Efficient Out-of-Domain Detection for Sequence to Sequence Models

Artem Vazhentsev[1,2] ♢, Akim Tsvigun[6,7] ♢, Roman Vashurin[4] ♢, Sergey Petrakov[2], Daniil Vasilev[5], Maxim Panov[4], Alexander Panchenko[2,1], and Artem Shelmanov[3]
[1]AIRI, [2]Skoltech, [3]MBZUAI, [4]TII, [5]HSE University,
[6]AI Center NUST MISiS, [7]Semrush
{vazhentsev, panchenko}@airi.net {sergey.petrakov}@skoltech.ru
{roman.vashurin, maxim.panov}@tii.ae artem.shelmanov@mbzuai.ac.ae

Abstract

Sequence-to-sequence (seq2seq) models based on the Transformer architecture have become a ubiquitous tool applicable not only to classical text generation tasks such as machine translation and summarization but also to any other task where an answer can be represented in a form of a finite text fragment (e.g., question answering). However, when deploying a model in practice, we need not only high performance but also an ability to determine cases where the model is not applicable. Uncertainty estimation (UE) techniques provide a tool for identifying out-of-domain (OOD) input where the model is susceptible to errors. State-of-the-art UE methods for seq2seq models rely on computationally heavyweight and impractical deep ensembles. In this work, we perform an empirical investigation of various novel UE methods for large pre-trained seq2seq models T5 and BART on three tasks: machine translation, text summarization, and question answering. We apply computationally lightweight density-based UE methods to seq2seq models and show that they often outperform heavyweight deep ensembles on the task of OOD detection[1].

1 Introduction

Sequence-to-sequence (seq2seq) models achieve state-of-the-art performance in various NLP tasks, such as neural machine translation (NMT; Vaswani et al. (2017); Song et al. (2019); Zhu et al. (2020); Liu et al. (2020)), abstractive text summarization (ATS; Zhang et al. (2020); Lewis et al. (2020)), question answering (QA; Raffel et al. (2020)), and others. Such models may encounter various user inputs when exposed to the general public. In many cases, it is preferable to detect and handle in a special way what is known as out-of-domain (OOD) inputs. OOD instances are significantly different from the data used during training, and as a result, model predictions on such inputs might be unreliable. OOD can be performed in supervised and unsupervised ways. In a supervised approach, one trains a discriminator between in-domain (ID) and OOD instances on a labeled dataset of such instances, which is manually annotated (Hendrycks et al., 2019) or synthetically generated (Liang et al., 2018). The drawback of such an approach is that the discriminator is also limited in what instances it can correctly process. Therefore, in many practical cases, it might be better to use an unsupervised approach, where OOD instances are detected using uncertainty estimation (UE) methods.

Related work. UE for text generation models is still an area of ongoing research with only a limited number of works. Malinin and Gales (2020) propose various ensemble-based UE methods for seq2seq models and evaluate them on two tasks: NMT and automatic speech recognition. Ensemble-based methods in conjunction with Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) are also investigated in (Lukovnikov et al., 2021). The authors find that the ensemble-based UE methods lead to the best results for OOD detection in the neural semantic parsing task. Xiao et al. (2020) introduce a novel UE method BLEUVar, which is also based on MC dropout. The uncertainty score is calculated as a sum of the squared complements of BLEU scores for all pairs of generated texts obtained with different dropout masks. The method shows improvements over the baselines in NMT. Lyu et al. (2020) further explore this method for OOD detection in question answering. Gidiotis and Tsoumakas (2022) show that BLEUVar can also be applied for UE in summarization. The aforementioned methods entail performing multiple model inferences for each individual input, resulting in high computational overhead. Recently, Kuhn et al. (2022) propose a method that does not leverage MC dropout, but samples multiple predictions without

[1]The code for reproducing experiments is available online at https://github.com/stat-ml/seq2seq_ood_detection

♢ Equal contribution

Findings of the Association for Computational Linguistics: ACL 2023, pages 1430–1454
July 9-14, 2023

additional inferences. It is called semantic entropy and is based on the idea that different samples can have the same meaning. It calculates the entropy of the probability distribution over meanings instead of their surface realizations. Semantic entropy outperforms the standard predictive entropy-based methods proposed in (Malinin and Gales, 2020) on the free-form question answering task.

Contributions. In this work, we show that there is significant room for improvement for existing OOD detection methods in seq2seq tasks. We find out that in some configurations, they even work worse than the random choice. Moreover, most of them are computationally intensive, which hinders their successful application in real-world settings.

To address these issues, we adopt methods based on fitting the probability density of latent instance representations obtained from a trained neural network (Lee et al., 2018; Yoo et al., 2022). While these methods are shown to be effective for text classification tasks, their application in text generation tasks has received limited research attention. We fill this gap by conducting an empirical investigation of these methods for OOD detection in NMT, ATS, and QA tasks and show their superiority over the baselines from previous work. The main contributions of our paper are as follows.

- We perform a large-scale empirical study of UE methods on three different sequence generation tasks: NMT, ATS, and QA, with various types of out-of-domain inputs: permutations of tokens from original input, texts from a new domain, and texts from another language.
- We show that the density-based approaches are both more effective and computationally efficient than previously explored state-of-the-art ensemble-based or MC dropout-based methods. The improvement is consistently observed in all considered tasks.

2 Out-of-domain Detection Methods

OOD detection using uncertainty estimation is a binary classification task, where an uncertainty score $U(\mathbf{x})$ of a given input $\mathbf{x}$ is a predictor of $\mathbf{x}$ coming from an unknown domain. In practice, a threshold δ is specified so that all $\mathbf{x}$: $U(\mathbf{x}) > \delta$ are considered to be OOD.

The task of text generation involves complex autoregressive probabilistic models and usually requires making not one but multiple predictions (one per output token). These two factors make UE of predictions in text generation tasks much more complicated than in standard text classification tasks. Below, we provide a short overview of the approaches for uncertainty estimation of autoregressive model predictions investigated in our work. More comprehensive details can be found in Appendix A. All methods described below can be applied to the majority of modern Transformer-based pre-trained seq2seq models.

2.1 Information-based Uncertainty Estimation

Usually, seq2seq models for each input $\mathbf{x}$ can generate multiple candidate sequences $\mathbf{y}$ via beam-search, where the resulting set of sequences $\mathcal{B}(\mathbf{x}) = \{\mathbf{y}^{(b)}\}_{b=1}^{B}$ is called a "beam". To get the uncertainty score associated with a prediction on $\mathbf{x}$, we can aggregate individual uncertainties for input-output pairs $(\mathbf{x}, \mathbf{y}^{(b)})$ of the whole beam.

The simplest aggregation method is to take the probability of a sequence $\mathbf{y}^*$ that has the maximum confidence and is usually selected as a final model output. We refer to this method as *Maximum Sequence Probability (MSP)*. The alternative approach is to consider the hypotheses in the beam $\mathbf{y}^{(b)}$ as samples from a distribution of possible sequences. In this case, we can compute the expected probabilities over the beam, yielding a method called *Normalized Sequence Probability (NSP)*. Another option is to compute the average *entropy* of the predictive token distributions over the beam.

2.2 Ensembling

One can train several models for a single task and benefit from their variability to estimate the uncertainty. In this section, we mostly follow Malinin and Gales (2020) who give a comprehensive overview of the information-based UE techniques for ensembles and Bayesian methods in general.

First of all, note that hypotheses sequences that form the beam $\mathcal{B}(\mathbf{x}) = \{\mathbf{y}^{(b)}\}_{b=1}^{B}$ for the case of ensembling can be generated naturally by generating tokens sequentially according to the average of the probabilities of ensemble members. Such an ensembling approach is usually referred to as *Product of Expectations (PE)* ensemble. We consider two types of ensemble-based UE methods: sequence-level and token-level.

Sequence-level methods obtain uncertainty scores for the whole sequence at once. *Total Uncertainty (TU)* is measured via entropy and *Reverse*

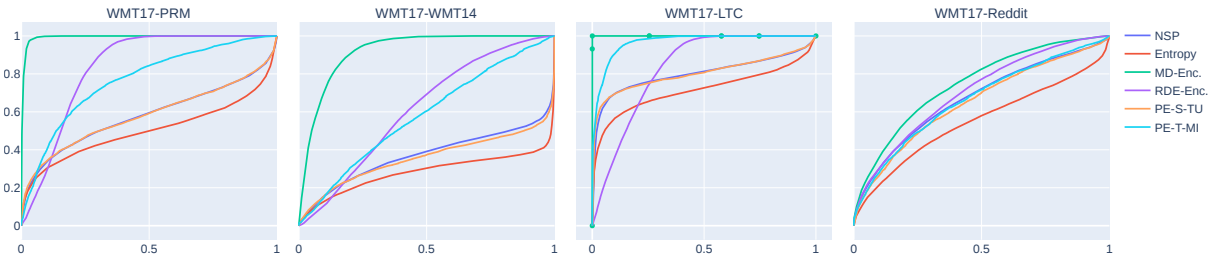

Figure 1: Average ROC curves in various configurations on the NMT task for the selected UE methods. The first dataset in the title represents the ID dataset, the second is the OOD dataset.

Mutual Information (RMI). We refer to these scores as PE-S-TU and PE-S-RMI in our experiments. One can also consider an alternative way of ensembling models that is usually called the *Expectation of Products (EP)* ensemble. It averages the probabilities of whole sequences computed by different models. This approach gives us two more variants of TU and RMI: EP-S-TU and EP-S-RMI.

