty estimation is challenging. Although there are some available models for uncertainty estimation, e.g., Bayesian neural network (BNN), most Twitter inference methods are not based on these models. One possible solution is to treat these methods as black-box models. Mena et al. developed a wrapper for estimating uncertainties of black-box models [3], [4]. But this wrapper can only estimate uncertainties from input data. Since this wrapper cannot access model parameters for measuring the uncertainties caused by inference models, this approach still cannot compute overall inference uncertainties. New approaches to bridge the gap between inference methods and uncertainty estimation are required.

Inspired by the uncertainty measurement wrapper of Mena et al., we propose a simple uncertainty estimation framework to address the challenge of uncertainty estimation. This framework is twofold. First, the originally trained models teach student BNNs by knowledge distillation. Second, the uncertainty estimation is then conducted on the student BNNs. Besides estimating inference uncertainty, this framework gives users flexibility in model selection. We test this framework on an existing Twitter inference model for tweet geolocation. Experimental results verify that our general framework can effectively estimate inference results.

II. RELATED WORKS

A. Bayesian Neural Network

Theoretically, we can estimate uncertainty using stochastic neural networks, which contain stochastic components. For example, some stochastic neural networks use stochastic weights, which are random variables following specific probability distributions. Also, some stochastic neural networks use stochastic activation functions, whose outputs are random variables. These stochastic components can assist in measuring output uncertainties. Training stochastic neural networks is non-trivial. Bayesian inference is a widely used training strategy. Stochastic neural networks trained by Bayesian inference are defined as Bayesian neural networks (BNNs) [5]. Two types of approaches are popular in BNN training: Markov Chain Monte Carlo (MCMC) and variational inference [6], [7]. MCMC is a class of statistical simulation methods based on the Markov Chain. In practice, the Metropolis-Hastings algorithm is the most used MCMC method for training BNNs [8]. To improve the scalability of BNN training, researchers developed variational inference methods that attempt to learn a relatively simple variational distribution from the training data. Because of the advantage

978-1-6654-5841-2/21/$31.00 ©2021 IEEE
DOI 10.1109/CSCI54926.2021.00024

on training large-size BNNs, variational inference methods have received significant research attention.

B. Knowledge Distillation

Knowledge distillation (KD) is a machine learning technique that transfers knowledge from a teacher model to a student model [9]. A teacher model is usually complex and learns well from original training data, while a student model is usually easy to analyze and deploy. Knowledge distillation is composed of two steps: the teacher model is trained on the original training set and is then used to train the student model(s). Three types of knowledge distillation have been studied: response-based knowledge distillation, feature-based distillation, and relation-based knowledge. These methods focus on distilling knowledge with different types of mappings or relations [10]. The proposed framework will use response-based knowledge distillation, which uses the input and corresponding output of the output layer of the teacher model to train the student model.

III. Method

Inspired by the uncertainty measurement wrapper of [3], [4], we propose a simple uncertainty estimation framework. This framework transfers the knowledge of trained inference models to BNN-based models and measures the uncertainties of BNN-based models. The framework is briefly outlined in Fig. 1 and detailed in the rest of this section.

A. Build BNN-based Student Models

Here, we present a guideline for building BNN-based student models to measure inference uncertainties. Different inference models have been used in Twitter inference [11]–[13]. If an inference model is a neural network, we can build the corresponding student BNN model based on the original model with the BNN component added to its output layer. BNN components can also be added to other internal layers of this neural network, but it requires additional computational time for knowledge distillation and uncertainty estimation. Therefore, we recommend adding BNN components only at the output layer. If an inference model is composed of many different neural networks, e.g., an ensemble of several neural networks, this framework will use the dominant neural network, whose architecture is dominant in the inference model, as the student model with the BNN component in the output layer. If an inference model is not using neural networks, we recommend designing a BNN-model whose input and output formats are consistent with the original inference model. Knowledge distillation prefers relatively simple student models for response-based distillation [10]. Therefore, we can use a simple network architecture that meets the requirements of input and output formats. Similarly, the BNN component is added to the output layer.

B. Transfer Knowledge and Measure Uncertainties

The proposed framework uses originally trained models as the teacher models in knowledge distillation, which distillates learned knowledge from original models to the BNN-based student models. The framework follows the standard steps of response-based knowledge distillation [14].

