Word-Level Uncertainty Estimation for Black-Box Text Classifiers using RNNs

Jakob Smedegaard Andersen[1], **Tom Schöner**[2] and **Walid Maalej**[1]
[1]University of Hamburg / [2]HAW Hamburg
Hamburg, Germany
{andersen,maalej}@informatik.uni-hamburg.de

Abstract

Estimating uncertainties of Neural Network predictions paves the way towards more reliable and trustful text classifications. However, common uncertainty estimation approaches remain as black-boxes without explaining which features have led to the uncertainty of a prediction. This hinders users from understanding the cause of unreliable model behaviour. We introduce an approach to decompose and visualize the uncertainty of text classifiers at the level of words. Our approach builds on top of Recurrent Neural Networks and Bayesian modelling in order to provide detailed explanations of uncertainties, enabling a deeper reasoning about unreliable model behaviours. We conduct a preliminary experiment to check the impact and correctness of our approach. By explaining and investigating the predictive uncertainties of a sentiment analysis task, we argue that our approach is able to provide a more profound understanding of artificial decision making.

1 Introduction

Neural Networks or variations of them achieve state-of-the-art accuracy across a wide range of text classification tasks like sentiment analysis (Nakov et al., 2016) or spam detection (Wu et al., 2017). Neural Networks are however not interpretable, since they provide no information about why particular decisions were made (Baehrens et al., 2010). This lack of transparency makes it hard for users to assess the certainty and trustfulness of predictions. In the worst case, an unreliable prediction is considered correct even if it is not. It is thus important to know what a model does not know. This would allow treating particular error prone or unreliable predictions with additional care – enabling a better understanding of why wrong predictions occur.

Several techniques to assess the uncertainty of individual predictions have been successfully applied to Neural Networks (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Kendall and Gal, 2017). However, a global uncertainty estimation for the whole input text does not describe which features lead to an uncertain prediction. It is thus unclear, why a Neural Networks classifier is, e.g., uncertain whether a text represents a positive sentiment or whether it is a spam.

As a first step towards a better understanding of Neural Networks-based text classification tasks, we suggest the decomposition of uncertainties on the level of words. We present a novel uncertainty modelling approach that estimates word-level uncertainties in any text classification task. Our approach applies Bayesian modelling to a sequence attribution technique for Recurrent Neural Networks (RNNs). We implement the approach using TensorFlow and Keras and demonstrate its effectiveness by investigating word-level uncertainties in a sentiment analysis task.

2 Word-Level Uncertainty Estimation

We first introduce how we model uncertainty in a classification task and then describe how we decompose the prediction uncertainties on the level of words.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 28th International Conference on Computational Linguistics, pages 5541–5546
Barcelona, Spain (Online), December 8-13, 2020

2.1 Modelling Predictive Uncertainty in Classification Tasks

To measure the uncertainty of Neural Networks predictions, we use the uncertainty modelling technique called Monte Carlo Dropout: According to Gal and Ghahramani (2016), the regularization method dropout (Srivastava et al., 2014) can be interpreted as a Bayesian approximation of a Gaussian process (Rasmussen, 2003). By enabling dropout at inference time, each forward pass uses a random sample of weights resulting in a probabilistic model. We obtain a sample of an approximated posterior distribution by feeding the same input e multiple times to the model. We approximate the mean predictive posterior probability denoted as $p(y = c|e, D_{train})$ by averaging the posterior probabilities of multiple forward passes. That is, $p(y = c|e, D_{train}) \approx \frac{1}{F}\sum_{f=1}^{F} p(y = c|e, \omega_f)$, where F is the number of forward passes, ω_f the parameters of the f^{th} sample, and D_{train} the data used for training.

A measurement of the predictive uncertainty regarding an input e is derived by analysing the statistical dispersion of the output distribution. Kwon et al. (2020) propose a natural way of estimating uncertainties in Neural Networks. Their approach relies on a variation of the law of the total variance:

$$Var[y = c|e, D_{train}] \approx \underbrace{\frac{1}{F}\sum_{f=1}^{F}(p_f - \bar{p})(p_f - \bar{p})^{\mathsf{T}}}_{\text{epistemic}} + \underbrace{\frac{1}{F}\sum_{f=1}^{F}\left(\text{diag}(p_f) - (p_f)(p_f)^{\mathsf{T}}\right)}_{\text{aleatory}} \tag{1}$$

where $\bar{p} = \frac{1}{F}\sum_{f=1}^{F} p_f$, p_f is a vector of posterior probabilities $p(y = c|e, \omega_f)$ for each class $c \in C$ of the f^{th} forward pass, and $\text{diag}(p_f)$ is a diagonal matrix with elements of the vector p_f. Equation 1 allows to decompose uncertainties in its aleatory or epistemic components (Der Kiureghian and Ditlevsen, 2009). Aleatory uncertainty captures irreducible noise and randomness inherent in the observation. Epistemic uncertainty has its source in inadequate and missing knowledge and can be reduced by additional learning.