In token-level UE methods, we compute some uncertainty measure for each token first and then average these scores over all tokens in a sequence. We consider *Total Uncertainty* measured via entropy, *Mutual Information (MI)*, *Expected Pairwise KL Divergence (EPKL)* and *Reverse Mutual Information (RMI)*. The resulting token-level uncertainties can be averaged via the *PE* approach leading to PE-T-TU, PE-T-MI, PE-T-EPKL, and PE-T-RMI methods. The alternative is to use *EP* averaging that gives us another four metrics to consider: EP-T-TU, EP-T-MI, EP-T-EPKL and EP-T-RMI.

2.3 Density-based Methods

Recently, density-based methods exhibited outstanding performance in UE of deep neural network predictions (Lee et al., 2018; van Amersfoort et al., 2020; Kotelevskii et al., 2022; Yoo et al., 2022). Yet, none of them has been applied to seq2seq models.

The basic idea behind density-based UE methods is to leverage the latent space of the model and fit the probability density of the training input representations within it. The lower value of the density is then considered as an indicator of a higher uncertainty due to the scarce training data used to make the prediction.

We adopt two state-of-the-art methods of this type for seq2seq models: *Mahalanobis Distance* (MD; Lee et al. (2018)) and *Robust Density Estimation* (RDE; Yoo et al. (2022)). Let $h(\mathbf{x})$ be a hidden representation of an instance $\mathbf{x}$. The MD method fits a Gaussian centered at the training data centroid μ with an empirical covariance matrix Σ. The uncertainty score is the Mahalanobis distance between $h(\mathbf{x})$ and μ:

$$U^{\text{MD}}(\mathbf{x}) = (h(\mathbf{x}) - \mu)^T \Sigma^{-1} (h(\mathbf{x}) - \mu).$$

We suggest using the last hidden state of the encoder averaged over non-padding tokens or the last hidden state of the decoder averaged over all generated tokens as $h(\mathbf{x})$. An ablation study of various embeddings extraction and reduction methods is provided in Appendix D.

The RDE method improves over MD by reducing the dimensionality of $h(\mathbf{x})$ via PCA decomposition. It also computes the covariance matrix in a robust way using the Minimum Covariance Determinant estimate (Rousseeuw, 1984). The uncertainty score $U^{\text{RDE}}(\mathbf{x})$ is also the Mahalanobis distance but in the space of reduced dimensionality.

3 Experiments

Following (Malinin and Gales, 2020), we use two approaches to generating OOD data for a given "in-domain" (ID) dataset. In the first approach, we simply take texts from another dataset, which is distinct from the training set of the model in terms of domain and/or structure. In the second approach, we corrupt the dataset by randomly permuting the source tokens (PRM). The details of OOD data creation are provided in Appendix B.

Following the previous works on OOD detection (Hendrycks and Gimpel, 2017; Malinin and Gales, 2020), we report the AU-ROC scores of detecting OOD instances mixed into the test set. To ensure stability, we run each experiment with 5 different random seeds and report the standard deviation. For brevity, in the main part, we report the results of only the two best-performing methods from each method group. Hardware configuration for experiments is provided in Appendix B.

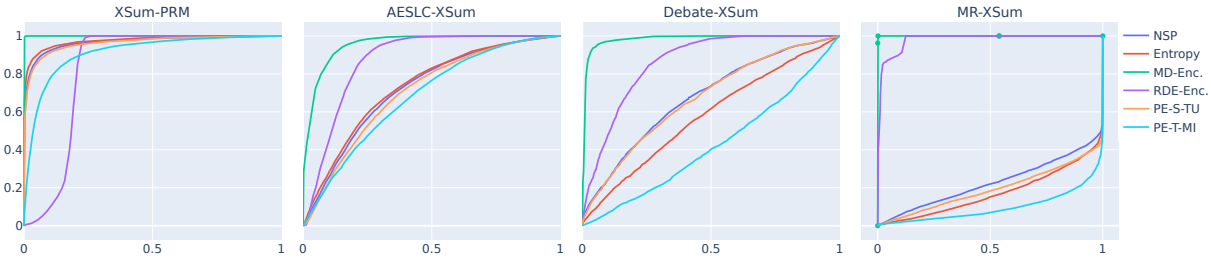

Figure 2: Average ROC curves on the ATS task for the selected UE methods when XSum is the OOD dataset. The first dataset in the title represents the ID dataset.

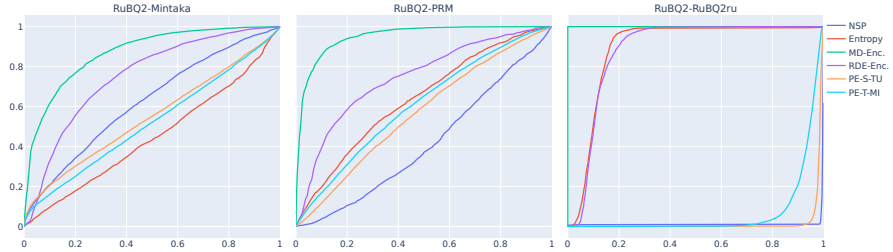

Figure 3: Average ROC curves for QA task on datasets with links to Wikidata KG. The first dataset in the title is the ID dataset, the second represents the OOD dataset. Also, the language is English except for the case with "ru", which identifies the Russian language.

3.1 Performance on ID vs OOD

First and foremost, we must ensure that the task of identifying OOD examples is indeed crucial in text generation tasks. To do so, we compare the model's performance on ID and OOD data. Tables 3, 6, 8 in Appendix C depict a comparison of the model performance on ID and OOD observations in various ID-OOD settings for the NMT, ATS, and QA tasks. We can see that the model's quality is significantly worse on OOD data in all possible settings. This underlines the necessity of identifying OOD examples in real-world applications since the model is incapable of generating adequate predictions for such observations.

3.2 Machine Translation

Experimental setup. We conduct experiments on two ID datasets: WMT'17 English-to-German (En-De; Bojar et al. (2017)) and WMT'20 English-to-Russian (En-Ru; Barrault et al. (2020)). The OOD datasets were selected according to the benchmark of Malinin and Gales (2020). Since in real-life settings, OOD data come from various sources, we want to cover as many domains of data as possible with these datasets. For OOD data generation, we use texts from WMT'14 (Bojar et al., 2014) in French, the LibriSpeech test-clean (LTC) reference texts (Panayotov et al., 2015), and English comments from Reddit from the Shifts dataset (Malinin et al., 2022). The predictions are made by the multilingual mBART model (Liu et al., 2020). The details of the datasets and the model are provided in Appendix B.

Results. The performance of the selected methods is presented in Figure 1 and Figure 4 in Appendix H. For both ID datasets with LTC and PRM being OOD datasets, MD separates ID and OOD instances very clearly. It achieves an AU-ROC score very close to the optimal one, outperforming all the ensemble-based methods.

When WMT'14 is used as OOD, for the model trained on the WMT'17, most of the ensemble-based methods notably fall behind even the random choice, which means that the model is overconfident in OOD instances. In contrast, MD and RDE yield adequate results. MD based on encoder-derived embeddings shows the best quality in this setting. In the hardest setting, where Reddit is used as an OOD dataset, MSP and ensembles poorly detect OOD instances, while the density-based methods outperform all other techniques by a large margin. The only case where density-based methods show slightly lower performance is when WMT'14 and Reddit are considered OOD for the model trained on WMT'20.

Overall, we can see that in most of the considered settings, MD substantially outperforms all other methods, and it is steadily better than the random choice baseline, while other methods are sometimes worse than the random choice. The compute time of the selected methods is presented in Table 13 in Appendix E. We see that the efficient density-based methods introduce only a small com-

putational overhead compared to ensemble-based approaches. The complete results of all the considered methods are presented in Table 15 in Appendix H.

Finally, the qualitative analysis of model performance and examples of ID/OOD predictions are presented in Tables 4,5 in Appendix C.

3.3 Abstractive Text Summarization

Experimental setup. We experiment with four widely used datasets for ATS with each being ID and OOD: XSum (Narayan et al., 2018), AESLC (Zhang and Tetreault, 2019), Movie Reviews (MR; Wang and Ling (2016)), and Debate (Wang and Ling, 2016). Predictions are made by the standard BART model (Lewis et al., 2020). The details on the datasets and the model are provided in Appendix B.

Results. For brevity, in the main part of the paper, we only keep the results with XSum being an OOD dataset. The results for other settings are presented in Appendix G. Figure 2 and Figure 5, Tables 16 and 17 in Appendix G illustrate the results of OOD detection in different corruption scenarios.

First, we can clearly see that the density-based methods relying on both encoder and decoder features provide a large improvement over both information-based and ensemble-based methods. In each corruption scenario, at least one of the MD versions yields the highest AU-ROC scores.

Second, we can observe that some OOD configurations where density-based methods achieve the optimal quality (e.g. MR-XSum, MR-Debate) turn out to be challenging for both information-based and ensemble-based methods. These methods perform worse than the random choice baseline.

Third, when XSum is the ID dataset, RDE based on encoder features fails to perform well. MD, however, achieves the best results in these cases.

Finally, the ensemble-based methods struggle to work stable across different settings. We can see that both PE-S-TU and PE-T-MI are even inferior to information-based methods in some ID-OOD dataset configurations (e.g. AESLC-XSum, Debate-XSum). MD, on the contrary, shows robust results without performance gaps.