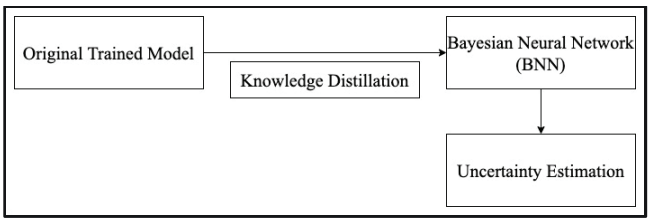

Fig. 1. Framework Outline

A Twitter inference model is essentially a multiclass classification model. The output classification results can be normalized as class probabilities by the softmax function, which is given as follows:

$$q(z)_i = e^{z_i} / \sum_{j=1}^{C} e^{z_j} \quad (1)$$

where z is the set of class prediction logits, z_i is the i-th element of z, C is the number of classes, and $q(z)_i$ is the i-th class prediction probability. To further improve performance, we consider temperature T in the following expression.

$$q(z,T)_i = e^{\frac{z_i}{T}} / \sum_{j=1}^{C} e^{\frac{z_i}{T}} \quad (2)$$

Equation (1) is the special case of (2) when T is 1. After T is specified, this framework applies (2) to normalize teacher models' outputs and uses normalized outputs to train student models. Also, the loss function used in training is the evidence lower bound (ELBO) loss function, which combines cross entropy and KL divergence [15].

The resulting student models can conduct inference on Twitter data like teacher models. Because the output layer of student models contains BNN components, the uncertainties of inference results can be measured. Specifically, this framework uses the uncertainty approximation method detailed by Chai [16]. As for an input tweet, the BNN-based model parameters are sampled for K times. The k-th sample of model parameters is represented by w^k. This framework applies the softmax function to normalize original outputs into probabilities of predicted classes. The probability of classifying the i-th testing tweet to class c based on w^k is denoted by $p_{i,c}^k$. The predictive uncertainty of i-th classification, represented by H_i, can be approximated as the entropy using the following expression:

$$H_i = -\sum_{c=1}^{C} \left[\left(\frac{1}{K} \sum_{k=1}^{K} p_{i,c}^k \right) log \left(\frac{1}{K} \sum_{k=1}^{K} p_{i,c}^k \right) \right] \quad (3)$$

where C is the number of classes.

C. Inference Model Selection After Uncertainty Estimation

The proposed framework applies knowledge distillation to switch inference models. After knowledge distillation is finished, student models can replace the original teacher models in Twitter inference. As for a specific inference result, if its uncertainty meets the user's need, the user can keep this result. Otherwise, the user can select another student model and repeat knowledge distillation and uncertainty measurement. This design gives users the flexibility of choosing models for lowering uncertainties.

IV. Experiments

To verify the effectiveness of our framework, we applied this framework to estimate the uncertainties of a geolocation inference tool [12].

A. Data Collection and Preprocessing

We tested our framework on the WNUT'16 shared task data, which was also used in [12]. Tweepy [17] collected 1,197,798 tweets according to tweet IDs of the WNUT'16 shared task data. Each tweet is associated with a city. We observed 2,965 different cities. All collected tweets were divided into training and testing sets. About 1% tweets were assigned to the testing set in [12]. Similarly, we randomly assigned 11,978 tweets for testing, while the rest of the tweets were assigned to the training set.

B. Baseline Models

Our experiments verified whether the uncertainties of student model outputs are close to that of teacher models. In [12], Thomas et al. developed an inference method using two types of neural networks. One was long short-term memory (LSTM). The other was perception with rectified linear unit (ReLU) activation function. These two types of neural networks handled different types of Twitter data such as text, user language, and time zone. The results of neural networks were combined for generating the final inference results. This inference model had two training modes. One was to train all neural networks independently first, then train the combination layer without updating parameters of LSTMs and perceptions. This mode was called full-fixed. Another mode, called full-scratch, was to train all neural networks and the combination layer at the same time. Our experiments used both modes as baseline methods. Furthermore, we directly added BNN components to the output layer, i.e., the combination layer, to build full-fixed-BNN and full-scratch-BNN. Both were also used as baseline models if they were trained without using knowledge distillation.