2.2 Attribution of Recurrent Neural Network Inputs

We build our approach to decompose prediction uncertainties on the level of words on top of Long Short Term Memory (LSTM) models, a Recurrent Neural Network variation initially described by Hochreiter and Schmidhuber (1997). An LSTM consists of a range of repeated cells, each computing a hidden state h_t. For each index t of the input sequence e, the output of a corresponding cell is controlled by a set of gates as a function of a cell input x_t and the previous hidden state h_{t-1}. To use an LSTM for text classification, we add a discriminative layer after the last activation vector h_T to obtain class activation scores $S_c^{\omega}(e) = W_c h_T$ using a weight matrix W.

Since the i^{th} hidden state vector h_i is updated by prior hidden states, previous elements of the input sequence $(\epsilon_1, ..., \epsilon_{i-1})$ are already taken into account in the evaluation of h_i. Thus, $S_c^{\omega}(e_i) = W_c h_i$ describes the accumulated class activation of the first i word embeddings $e_i = (\epsilon_1, ..., \epsilon_i)$ of an input e. In order to assess the contribution of a single word to the final prediction, we decompose the final class activation score into the sum of multiple individual word contributions:

$$S_c^{\omega}(e) = \sum_{i=1}^{E} S_c^{\omega}(e_i) - S_c^{\omega}(e_{i-1}) = \sum_{i=1}^{E} W_c(h_i - h_{i-1}) \tag{2}$$

Computing the mean posterior probability $p(y = c|e_i, D_{train})$ for each index $1 \leq i \leq E$ by applying Monte Carlo Dropout allows us to assess the development of uncertainties along the input sequence. Analogous to Equation 2, we measure the word-level aleatory uncertainty U_a, epistemic uncertainty U_e, or total **uncertainty** U_t as the change of uncertainty contributed by a single word:

$$U_x(\epsilon_i) = U_x(e_i) - U_x(e_{i-1}), \quad x \in \{a, e, t\} \tag{3}$$

Additionally, we derive the **relevance** R_c of each word regarding its contribution to the final class activation score $S_c(e) = \frac{1}{F}\sum_{f=1}^{F} S_c^{\omega_f}(e)$. The relevance of a word is calculated as the class activation contribution by a word compared to its prior sequence:

$$R_c(\epsilon_i) = S_c(e_i) - S_c(e_{i-1}) \tag{4}$$

3 Experiments

We conducted a preliminary evaluation of the advantage and correctness of our approach, by applying it to a common sentiment analysis task. We use the IMDB dataset (Maas et al., 2011), which consists of polarized film reviews. In our experiments we use an LSTM with an additional dropout-layer after the embedding-layer with $p_{drop} = 0.5$. Further, we consider the LSTM configuration used in the official Tensorflow example[1] with pre-trained word2vec embeddings. Our implementation and experimental results are publicly available online[2].

3.1 Decomposition of Classifier Outputs

First, we study the information gained by decomposing Neural Networks predictions and their uncertainties. We append a clearly positive review with 239 words to a clearly negative review with 140 words. For the new created review, Figure 1a shows the path of the mean posterior across the word index i of the evaluated input sequence e_i. Figure 1b plots the corresponding total uncertainty as well as its aleatory and epistemic components. At the beginning of the second review, the mean posterior probability drops and starts to become highly uncertain. Furthermore, Figure 1b shows that the uncertainty starts to increase when the sentiment shifts. Thus, our approach seems to correctly infer sentiment changes in the input sequence. Overall this example indicates that the decomposition of Neural Networks outputs can provide valuable information to support the understanding of Neural Networks decisions.