3.4 Question Answering

Experimental setup. For the QA task, we select several widely-used KGQA datasets: Simple Questions (Bordes et al., 2015), Mintaka (Sen et al., 2022), and RuBQ 2.0 (Rybin et al., 2021). For predictions, we use the T5 model pre-trained for the QA task (Roberts et al., 2020). The details on the datasets and the model are given in Appendix B. The T5 model is used in zero-shot and if no sampling technique is undertaken, there will be no diversity for single model-based and density-based methods. Thus, we apply the bootstrap technique to estimate the confidence of the results obtained by calculating the standard deviation from the mean results.

Results. Experiments on the QA task demonstrate similar behavior of UE methods. From Figure 3 and Table 18 in Appendix H, we can see that the density-based estimates obtained from encoder-derived embeddings outperform all the other uncertainty methods by a large margin.

They achieve high-quality results even in cases when the ensemble-based methods completely miss the target (e.g. RuBQ2-RuBQ2ru). This confusion can be explained by the fact that in the case when the model receives input data that is significantly different from what it was trained on, for example, the pre-training was mostly in English, and the question in Russian, the network is forced into default mode distribution based on the frequency of tokens. Example of such generation mode is illustrated in Table 7 in Appendix H.

For experiments in settings RuBQ2-Mintaka and RuBQ2-PRM, we do not observe such a significant outlier as in the previous example. MD is the obvious leader, followed by RDE with a significant gap. Additional qualitative analysis in Table 7 in Appendix H shows that for a particular OOD example, often the uncertainty metric based on a single model and MC ensemble is not so different from the ID counterpart which explains their poor performance.

4 Conclusion

We adopted the density-based UE methods for seq2seq models and demonstrated that they provide the best results in OOD detection across three sequence generation tasks: NMT, ATS, and QA. They appear to be superior to the ensemble-based methods in terms of both performance and compute time, which makes them a good choice for applying in practice.

In future work, we are going to extend the application of density-based methods to seq2seq models in other UE tasks such as selective classification.

Acknowledgements

The work of Akim Tsvigun was prepared in the framework of the strategic project “Digital Business” within the Strategic Academic Leadership Program "Priority 2030" at NUST MISiS. This work was also supported in part by computational resources of the HPC facilities at the HSE University (Kostenetskiy et al., 2021).

Limitations

In our experiment, we presented results for three diverse sequence-to-sequence tasks, namely, machine translation, text summarization and knowledge graph question answering. While for these three tasks, we managed to observe common trends (i.e. some methods consistently outperformed other methods) a more large-scale study of various sequence-to-sequence tasks is needed to further confirm this observation and robustness of the best-performing method as identified in this work.

Ethics Statement

Uncertainty estimation methods are useful for building safer and more robust machine learning models. However, the extent to which they may interfere with other model tailoring methods, such as debiasing or compression models is not currently studied. In principle, we do not see large ethical implications in our research or risks.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Alexios Gidiotis and Grigorios Tsoumakas. 2022. Should we trust this summary? bayesian abstractive summarization to the rescue. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4119–4131. Association for Computational Linguistics.

Dan Hendrycks, Steven Basart, Mantas Mazeika, and Duncan Wilson. 2019. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

P. S. Kostenetskiy, R. A. Chulkevich, and V. I. Kozyrev. 2021. HPC Resources of the Higher School of Economics. *Journal of Physics: Conference Series*, 1740(1):012050.

Nikita Yurevich Kotelevskii, Aleksandr Artemenkov, Kirill Fedyanin, Fedor Noskov, Alexander Fishkov, Artem Shelmanov, Artem Vazhentsev, Aleksandr Petiushko, and Maxim Panov. 2022. Nonparametric uncertainty quantification for single deterministic neural network. In *Advances in Neural Information Processing Systems*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *NeurIPS ML Safety Workshop*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*

31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, volume 31, pages 7167–7177.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Shiyu Liang, Yixuan Li, and R Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Denis Lukovnikov, Sina Däubener, and Asja Fischer. 2021. Detecting compositionally out-of-distribution examples in semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 591–598. Association for Computational Linguistics.

Zhihao Lyu, Danier Duolikun, Bowei Dai, Yuan Yao, Pasquale Minervini, Tim Z Xiao, and Yarin Gal. 2020. You need only uncertain answers: Data efficient multilingual question answering. *Workshop on Uncertainty and Robustness in Deep Learning*.

Andrey Malinin, Andreas Athanasopoulos, Muhamed Barakovic, Meritxell Bach Cuadra, Mark J. F. Gales, Cristina Granziera, Mara Graziani, Nikolay Kartashev, Konstantinos Kyriakopoulos, Po-Jui Lu, Nataliia Molchanova, Antonis Nikitakis, Vatsal Raina, Francesco La Rosa, Eli Sivena, Vasileios Tsarsitalidis, Efi Tsompopoulou, and Elena Volf. 2022. Shifts 2.0: Extending the dataset of real distributional shifts.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5206–5210. IEEE.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Peter J Rousseeuw. 1984. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880.

Ivan Rybin, Vladislav Korablinov, Pavel Efimov, and Pavel Braslavski. 2021. Rubq 2.0: an innovated russian question answering dataset. In *European Semantic Web Conference*, pages 532–547. Springer.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Comput. Linguistics*, 33(1):9–40.

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. Uncertainty estimation using a single deep deterministic neural network. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9690–9700. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 47–57. The Association for Computational Linguistics.

Tim Z. Xiao, Aidan N. Gomez, and Yarin Gal. 2020. Wat zei je? detecting out-of-distribution translations with variational transformers. *CoRR*, abs/2006.08344.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3656–3672, Dublin, Ireland. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Rui Zhang and Joel R. Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 446–456. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.

A Methods

A.1 Base Probabilistic Uncertainty Measures

The task of sequence generation involves relatively complex autoregressive probabilistic models and there exist several variants of defining uncertainties for them. Let us consider the input sequence $\mathbf{x}$ and the output sequence $\mathbf{y} \in \mathcal{Y}$ of the length L, where $\mathcal{Y}$ is a set of all possible output sequences. Then the standard autoregressive model parametrized by $\boldsymbol{\theta}$ is given by:

$$P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \prod_{l=1}^{L} P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}), \tag{1}$$

where the distribution of each y_l is conditioned on all the previous tokens in a sequence $\mathbf{y}_{<l} = \{y_1, \ldots, y_{l-1}\}$.

The probability $P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$ immediately gives a so-called **Unnormalized Sequence Probability (USP)** uncertainty metric: $\mathrm{USP}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = 1 - P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$. However, this metric tends to increase with the increase of the sequence length L which is usually undesirable in practice. That is why some alternatives are proposed.

Normalized Sequence Probability (NSP; Ueffing and Ney (2007)) metric directly deals with the variable length via the appropriate normalization that corresponds to average token log-probability $\bar{P}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \exp\Big\{\frac{1}{L} \log P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})\Big\}$:

$$\mathrm{NSP}(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = 1 - \bar{P}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}). \tag{2}$$

Average token-wise entropy (Malinin and Gales, 2020) allows to generalize the notion of standard entropy-based uncertainty metrics for the case of autoregressive models:

$$\mathcal{H}(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{L} \sum_{l=1}^{L} \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}), \tag{3}$$

where $\mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})$ is an entropy of the token distribution $P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})$.

A.2 Aggregation of Uncertainties over Beam

In practice, seq2seq models for each input $\mathbf{x}$ usually generate several candidate sequences via beam-search procedure. The resulting set $\mathcal{B}(\mathbf{x}) = \{\mathbf{y}^{(b)}\}_{b=1}^{B}$ is usually called beam. Thus, for the solution of OOD detection problems, one needs to aggregate uncertainties of particular pairs $(\mathbf{x}, \mathbf{y}^{(b)})$ into one uncertainty measure associated with an input $\mathbf{x}$.

The simplest method to measure the uncertainty for a beam of sequences is to take the sequence having maximum confidence as exactly this sequence is usually selected as a resulting output of the model. In this work, we consider the particular instantiation of this approach based on NSP measure (2) that we call **Maximum Sequence Probability (MSP)**:

$$\mathrm{MSP}(\mathbf{x}; \boldsymbol{\theta}) = 1 - \max_{b \in \overline{1,B}} \bar{P}(\mathbf{y}^{(b)} \mid \mathbf{x}, \boldsymbol{\theta}). \tag{4}$$

The alternative approach is to consider the hypotheses sequences $\mathbf{y}^{(b)}$ as samples from a distribution of sequences $\bar{P}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$. Each sequence is seen only once and to correctly compute the expectation of some uncertainty measure $U(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta})$ over this distribution one needs to perform some corrections. The natural choice is importance weighting that leads to the following uncertainty estimate:

$$\mathrm{U}(\mathbf{x}; \boldsymbol{\theta}) = \sum_{b=1}^{B} \mathrm{U}(\mathbf{y}^{(b)}, \mathbf{x}; \boldsymbol{\theta}) \frac{\bar{P}(\mathbf{y}^{(b)} \mid \mathbf{x}, \boldsymbol{\theta})}{\sum_{j=1}^{B} \bar{P}(\mathbf{y}^{(j)} \mid \mathbf{x}, \boldsymbol{\theta})}.$$

Thus, we got an averaged versions of **NSP** (2):

$$\text{NSP}(\mathbf{x};\boldsymbol{\theta}) = 1 - \sum_{b=1}^{B} \frac{\bar{P}(\mathbf{y}^{(b)} \mid \mathbf{x},\boldsymbol{\theta})^2}{\sum_{j=1}^{B} \bar{P}(\mathbf{y}^{(j)} \mid \mathbf{x},\boldsymbol{\theta})}$$

and **entropy** (3):

$$\mathcal{H}(\mathbf{x};\boldsymbol{\theta}) = \sum_{b=1}^{B} \mathcal{H}(\mathbf{y},\mathbf{x};\boldsymbol{\theta}) \frac{\bar{P}(\mathbf{y}^{(b)} \mid \mathbf{x},\boldsymbol{\theta})}{\sum_{j=1}^{B} \bar{P}(\mathbf{y}^{(j)} \mid \mathbf{x},\boldsymbol{\theta})}.$$

A.3 Ensembling

The uncertainty metrics in previous sections are applicable to a single model, while in many applications one can train several models for a single task and benefit from their variability. We assume that an ensemble of M models has been trained with resulting parameters $\boldsymbol{\theta}_i, i = 1, \ldots, M$. In what follows, we discuss the variety of uncertainty measures that can be computed based on the ensemble of models.