C. Knowledge Distillation Setting

In knowledge distillation experiments, we used training set input and teacher model output to train student models. Student models were tested on the original testing set. Our experiments tested two knowledge distillation settings. One was to use full-fixed-BNN as teacher and student models. The other was to use full-scratch-BNN as teacher and student models. The purpose was to verify whether the uncertainties of the student models were close to that of the corresponding baseline methods. Also, the experiments evaluated the impact of knowledge distillation on inference accuracy. The experiments were conducted with distillation temperature T equal to 1, 2, 5, 10, 20, and 50. The purpose was to evaluate the impact of distillation temperature on uncertainty.

D. Performance Measurement

We measure accuracy, overall uncertainty, and overall uncertainty difference in experiments. The model accuracy, represented by Acc, were measured by the following expression:

$$Acc = \frac{n}{KN} \tag{4}$$

where N is the number of tweets in the testing set, n is the number of predictions whose highest values refer to the correct classes, and K is the number of sampling times. Please note that K is 1 for non-BNN models such as full-fixed and full-scratch. The overall uncertainty of a model with respect to the testing set is represented by U, which can be computed by the following expression:

$$U = \sqrt{\frac{\sum_{i=1}^{N} H_i^2}{N}} \tag{5}$$

where N is the number of tweets in the testing set, and H_i is the uncertainty of the i-th tweet. H_i can be computed by (3) Similarly, we define the overall uncertainty difference between two inference models, represented by D, as follows:

$$D = \sqrt{\frac{\sum_{i=1}^{N} (H_{a,i} - H_{b,i})^2}{N}} \tag{6}$$

where $H_{a,i}$ and $H_{b,i}$ are uncertainties of the predictions of i-th tweet with inference models a and b, respectively.

E. Experimental Results

All experimental results are detailed in Tables I and II. Table I focuses on full-fixed and related BNN models, while Table II gives results of full-scratch and related BNN models. Table I shows the accuracies of BNN models are better than that of the non-BNN baseline model. Furthermore, the overall uncertainties of models trained by knowledge distillation are close to that of the BNN model trained without using knowledge distillation. It also matches the small overall uncertainty differences between the models with and without using knowledge distillation. Additionally, adjusting temperatures of knowledge distillation between 1 and 20 does not significantly change the overall uncertainty and uncertainty difference, while the overall uncertainty difference with temperature 50 is higher than that of other temperatures. Table II also suggests similar results on full-scratch. The accuracies of BNN model are consistently greater than that of full-scratch. Also, the overall uncertainties of BNN models based on knowledge distillation are close to that of the BNN model without using knowledge distillation. It is also reflected on small overall uncertainties differences between two types of BNN models. With the full-scratch mode, adjusting temperatures of knowledge distillation between 1 and 50 does not significantly change the overall uncertainty and uncertainty difference.

V. Discussion

This paper proposes a simple framework for estimating uncertainties of Twitter inference results. This framework transfers knowledge from the original models to BNN-based models. Experiments results show that the uncertainties of student models are close to that of teacher models if model architectures are similar. We can use this framework to estimate uncertainties of original inference models, whose most model details are clear. Furthermore, we can also use this framework to select inference models with low uncertainties, even if the original inference models are black-

TABLE I. RESULTS ON MODELS RELATED TO FULL-FIXED

Models	Accuracy	Overall Uncertainty	Overall Uncertainty Difference
Full-Fixed	0.693	NA	NA
Full-Fixed-BNN without KD	0.773	2.88	0
Full-Fixed-BNN with KD T=1	0.737	3.14	0.900
Full-Fixed-BNN with KD T=2	0.739	3.14	0.869
Full-Fixed-BNN with KD T=5	0.744	3.20	0.900
Full-Fixed-BNN with KD T=10	0.742	3.23	0.916
Full-Fixed-BNN with KD T=20	0.740	3.26	0.963
Full-Fixed-BNN with KD T=50	0.699	3.50	1.394

TABLE II. RESULTS ON MODELS RELATED TO FULL-SCRATCH

Models	Accuracy	Overall Uncertainty	Overall Uncertainty Difference
Full-Scratch	0.654	NA	NA
Full-Scratch-BNN without KD	0.687	1.69	0
Full-Scratch-BNN with KD T=1	0.793	2.66	1.68
Full-Scratch-BNN with KD T=2	0.815	2.60	1.65
Full-Scratch-BNN with KD T=5	0.833	2.61	1.65
Full-Scratch-BNN with KD T=10	0.838	2.59	1.64
Full-Scratch-BNN with KD T=20	0.836	2.60	1.64
Full-Scratch-BNN with KD T=50	0.830	2.65	1.68

boxes. According to Table II, in comparison of the BNN model without knowledge distillation, other BNN models have better accuracies and worse uncertainties. These results suggest accuracy improvement cannot guarantee uncertainty improvement. It highlights the importance of uncertainty estimation on inference results. The success of our framework on the experiments encourages us to extend it to more areas for uncertainty estimation.