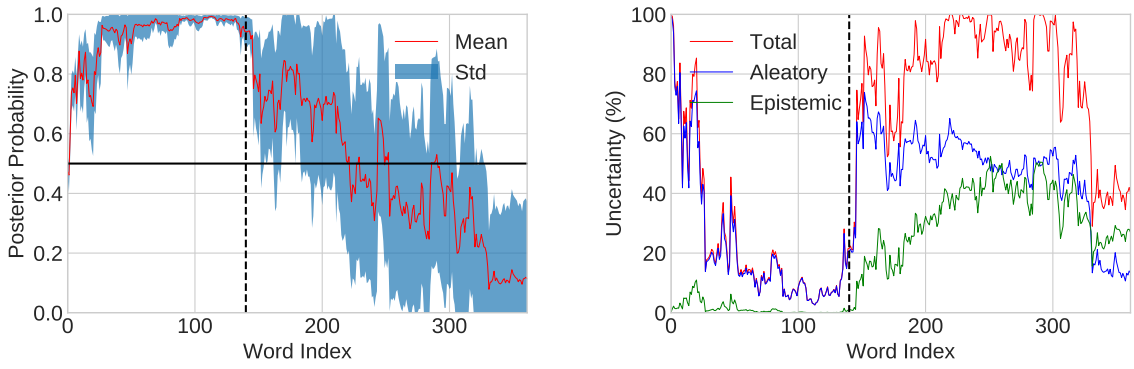


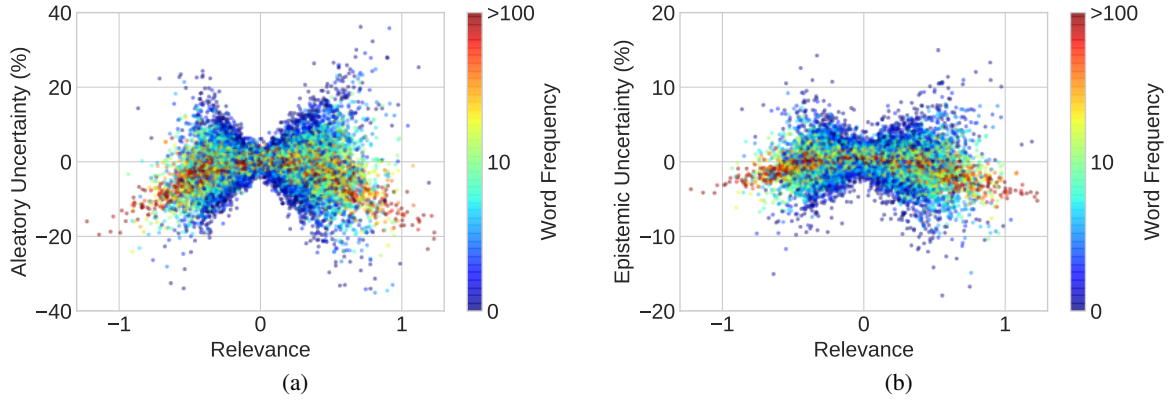
Figure 1: A negative review with 140 words concatenated with a positive review. The figure shows the effect of a changing sentiment along the input.

3.2 Word-Level Uncertainties

To check the contribution of single words to the models output, we analyse the connection between the average word relevance and uncertainty contributed by each word, as shown on Figure 2. The x-axis denotes the word relevance R_c and the y-axis refers to the aleatory and epistemic uncertainty. The plot reveals that relevant words are more likely to increase or decrease the uncertainty of the model. Furthermore, uncommon words are likely to contribute to the uncertainty, whereas most frequently used words reduce uncertainty. Comparing the figures shows a similar behaviour of the aleatory and epistemic uncertainty. It is worth to note that the model is overall less affected by epistemic uncertainty compared to aleatory uncertainty.

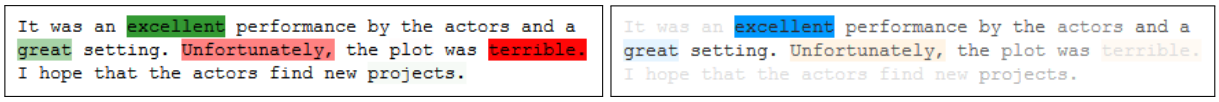
In Figure 3, we visualize word-level relevance and uncertainties obtained by our approach with a Heatmap. Figure 3a shows the word relevance, where negative sentiment words are highlighted in red and positive sentiments are highlighted in green. The example is classified positive. Figure 3b shows the total uncertainty contributed by each word. Words which reduce the uncertainty are marked in blue and

[1] https://keras.io/examples/imdb_lstm/
[2] https://github.com/jsandersen/WU-RNN



(a) (b)

Figure 2: Dependencies between word relevance and (a) aleatory as well as (b) epistemic uncertainty.