A.3.1 Beam Generation

First of all, we need to discuss how to generate hypotheses sequences for the case of ensembling. We follow the most natural way by generating tokens sequentially according to the average of the probabilities of ensemble members:

$$y_l \sim P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}), \tag{5}$$

where for $l = 1, \ldots, L$ we defined

$$P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}; \boldsymbol{\theta}_i). \tag{6}$$

Such an ensembling approach is usually referred as *Product of Expectations (PE)* ensemble.

In what follows, we assume that the beam $\mathcal{B}(\mathbf{x}) = \{\mathbf{y}^{(b)}\}_{b=1}^{B}$ is generated via PE ensemble. The corresponding importance weights are given by

$$\pi^{(b)} = \frac{\bar{P}(\mathbf{y}^{(b)} \mid \mathbf{x})}{\sum_{j=1}^{B} \bar{P}(\mathbf{y}^{(j)} \mid \mathbf{x})},$$

where $\bar{P}(\mathbf{y}^{(b)} \mid \mathbf{x}) = \exp\Big\{\frac{1}{L^{(b)}} \log P(\mathbf{y}^{(b)} \mid \mathbf{x})\Big\}$ with L being the length of the sequence $\mathbf{y}^{(b)}$ and $P(\mathbf{y}^{(b)} \mid \mathbf{x}) = \prod_{l=1}^{L^{(b)}} P(y_l^{(b)} \mid \mathbf{y}_{<l}^{(b)}, \mathbf{x})$.

A.3.2 Sequence Level Ensembling

For the ensembling on a sequence level, we consider two uncertainty measures: total uncertainty (TU) measured via entropy

$$\mathcal{H}_S(\mathbf{x}) = \sum_{b=1}^{B} \pi^{(b)} \log \bar{P}(\mathbf{y}^{(b)} \mid \mathbf{x}) \tag{7}$$

and

$$\mathcal{M}_S(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{b=1}^{B} \frac{\pi^{(b)}}{L^{(b)}} \log \frac{P(\mathbf{y}^{(b)} \mid \mathbf{x})}{P(\mathbf{y}^{(b)} \mid \mathbf{x}, \boldsymbol{\theta}_i)}, \tag{8}$$

which is known as reverse mutual information (RMI). We refer to these measures as PE-S-TU and PE-S-RMI in our experiments. We note that one can also consider an alternative way of ensembling models that is usually called *Expectation of Products (EP)* ensemble:

$$\breve{P}(\mathbf{y} \mid \mathbf{x}) = \exp\Big\{\frac{1}{L} \log \frac{1}{M} \sum\nolimits_{i=1}^{M} P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_i)\Big\},$$

and compute TU and RMI by substituting $\bar{P}(\mathbf{y} \mid \mathbf{x})$ with $\breve{P}(\mathbf{y} \mid \mathbf{x})$ in equations (7) and (8) respectively. We refer to these methods as EP-S-TU and EP-S-RMI in our experiments.

A.3.3 Token Level Ensembling

In the previous section, all the computation of uncertainties was performed on the level of the full sequences. However, multiple opportunities exist to perform it on the level of individual tokens and then aggregate the resulting token uncertainties over the whole sequence. Below we discuss this in detail.

We start from a total uncertainty estimate via entropy:

$$\mathcal{H}_T(\mathbf{x}) = \sum_{b=1}^{B} \frac{\pi^{(b)}}{L^{(b)}} \sum_{l=1}^{L^{(b)}} \mathcal{H}(y_l \mid \mathbf{y}_{<l}^{(b)}, \mathbf{x}), \tag{9}$$

where $\mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x})$ is an entropy of the token distribution $P(y_l \mid \mathbf{y}_{<l}, \mathbf{x})$ given in (6).

Additionally, for the ensemble one can compute the variety of other token level uncertainty measures including *Mutual Information (MI)*:

$$\mathcal{M}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \mathcal{H}(y_l \mid \mathbf{y}_{<l}^{(b)}, \mathbf{x}) - \frac{1}{M} \sum_{i=1}^{M} \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_i) \tag{10}$$

and *Expected Pairwise KL Divergence (EPKL)*:

$$\mathcal{K}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \binom{M}{2}^{-1} \sum_{i \neq j} \mathcal{KL}\big(P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_i) \parallel P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_j)\big),$$

where $\mathcal{KL}(P \parallel Q)$ refers to a KL-divergence between distributions P and Q.

Finally, *Reverse Mutual Information (RMI)* also can be computed on the token level via a simple equation

$$\mathcal{M}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \mathcal{K}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) - \mathcal{I}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}). \tag{11}$$

The resulting token-level uncertainties computed via MI (10), EPKL (11) and RMI (11) can be plugged-in in equation (9) on the place of entropy leading to corresponding sequence level uncertainty estimates. We refer to the resulting methods as PE-T-TU, PE-T-MI, PE-T-EPKL and PE-T-RMI.

Additionally, instead of considering the distribution $P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_i)$ one might consider the expectation of products averaging leading to the distribution:

$$\breve{P}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \frac{\sum_{i=1}^{M} P(y_l, \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_i)}{\sum_{j=1}^{M} P(\mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_i)}. \tag{12}$$

This gives us another four metrics to consider: EP-T-TU, EP-T-MI, EP-T-EPKL and EP-T-RMI.

B Experimental Details

B.1 OOD Dataset Creation

In both corruption scenarios, we use test samples of the ID and OOD datasets. From the ID dataset, all the observations are used. If the number of texts in the test sample of the OOD dataset is less than that of the ID dataset, we add observations from the training and validation sets until the number of OOD instances equals the number of ID ones. Note that we do not clip the ID dataset if the OOD dataset still contains fewer observations.

B.2 Datasets Description

B.2.1 Machine Translation

We select the WMT'14 dataset (Bojar et al., 2014), LTC (Panayotov et al., 2015), and Comments from Reddit (Malinin et al., 2022) for the following reasons. WMT'14 is different from the source datasets (WMT'17 En-De and WMT'20 En-Ru) in terms of the source language. The scenario when OOD data comes from different languages can be practical because one usually does not control the input data given by users, while the model output given the input in a different language might be unpredictable and cause reputational risks. In the next two settings, OOD texts only differ from ID in their formality level. Thus, LTC represents a new domain for the model with a completely different structure of texts as a spoken language. Comments from Reddit also refer to spoken language, embodying a structural shift in the data.

B.2.2 Abstractive Summarization

We select the following datasets since they all represent different domains. XSum (Narayan et al., 2018) consists of BBC news with their one-sentence introductory sentences as summaries. AESLC (Zhang and Tetreault, 2019) contains emails with their headlines as summaries. Movie Reviews dataset (Wang and Ling, 2016) (MR) is a collection of critics' opinions on the movie and their consensus. Finally, the Debate dataset (Wang and Ling, 2016) contains arguments and the debate topic pairs, with the former standing for documents and the latter embodying summaries.

B.2.3 Question Answering

Mintaka (Sen et al., 2022), as stated in the original article, is a complex, natural, and multilingual dataset designed for experimenting with end-to-end question-answering models. The advantage of this dataset is that it is large enough and has a decent quality of data at the same time. The trade-off between size and quality is the problem of such datasets as mentioned in (Sen et al., 2022). Besides, it provides professional translation of the questions in 8 languages and Wikidata knowledge graph IDs to cope with disambiguation.

The second dataset that we use is RuBQ 2.0 (Rybin et al., 2021). It contains Russian questions, coupled with English machine translations, SPARQL queries and answers with Russian labels, and a subset of Wikidata knowledge graph identifiers. Different complexity of questions allows us to work with data that does not have a shift towards simple or complex questions.

We also conduct experiments on the most popular and oldest Simple Questions (Bordes et al., 2015) dataset for KGQA that contains various questions. We select only the answerable ones.

Thus, we work on the task of answering questions over datasets with links to the Wikidata Knowledge Graph.

B.2.4 Dataset Statistics

We give the summary statistics about the considered datasets in Table 1.

B.3 Models

B.3.1 Machine Translation

We use the "large-CC25" version of mBART. We train an ensemble of 5 models with different random seeds for En-De and En-Ru tasks. As for the training settings, we follow the original setup and

https://huggingface.co/facebook/mbart-large-cc25

hyperparameters from (Liu et al., 2020) and train models with 100K update steps.

B.3.2 Abstractive Summarization

In this experiment, we use the "bart-base" version of BART. For each dataset, we construct 5 ensembles each consisting of 5, with a total of 25 trained models. We leverage the hyperparameters and training setup proposed in the original paper (Lewis et al., 2020).

B.3.3 Question Answering

We use the checkpoint "t5-small-ssm-nq" of the T5 model (Raffel et al., 2020) . It is considered to be a state-of-the-art model for the QA task even in closed book setting.