VI. Conclusion and Future Work

This paper presents a simple framework for estimating uncertainties of Twitter inference results. This framework distillates learned knowledge from original inference models to BNN-based student models, and measures uncertainties on student models. The experimental results suggest that the proposed framework can accurately model the overall inference uncertainties. We will extend this framework to be a general tool for estimating predictions made by all machine learning models.

Acknowledgment

This work was partially supported by the National Science Foundation under grant number 1852042.

References

[1] O. Ajao, J. Hong, and W. Liu, "A survey of location inference techniques on Twitter," J. Inf. Sci., vol. 41, no. 6, pp. 855–864, 2015, doi: 10.1177/0165551515602847.

[2] X. Chen, Y. Wang, E. Agichtein, and F. Wang, "A comparative study of demographic attribute inference in Twitter," in Proceedings of the International Conference on Web and Social Media (ICWSM), 2015, pp. 590–593.

[3] J. Mena, O. Pujol, and J. Vitrià, "Dirichlet uncertainty wrappers for actionable algorithm accuracy accountability and auditability," in The Conference on Fairness, Accountability, and Transparency (FAT), 2020, pp. 581–581, doi: 10.1145/3351095.3372825.

[4] J. Mena, O. Pujol, and J. Vitria, "Uncertainty-based rejection wrappers for black-box classifiers," IEEE Access, vol. 8, pp. 101721–101746, 2020, doi: 10.1109/ACCESS.2020.2996495.

[5] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," Neural Comput., vol. 4, no. 3, pp. 448–472, 1992, doi: 10.1162/neco.1992.4.3.448.

[6] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," J. Am. Stat. Assoc., vol. 112, no. 518, pp. 859–877, 2017, doi: 10.1080/01621459.2017.1285773.

[7] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," Learn. Graph. Model., vol. 233, pp. 105–161, 1998, doi: 10.1007/978-94-011-5014-9_5.

[8] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," Am. Stat., vol. 49, no. 4, pp. 327–335, 1995, [Online]. Available: https://www.jstor.org/stable/2684568.

[9] L. Wang and K. J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," IEEE Trans. Pattern Anal. Mach. Intell., pp. 1–20, 2020, doi: 10.1109/TPAMI.2021.3055564.

[10] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," Int. J. Comput. Vis., vol. 129, no. 6, pp. 1789–1819, 2021, doi: 10.1007/s11263-021-01453-z.

[11] S. Mac Kim, Q. Xu, L. Qu, S. Wan, and C. Paris, "Demographic inference on twitter using recursive neural networks," in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2017, pp. 471–477, doi: 10.18653/v1/P17-2075.

[12] P. Thomas and L. Hennig, "Twitter geolocation prediction using neural networks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10713 LNAI, pp. 248–255, 2018, doi: 10.1007/978-3-319-73706-5_21.

[13] J. Cornelisse and R. G. Pillai, "Age Inference on Twitter using SAGE and TF-IGM," in Proceedings of the International Conference on Natural Language Processing and Information Retrieval (NLPIR), 2020, pp. 24–30, doi: 10.1145/3443279.3443300.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in NIPS Deep Learning and Representation Learning Workshop, 2015, pp. 1–9, [Online]. Available: http://arxiv.org/abs/1503.02531.

[15] X. Yang, "Understanding the variational lower bound." pp. 1–4, 2017, [Online]. Available: https://xyang35.github.io/2017/04/14/variational-lower-bound/.

[16] L. R. Chai, "Uncertainty estimation in Bayesian neural networks and links to interpretability," University of Cambridge, 2018.

[17] "Tweepy." https://www.tweepy.org/ (accessed Jun. 26, 2020).