(a) (b)

Figure 3: A visualization of (a) word relevance and (b) world-level uncertainty as calculated by our approach. Red denotes negative sentiment words and green denotes positive words. Blue marks Words which reduce the uncertainty and orange marks words which add uncertainty. Low opacity indicates a high sequence uncertainty.

words which add uncertainty in orange. Further, we vary the opacity of the font to indicate the sequence uncertainty $U_t(e_i)$ at each index i. A low opacity indicates a high sequence uncertainty and vice versa. In the given example, the model is uncertain until it observes a relevant term. The term 'excellent' reduces the uncertainty. Hereby, the model becomes more confident that the example belongs to the positive sentiment class. When the contradicting term 'unfortunately' is observed in the sequence, the class probability drops, resulting in an increased uncertainty. Finally, the model remains highly uncertain in this example about the overall prediction.

4 Related Work

Uncertainty Estimation in Neural Networks. Prior work aimed at estimating predictive uncertainties in Neural Networks (Xiao and Wang, 2019; Kendall and Gal, 2017; Kwon et al., 2020; Gal and Ghahramani, 2016) and applied these to text classification tasks (Xiao and Wang, 2019; Burkhardt et al., 2018; Siddhant and Lipton, 2018). However, these techniques are generally used to only assess input-level rather than word-level uncertainties. Li et al. (2017) investigate the detection of uncertain words using Neural Networks, too. However, they learn the words' uncertainty from a labelled dataset, while our approach can be considered unsupervised as it does not require additional labelling.

Attribution of Neural Networks. The field of explainable AI (Adadi and Berrada, 2018) seeks to overcome the black-box processing of Neural Networks by providing humans interpretable information to reason about artificial decision-making. Techniques such as gradient based sensitivity analysis (Simonyan et al., 2013) or layer-wise relevance propagation (Bach et al., 2015) allows us to infer word relevance, which can also be achieved by our approach. Techniques to explain word relevance have been previously applied to text classifiers (Li et al., 2016; Arras et al., 2017). However, these techniques do not assess the uncertainty of a prediction. Du et al. (2019) follow a different approach to decompose RNNs outputs to assess feature relevance. Our approach might be further improved by their findings.

5 Conclusion and Further Work

This paper proposes a simple novel approach to estimate word-level uncertainties in text classification tasks. Our approach uses Monte Carlo Dropout in conjuncture with a sequence modelling technique to decompose uncertainties. Our approach does not require additional labelling effort beside the original training data for the classifier. We exemplary show that the transparency gained by applying our approach enables a deeper understanding of artificial decision-making. The visualization in Figure 3 can, e.g., help a human moderator to understand the classifier uncertainty. We plan several empirical studies to examine the impact and benefits of word-level uncertainty awareness in Human-in-the-Loop applications (Zanzotto, 2019). Further, we plan to adapt and compare our approach to additional RNN variations like bidirectional LSTMs and gated recurrent units (GRUs).

Acknowledgements

The paper was supported by BWFGB Hamburg within the "Forum 4.0" project as part of the ahoi.digital funding line.

References

Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. "what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8).

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7).

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831, August.

Sophie Burkhardt, Julia Siekiera, and Stefan Kramer. 2018. Semisupervised bayesian active learning for text classification. In *Bayesian Deep Learning Workshop at NeurIPS*.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112.

Mengnan Du, Ninghao Liu, Fan Yang, Shuiwang Ji, and Xia Hu. 2019. On attribution of recurrent neural network predictions via additive decomposition. In *The World Wide Web Conference*, pages 383–393.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc.

Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. 2020. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6405–6416. Curran Associates Inc.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691.

Binyang Li, Kaiming Zhou, Wei Gao, Xu Han, and Linna Zhou. 2017. Attention-based lstm-cnns for uncertainty identification on chinese social media texts. In *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pages 609–614. IEEE.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. SemEval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18, San Diego, California, June. Association for Computational Linguistics.

Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer.

Aditya Siddhant and Zachary C Lipton. 2018. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January.

Tingmin Wu, Shigang Liu, Jun Zhang, and Yang Xiang. 2017. Twitter spam detection based on deep learning. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '17, New York, NY, USA. Association for Computing Machinery.

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.

Fabio Massimo Zanzotto. 2019. Human-in-the-loop artificial intelligence. *Journal of Artificial Intelligence Research*, 64:243–252.