Table 1: Dataset statistics. We provide a number of instances for the training / validation / **test** sets, average lengths of texts and targets (answers / translations / summaries) in terms of tokens, and source / target languages.

Dataset	Num. instances	Av. document len.	Av. target len.	Language
		NMT		
WMT'20	62M / 1997 / **3000**	23.9 / 25.1	20.9 / 24.1	English-to-Russian
WMT'17	5.9M / 3000 / **3003**	26.2 / 27.0	24.8 / 28.2	English-to-German
WMT'14	40.8M / 3000 / **3003**	29.2 / 27.0	33.5 / 32.1	English-to-French
Shifts Reddit	- / 1362 / **3063**	- / 16.1	- / 16.4	English
LibriSpeech	28539 / 2703 / **2620**	-	33.4 / 20.1	English
		ATS		
XSum	204045 / 11332 / **11334**	454.6	26.1	English
Movie Reviews	2685 / 299 / **747**	972.9	28.5	English
AESLC	14436 / 1960 / **1906**	165.5	6.7	English
Debate	1626 / 181 / **452**	216.7	13.0	English
		KGQA		
RuBQ 2.0	- / 580 / **2330**	36.46 / 13.85	12.66 / 3.07	Russian / English
Mintaka	14000 / 2000 / **4000**	14.36	3.97	English
Simple Questions	19481 / 2821 / **5622**	12.56	4.01	English

B.4 Hardware & Resources

NMT and KGQA experiments were performed using the following hardware: Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz, 36 cores CPU, NVIDIA Tesla v100 GPU, 16 Gb of VRAM.

ATS experiments were performed using the following hardware: 2 Intel Xeon Platinum 8168, 2.7 GHz, 24 cores CPU; NVIDIA Tesla v100 GPU, 32 Gb of VRAM.

We provide the information about the resources employed for each experiment in Table 2.

https://huggingface.co/facebook/bart-base
https://huggingface.co/google/t5-small-ssm-nq

Table 2: Models used for experiments with their parameter counts and approximate GPU hours used for inference and training

ID Dataset / Experiment	Model	Num. Params	Avg. GPU hours
NMT			
WMT'20	mBART-large	611M	1592
WMT'17			1392
ATS			
AESLC	BART-base	139M	30
Debate			15
MR			30
XSum			150
KGQA			
Simple Questions PRM	T5-small	77M	80
RuBQ2.0 PRM			40
RuBQ2.0 En vs Ru			40
RuBQ2.0 vs Mintaka			60

C Qualitative Analysis

C.1 Machine Translation

Table 3 presents the BLEU score for the NMT task on ID and OOD datasets. We can see a significant decrease in model performance on the OOD dataset. These results demonstrate the necessity of the detection of OOD instances for maintaining the high quality of the model performance.

Dataset	WMT20 En-Ru		WMT17 En-De	
	PRM	Reddit	PRM	Reddit
ID	30.98±0.06		30.85±0.06	
OOD	6.85±0.15	24.44±0.20	8.63±0.06	11.04±0.02

Table 3: Model performance for various ID/OOD settings on the NMT task. The first row demonstrates the BLEU↑ score on the ID test dataset for the considered models. The second row demonstrates the BLEU↑ score on the OOD test dataset, presented in the header of the table.

Tables 4 and 5 present the textual examples for the models trained on the WMT17 En-De and the WMT20 En-Ru task for ID and OOD datasets. We can see, that for the PRM and WMT14 Fr as OOD, a model trained on the WMT17 En-De performs copying of the input to the output with a high probability. Therefore, the MSP uncertainty is quite low for these examples. However, MD-Encoder is able to correctly spot these instances with high uncertainty.

We can see, that for instances from the LTC dataset, both models produce poor translations, and MD-Encoder precisely detects these instances with high uncertainty. The Reddit dataset consists of challenging texts, and a model trained on the WMT20 En-Ru generates translation with a low BLEU score. However, MD-Encoder produces higher uncertainty than MSP for these examples, and we are able to correctly detect these erroneous instances.

Dataset	Input	Output	MSP	MD-Enc.	BLEU
WMT17	So what?	Was also?	0.99	0.39	37.99
WMT17	Well-known platforms include Twitch and YouTube Gaming.	Zu den bekannten Plattformen gehören Twitch und YouTube Gaming.	0.03	0.47	47.11
WMT14 Fr	"Son côté humain est ressorti", raconte-t-il.	"Son côté humain est ressorti", raconte-t-il.	0.05	0.97	13.84
WMT14 Fr	Du chant classique pour adolescents	Klassisches Gesang für Jugendliche	0.35	0.99	0.0
PRM	Young The planning " musical association and Theatreima project"N a now. ares'	Jung Die Planung "musikalische Vereinigung und Theatreima Projekt"N a jetzt.	0.4	0.68	9.97
PRM	just timees did in Hind They so as.d emerge	just timees did in Hind They so as.d emerge	0.17	0.94	0.0
LTC	AT ANOTHER TIME HARALD ASKED	ZUR ANDEREN ZEIT HARALD	0.18	0.99	-
LTC	NO ITS NOT TOO SOON	NOCH NICHT VORher	0.27	0.99	-
LTC	YOU DON'T SEEM TO REALIZE THE POSITION	DIE POSITION IST NICHT ERWEITERT	0.45	0.98	-

Table 4: Textual examples with the input and output of the model trained on the WMT17 En-De task. We demonstrate uncertainty estimates from MSP and MD-Encoder and BLEU scores for the NMT task. For LTC, we do not show the BLEU score since ground-truth translation is not presented in the dataset. Uncertainty for each method is presented in the range [0-1]. The less saturated color indicates lower uncertainty.

Dataset	Input	Output	MSP	MD-Enc.	BLEU
WMT20	All the species of Taxus are known to produce Taxol	Известно, что все виды Taxus производят Taxol	0.92	0.8	39.76
WMT20	The Abe government's school closure plan was immediately criticized.	План правительства Абэ о закрытии школы был подвергнут резкой критике.	0.71	0.55	24.7
Reddit	"r slur" Lmfaooo imagine being that soft	Лмфаооо представить, чтобы быть мягким	0.64	0.89	5.43
Reddit	These guys are so tough they can be considered as a boss fight	Эти ребята настолько жесткие, что их можно считать боссом боя	0.05	0.69	18.58
PRM	two These days next gain are will. the we over skills the	два В эти дни следующий выигрыш есть воля.	0.22	0.92	9.79
PRM	all? at, accurate not worst And at	все на, точно не худшее И на	0.37	0.97	0.0
LTC	HE HAD BROKEN INTO HER COURTYARD	Он перебрался в ее тюрьму	0.28	0.84	-
LTC	HOW KIND MAN IS AFTER ALL	ЧТО ТАКОЕ ЧЕЛОВЕКА НА ВОЗМОЖНЫХ	0.2	0.98	-
LTC	YOU DON'T SEEM TO REALIZE THE POSITION	НЕ ПОНИМАЮТ, ЧТО ВЫ ОСУЩЕСТВЛЯете ВОЗМОЖНОСТЬ ОСУЩЕСТВЛЕНИЯ ВОЗМОЖНОСТИ	0.21	0.9	-

Table 5: Textual examples with the input and output of the model trained on the WMT20 En-Ru task. We demonstrate uncertainty estimates from MSP and MD-Encoder and BLEU scores for the NMT task. For LTC, we do not show the BLEU score, since ground-truth translation is not presented in the dataset. Uncertainty for each method is presented in the range [0-1]. The less saturated color indicates lower uncertainty.

C.2 Abstractive Summarization

Table 6 illustrates the ROUGE-2 score for the ATS task on ID and OOD datasets. Similar to NMT, the model performs much very poorly on OOD data. Therefore, detection of OOD instances is crucial for

maintaining the high quality of the model performance.

ID Dataset	AESLC				Debate				M.R.				XSum			
OOD Dataset	**PRM**	**Deb.**	**M.R.**	**XSum**	**AESLC**	**PRM**	**M.R.**	**XSum**	**AESLC**	**Deb.**	**PRM**	**XSum**	**AESLC**	**Deb.**	**M.R.**	**PRM**
ID ROUGE-2	0.220				0.184				0.113				0.197			
OOD Rouge-2	0.015	0.047	0.068	0.029	0.067	0.038	0.07	0.034	0.07	0.058	0.025	0.043	0.020	0.042	0.061	0.052

Table 6: Model performance for various ID/OOD settings on the ATS task. The first row demonstrates the ROUGE-2↑ score on the ID test dataset for the considered models. The second row demonstrates the ROUGE-2↑ score on the OOD test dataset, presented in the header of the table.

C.3 Question Answering

For the KGQA task, we also analyze the behaviour of the uncertainty metrics to further illustrate the effectiveness of density based approaches on particular examples. Table 7 depicts this analysis. It is evident that values of MD-Encoder estimates show clear difference between ID and OOD inputs. We can also clearly see that for most of the OOD inputs considered, output of the model is either factually incorrect or simply incomprehensible.

We also report model quality on the ID/OOD datasets, further justifying this choice of datasets. Results are present in Table 8. For this analysis we have chosen a larger versions of the same model – t5-large-ssm-nq. It's clear that the model performs significantly better on both ID datasets, which motivates the need to detect OOD inputs with lower expected quality of the output.

ID/OOD	Question (Input)	Answer (Output)	MD-Enc.	RDE-Enc.	NSP	Entropy	PE-S-TU	PE-T-MI
ID	What is the name of the capital of Romania?	Bucharest	0.02	0.36	0.1	0.77	0.67	0.24
ID	What country owns the island of Tahiti?	France	0.05	0.22	0.13	0.81	0.78	0.3
OOD (different language)	Как называется столица Румынии?	естар умни	0.94	0.75	0.03	0.89	0.2	0.09
OOD (different language)	Какой стране принадлежит остров Таити?	лександр едеране	0.82	0.54	0.03	0.91	0.18	0.14
OOD (permutation)	of name of capital the the? is Romania	Chisinău	0.62	0.33	0.06	0.74	0.7	0.24
OOD (permutation)	thehit island? owni Tas country of	Lausanne, New Hampshire	0.51	0.33	0.04	0.8	0.74	0.26
OOD (different domain)	How many children did Donald Trump have?	132,656	0.62	0.33	0.06	0.74	0.7	0.24
OOD (different domain)	Who performed at the Super Bowl XXIII halftime show?	Whoopi Goldberg	0.51	0.33	0.04	0.8	0.74	0.26

Table 7: Textual examples with the input and output of the model T5 (t5-small-ssm-nq) used in zero shot. We demonstrate uncertainty estimates for several illustrative examples for MD and RDE calculated on encoder embeddings, NSP, Entropy, PE-S-TU and PE-T-MI. The results presented in the table are standardised to the interval from 0 to 1 for the analysis of comparative values. The less saturated color indicates lower uncertainty.

ID Dataset	RuBQ En			WDSQ En		
OOD Dataset	**PRM**	**Ru**	**Mintaka**	**PRM**	**Ru**	**Mintaka**
ID Top-1 Acc	0.170			0.159		
OOD Top-1 Acc	0.053	0.0	0.116	0.070	0.011	0.107

Table 8: Model performance for various ID/OOD settings on the KGQA task. The first row demonstrates the Top-1 Accuracy on the ID test dataset for the considered models. The second row demonstrates the Top-1 Accuracy on OOD dataset, presented in the header of the table.

D Ablation Study of Various Embeddings Extraction and Reduction Methods

Tables 9 and 10 show ROC-AUC for MD-Encoder and MD-Decoder correspondingly with various embedding reduction methods for OOD detection for selected settings for the NMT task. The function in the reduction method column means a method for aggregation embeddings for tokens in sequence vector representation. The embedding layers column means a layer from which we extract embeddings. For embeddings from all layers, we first average them across all layers and then apply the reduction method.

The results show that the base method (mean+last layer) is the most stable embedding reduction method for OOD detection. For WMT20 as ID and Reddit as OOD, embeddings from the encoder from all layers are slightly better than from the last. However, in a setting with LTC as OOD, embeddings from the encoder from all layers significantly deteriorate OOD detection performance. For WMT17 as ID and LTC as OOD, embeddings from the decoder with maximum as the reduction method are slightly better than the mean embeddings. On the other hand, in a setting with Reddit as OOD and WMT17 or WMT20 as ID, embeddings from the decoder aggregated with maximum function significantly worsen OOD detection performance.

UE Method	Reduction Method	Embedding Layers	WMT20 En-Ru		WMT17 En-De	
			LTC	Reddit	LTC	Reddit
MD-Enc.	Mean	Last	0.86±0.01	0.72±0.0	**1.0±0.0**	0.75±0.0
MD-Enc.	Max	Last	0.63±0.03	0.56±0.01	**1.0±0.0**	0.67±0.0
MD-Enc.	Mean	All	0.78±0.01	0.75±0.0	**1.0±0.0**	0.73±0.0
MD-Enc.	Max	All	0.13±0.02	0.4±0.0	0.98±0.0	0.54±0.0

Table 9: ROC-AUC↑ for the various settings for MT task for MD-Encoder method with various reduction methods. The first row indicates for the standard embeddings extraction and reduction methods, which is used in all other tables and figures.

UE Method	Reduction Method	Embedding Layers	WMT20 En-Ru		WMT17 En-De	
			LTC	Reddit	LTC	Reddit
MD-Dec.	Mean	Last	0.77±0.01	**0.6±0.01**	0.94±0.0	**0.65±0.00**
MD-Dec.	Max	Last	**0.83±0.01**	0.38±0.01	**0.99±0.0**	0.57±0.0
MD-Dec.	Mean	All	0.65±0.04	0.57±0.03	0.88±0.01	0.58±0.01
MD-Dec.	Max	All	0.45±0.04	0.37±0.03	0.97±0.0	0.59±0.01

Table 10: ROC-AUC↑ for the various settings for MT task for MD-Decoder method with various reduction methods. The first row indicates for the standard embeddings extraction and reduction methods, which is used in all other tables and figures.

We additionally carried out an analysis for the KBQA task and the results are shown in the following Tables 11 and 12. WE can see that it is the averaging over last encoder hidden state gives the best results. We have compared such approaches as averaging over all latent states as well as taking the maximum from all hidden states as well as from the last one.

Moreover, we compute it both for encoder and decoder part and show that it is reasonably to focus specifically on the encoder's hidden states. Also, the calculation of the standard deviation of the estimate from the average ROC AOC allows us to trust the results, as there is no overlap between different standard deviations.

UE Method	Reduction Method	Embedding Layers	RuBQ 2.0 En		
			RuBQ 2.0 PRM	RuBQ 2.0 Ru	Mintaka
MD-Enc.	Mean	Last	**0.95±0.00**	**1.00±0.00**	**0.87±0.01**
MD-Enc.	Max	Last	0.89±0.00	**1.00±0.00**	0.81±0.01
MD-Enc.	Mean	All	0.86±0.01	**1.00±0.00**	0.80 ±0.01
MD-Enc.	Max	All	0.87±0.01	0.97±0.00	0.76±0.01

Table 11: ROC-AUC↑ for the various settings for KBQA task for MD-Encoder method with various reduction methods. The first row indicates for the standard embeddings extraction and reduction methods, which is used in all other tables and figures.

UE Method	Reduction Method	Embedding Layers	RuBQ 2.0 En		
			RuBQ 2.0 PRM	RuBQ 2.0 Ru	Mintaka
MD-Dec.	Mean	Last	**0.65±0.01**	**0.85±0.01**	**0.74±0.01**
MD-Dec.	Max	Last	0.62±0.01	0.62±0.01	0.69±0.01
MD-Dec.	Mean	All	0.62±0.01	0.82±0.01	0.73±0.01
MD-Dec.	Max	All	0.55±0.01	0.18±0.01	0.66±0.01

Table 12: ROC-AUC↑ for the various settings for KBQA task for MD-Decoder method with various reduction methods. The first row indicates for the standard embeddings extraction and reduction methods, which is used in all other tables and figures.

E Comparison of Computational Time of UE Methods

Table 13 presents the computational time for all considered methods for the NMT task with WMT17 as the ID dataset and PRM as the OOD dataset. These results demonstrate 1100% of the computational overhead time for the ensemble-based methods in comparison with the inference of a single model. Moreover, density-based methods show their computational efficiency and superior other methods by ROC-AUC with 18-20% additional overhead in comparison with the inference of a single model and only 1.5% in comparison with the ensemble-based methods.

UE Method	Inference Time, sec	UE Time, sec	Total, sec
NSP MSP	834.1±23.6	0.0±0.0	834.1±23.6
BLEUVAR	4053.0±67.9	57.1±0.6	4110.1±68.5
MD-ENCODER	150.3±1.8	2.0±2.9	152.3±4.7
MD-DECODER		0.5±0.2	150.8±2.0
RDE-ENCODER		14.6±1.5	164.9±3.3
RDE-DECODER		20.1±1.7	170.4±3.5
EP-SEQ EP-TOK PE-TOK PE-SEQ	9532.9±0.0	0.0±0.0	9532.9±0.0

Table 13: The computational time for the NMT task with WMT17 as the ID dataset and PRM as the OOD dataset. Inference time corresponds to the time needed for model generation for each UE method. UE time corresponds to the time needed for computing uncertainty estimates after the inference stage.

We also present a Table 14 with the time cost results for the KBQA task. The presented table displays the mean values and their corresponding standard deviations for the evaluated uncertainty methods in a specific experiment. The dataset used in this experiment is RUBQ 2.0 English questions, and the out-of-domain (OOD) questions are questions with permuted tokens from the same dataset. Despite the high variability observed in this problem, as indicated by the large standard deviations, we can assert with confidence that the density-based methods exhibit significantly faster performance compared to both the ensemble-based and single model-based methods.

UE Method	Inference Time, sec	UE Time, sec	Total, sec
NSP MSP	2087.2±571.2	0.0±0.0	2087.2±571.2
BLEUVAR	7875.0±3544.7	12.8±0.4	7887.8±3545.1
MD-ENCODER	729.8±246.1	0.7±0.4	730.5±246.5
MD-DECODER		0.7±0.4	730.5±246.5
RDE-ENCODER		0.5±0.0	730.3±246.1
RDE-DECODER		0.5±0.0	730.3±246.1
EP-SEQ EP-TOK PE-TOK PE-SEQ	7925.4±287.3	0.0±0.0	7925.4±287.3

Table 14: The computational time for the KBQA task with RuBQ 2.0 questions in english as the ID dataset and RuBQ 2.0 permuted questions in english as the OOD dataset. Inference time corresponds to the time needed for model generation for each UE method. UE time corresponds to the time needed for computing uncertainty estimates after the inference stage.

F Overall Comparison of OOD Detection Methods on the Machine Translation Task

Figure 4 presents the mean ROC curves over 5 seeds for the models trained on the WMT'20 En-Ru for the selected methods. The second dataset in the title of the figure represents OOD.

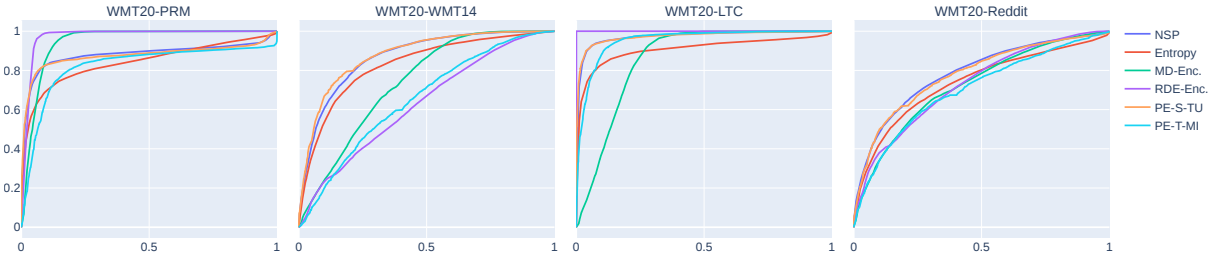

Figure 4: Average ROC curves in various configurations on the NMT task for the selected UE methods. The first dataset in the title represent the ID dataset, the second is the OOD dataset.

Table 15 presents the full results with all the considered methods. This table shows that density-based methods for most of the considered configurations outperform the best ensemble method by a large margin.

UE Method	WMT20 En-Ru				WMT17 En-De			
	PRM	WMT14 Fr	LTC	Reddit	PRM	WMT14 Fr	LTC	Reddit
NSP	0.88 ± 0.02	0.87 ± 0.0	0.97 ± 0.0	0.79 ± 0.01	0.58 ± 0.02	0.37 ± 0.02	0.8 ± 0.01	0.67 ± 0.0
MSP	0.88 ± 0.01	**0.88 ± 0.0**	0.98 ± 0.0	0.74 ± 0.0	0.55 ± 0.02	0.33 ± 0.02	0.78 ± 0.01	0.58 ± 0.0
Entropy	0.84 ± 0.01	0.83 ± 0.01	0.91 ± 0.01	0.74 ± 0.01	0.5 ± 0.02	0.28 ± 0.02	0.72 ± 0.01	0.55 ± 0.01
BLEUVar	0.78 ± 0.01	0.76 ± 0.01	0.97 ± 0.0	0.55 ± 0.0	0.54 ± 0.01	0.49 ± 0.02	0.85 ± 0.01	0.56 ± 0.0
MD-Enc.	0.95 ± 0.0	0.74 ± 0.01	0.86 ± 0.01	0.72 ± 0.0	**1.0 ± 0.0**	**0.92 ± 0.01**	**1.0 ± 0.0**	**0.75 ± 0.0**
MD-Dec.	0.77 ± 0.01	0.47 ± 0.03	0.75 ± 0.04	0.6 ± 0.01	0.86 ± 0.01	0.67 ± 0.01	0.94 ± 0.0	0.65 ± 0.0
RDE-Enc.	**0.97 ± 0.0**	0.63 ± 0.03	**1.0 ± 0.0**	0.73 ± 0.01	0.83 ± 0.01	0.61 ± 0.02	0.83 ± 0.02	0.7 ± 0.01
RDE-Dec.	0.38 ± 0.01	0.5 ± 0.02	0.67 ± 0.05	0.43 ± 0.01	0.53 ± 0.04	0.51 ± 0.03	0.6 ± 0.08	0.5 ± 0.09
EP-S-TU	0.49	0.57	0.76	0.46	0.54	0.4	0.75	0.55
EP-S-RMI	0.64	0.49	0.42	0.63	0.67	0.5	0.54	0.56
EP-T-TU	0.66	0.65	0.86	0.58	0.24	0.17	0.7	0.51
EP-T-MI	0.35	0.45	0.43	0.37	0.65	0.61	0.58	0.43
EP-T-DU	0.72	0.67	0.9	0.65	0.22	0.17	0.64	0.54
EP-T-EPKL	0.36	0.45	0.4	0.43	0.69	0.7	0.63	0.45
EP-T-RMI	0.41	0.45	0.39	0.5	0.75	0.86	0.78	0.56
PE-S-TU	0.88	**0.88**	0.97	0.78	0.58	0.36	0.8	0.65
PE-S-RMI	0.45	0.49	0.6	0.48	0.53	0.42	0.67	0.55
PE-T-TU	0.88	0.83	0.89	**0.81**	0.58	0.3	0.73	0.64
PE-T-MI	0.82	0.65	0.95	0.7	0.77	0.58	0.97	0.66
PE-T-DU	0.88	0.83	0.89	**0.81**	0.58	0.3	0.73	0.64
PE-T-EPKL	0.82	0.65	0.95	0.7	0.77	0.58	0.97	0.66
PE-T-RMI	0.82	0.65	0.95	0.7	0.77	0.58	0.97	0.66

Table 15: AU-ROC↑ for all the considered methods in NMT. The dataset in the first line in the header represent the ID dataset, in the second line is the OOD dataset. We select with **bold** the best results w.r.t. standard deviation.

G Overall Comparison of OOD Detection Methods on the Abstractive Text Summarization Task

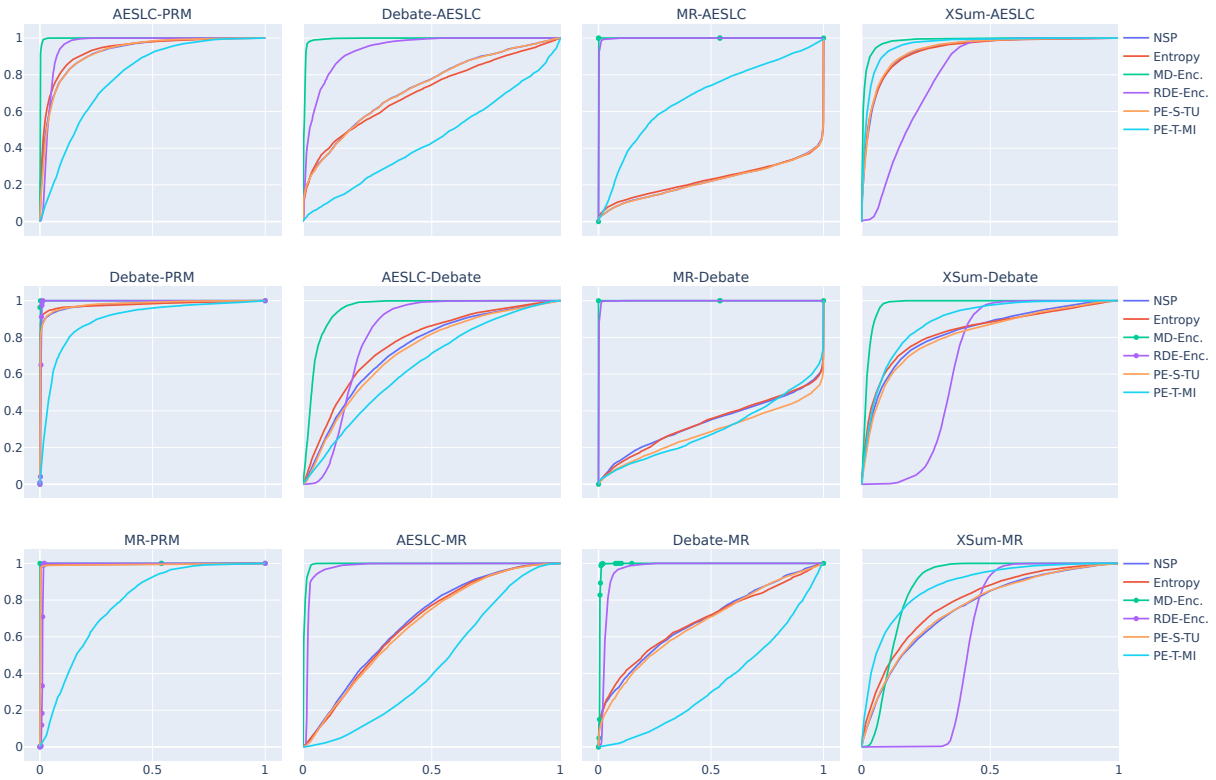

Figure 5: Average ROC curves in various configurations on the ATS task for the selected UE methods. The first dataset in the title represents the ID dataset, the second stands for the OOD dataset.

UE Method	XSum				M.R.			
	AESLC	Debate	M.R.	PRM	AESLC	Debate	XSum	PRM
NSP	0.94 ± 0.0	0.83 ± 0.05	0.76 ± 0.01	0.97 ± 0.0	0.22 ± 0.07	0.34 ± 0.06	0.23 ± 0.04	0.99 ± 0.0
MSP	0.93 ± 0.0	0.82 ± 0.05	0.74 ± 0.01	0.96 ± 0.01	0.2 ± 0.06	0.34 ± 0.06	0.23 ± 0.04	**0.99 ± 0.0**
Entropy	0.94 ± 0.0	0.84 ± 0.05	0.79 ± 0.01	0.98 ± 0.0	0.23 ± 0.07	0.34 ± 0.06	0.17 ± 0.03	**1.0 ± 0.0**
BLEUVar	0.92 ± 0.01	0.83 ± 0.02	0.71 ± 0.04	0.9 ± 0.01	0.64 ± 0.05	0.78 ± 0.03	0.75 ± 0.03	0.85 ± 0.02
MD-Enc.	**0.99 ± 0.0**	**0.98 ± 0.0**	0.87 ± 0.0	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**
MD-Dec.	**0.98 ± 0.0**	0.95 ± 0.0	**0.95 ± 0.01**	0.97 ± 0.0	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**
RDE-Enc.	0.8 ± 0.0	0.66 ± 0.0	0.58 ± 0.01	0.83 ± 0.01	**1.0 ± 0.0**	**1.0 ± 0.0**	0.98 ± 0.0	**0.99 ± 0.0**
RDE-Dec.	0.88 ± 0.01	0.85 ± 0.01	**0.95 ± 0.01**	0.9 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.86 ± 0.03
EP-S-TU	0.95 ± 0.0	0.82 ± 0.01	0.81 ± 0.0	0.97 ± 0.0	0.23 ± 0.06	0.29 ± 0.05	0.19 ± 0.03	**0.99 ± 0.0**
EP-S-RMI	0.85 ± 0.05	0.74 ± 0.01	0.81 ± 0.03	0.89 ± 0.02	0.43 ± 0.02	0.45 ± 0.1	0.32 ± 0.08	0.97 ± 0.03
EP-T-TU	0.94 ± 0.0	0.82 ± 0.01	0.79 ± 0.01	0.97 ± 0.0	0.23 ± 0.07	0.28 ± 0.06	0.14 ± 0.02	**0.99 ± 0.0**
EP-T-MI	0.84 ± 0.02	0.75 ± 0.01	0.83 ± 0.01	0.85 ± 0.01	0.26 ± 0.05	0.49 ± 0.14	0.44 ± 0.12	0.92 ± 0.04
EP-T-RMI	0.95 ± 0.01	0.84 ± 0.0	0.91 ± 0.01	0.94 ± 0.0	0.41 ± 0.03	0.3 ± 0.12	0.21 ± 0.1	0.98 ± 0.01
EP-T-DU	0.85 ± 0.03	0.75 ± 0.02	0.62 ± 0.04	0.94 ± 0.03	0.23 ± 0.07	0.29 ± 0.07	0.16 ± 0.03	**0.99 ± 0.01**
EP-T-EPKL	0.86 ± 0.02	0.76 ± 0.01	0.84 ± 0.01	0.87 ± 0.01	0.25 ± 0.04	0.48 ± 0.14	0.41 ± 0.12	0.93 ± 0.03
PE-S-TU	0.94 ± 0.01	0.81 ± 0.01	0.76 ± 0.02	0.96 ± 0.01	0.22 ± 0.07	0.28 ± 0.06	0.19 ± 0.04	0.99 ± 0.0
PE-S-RMI	0.8 ± 0.08	0.69 ± 0.03	0.77 ± 0.04	0.81 ± 0.0	0.56 ± 0.12	0.56 ± 0.13	0.64 ± 0.12	0.58 ± 0.45
PE-T-TU	0.94 ± 0.01	0.82 ± 0.01	0.79 ± 0.02	0.97 ± 0.0	0.23 ± 0.07	0.29 ± 0.07	0.15 ± 0.03	0.99 ± 0.0
PE-T-MI	0.97 ± 0.0	0.89 ± 0.01	0.88 ± 0.01	0.91 ± 0.02	0.68 ± 0.05	0.3 ± 0.05	0.1 ± 0.03	0.79 ± 0.05
PE-T-RMI	0.96 ± 0.0	0.89 ± 0.01	0.88 ± 0.01	0.91 ± 0.02	0.29 ± 0.03	0.3 ± 0.05	0.1 ± 0.03	0.76 ± 0.05
PE-T-DU	0.94 ± 0.01	0.82 ± 0.01	0.79 ± 0.02	0.97 ± 0.0	0.23 ± 0.07	0.29 ± 0.07	0.15 ± 0.03	0.99 ± 0.0
PE-T-EPKL	0.96 ± 0.0	0.89 ± 0.01	0.88 ± 0.01	0.91 ± 0.02	0.33 ± 0.02	0.3 ± 0.05	0.1 ± 0.03	0.78 ± 0.05

Table 16: Full results (AU-ROC↑) of OOD detection in ATS when XSum / Movie Reviews stand for the ID dataset. The dataset in the second line in the header represent the OOD dataset. We select with **bold** the best results w.r.t. standard deviation.

UE Method	AESLC				Debate			
	Debate	M.R.	XSum	PRM	AESLC	M.R.	XSum	PRM
NSP	0.73 ± 0.02	0.68 ± 0.05	0.72 ± 0.03	0.93 ± 0.01	0.73 ± 0.02	0.69 ± 0.04	0.67 ± 0.02	0.98 ± 0.0
MSP	0.72 ± 0.02	0.63 ± 0.04	0.72 ± 0.02	0.9 ± 0.01	0.69 ± 0.02	0.66 ± 0.05	0.67 ± 0.02	0.97 ± 0.01
Entropy	0.77 ± 0.02	0.67 ± 0.06	0.73 ± 0.04	0.94 ± 0.01	0.71 ± 0.02	0.69 ± 0.05	0.58 ± 0.01	0.98 ± 0.0
BLEUVar	0.67 ± 0.01	0.5 ± 0.01	0.66 ± 0.01	0.83 ± 0.01	0.68 ± 0.02	0.59 ± 0.04	0.69 ± 0.02	0.74 ± 0.02
MD-Enc.	**0.95 ± 0.0**	**0.99 ± 0.0**	**0.96 ± 0.01**	**1.0 ± 0.0**	**0.99 ± 0.0**	**0.99 ± 0.0**	**0.98 ± 0.0**	**1.0 ± 0.0**
MD-Dec.	0.88 ± 0.01	0.95 ± 0.01	**0.96 ± 0.01**	0.79 ± 0.02	0.97 ± 0.01	**1.0 ± 0.0**	**0.97 ± 0.0**	0.96 ± 0.01
RDE-Enc.	0.81 ± 0.01	**0.98 ± 0.0**	0.87 ± 0.01	0.96 ± 0.0	0.94 ± 0.0	0.97 ± 0.0	0.86 ± 0.01	0.99 ± 0.0
RDE-Dec.	0.58 ± 0.02	0.71 ± 0.04	0.72 ± 0.04	0.54 ± 0.01	0.64 ± 0.07	0.86 ± 0.04	0.77 ± 0.04	0.68 ± 0.06
EP-S-TU	0.73 ± 0.01	0.66 ± 0.02	0.72 ± 0.01	0.93 ± 0.0	0.74 ± 0.01	0.7 ± 0.05	0.65 ± 0.02	0.98 ± 0.0
EP-S-RMI	0.55 ± 0.03	0.55 ± 0.05	0.55 ± 0.02	0.82 ± 0.02	0.56 ± 0.01	0.6 ± 0.07	0.45 ± 0.03	0.89 ± 0.05
EP-T-TU	0.76 ± 0.02	0.66 ± 0.02	0.72 ± 0.01	0.95 ± 0.01	0.7 ± 0.02	0.67 ± 0.06	0.57 ± 0.02	0.98 ± 0.0
EP-T-MI	0.61 ± 0.02	0.54 ± 0.04	0.65 ± 0.02	0.72 ± 0.03	0.65 ± 0.02	0.67 ± 0.04	0.48 ± 0.04	0.74 ± 0.04
EP-T-RMI	0.61 ± 0.02	0.51 ± 0.02	0.66 ± 0.02	0.85 ± 0.01	0.44 ± 0.04	0.58 ± 0.11	0.38 ± 0.06	0.91 ± 0.02
EP-T-DU	0.76 ± 0.02	0.67 ± 0.03	0.71 ± 0.02	0.94 ± 0.01	0.7 ± 0.02	0.64 ± 0.07	0.58 ± 0.02	0.97 ± 0.01
EP-T-EPKL	0.61 ± 0.02	0.53 ± 0.04	0.66 ± 0.02	0.77 ± 0.02	0.63 ± 0.02	0.66 ± 0.05	0.46 ± 0.05	0.78 ± 0.04
PE-S-TU	0.72 ± 0.01	0.66 ± 0.04	0.7 ± 0.02	0.92 ± 0.0	0.73 ± 0.02	0.67 ± 0.07	0.67 ± 0.02	0.98 ± 0.01
PE-S-RMI	0.57 ± 0.03	0.53 ± 0.07	0.62 ± 0.01	0.7 ± 0.05	0.6 ± 0.03	0.64 ± 0.11	0.47 ± 0.03	0.7 ± 0.24
PE-T-TU	0.75 ± 0.01	0.65 ± 0.05	0.7 ± 0.02	0.94 ± 0.01	0.7 ± 0.02	0.67 ± 0.07	0.57 ± 0.03	0.98 ± 0.01
PE-T-MI	0.64 ± 0.02	0.46 ± 0.04	0.68 ± 0.02	0.79 ± 0.03	0.45 ± 0.02	0.35 ± 0.03	0.42 ± 0.05	0.9 ± 0.02
PE-T-RMI	0.64 ± 0.02	0.45 ± 0.04	0.68 ± 0.02	0.78 ± 0.03	0.55 ± 0.03	0.34 ± 0.03	0.41 ± 0.07	0.89 ± 0.03
PE-T-DU	0.75 ± 0.01	0.65 ± 0.05	0.7 ± 0.02	0.95 ± 0.01	0.7 ± 0.02	0.67 ± 0.07	0.58 ± 0.03	0.98 ± 0.01
PE-T-EPKL	0.64 ± 0.02	0.45 ± 0.04	0.68 ± 0.02	0.79 ± 0.03	0.53 ± 0.03	0.35 ± 0.03	0.41 ± 0.06	0.89 ± 0.03

Table 17: Full results (AU-ROC↑) of OOD detection in ATS when AESLC / Debate stand for the ID dataset. The dataset in the second line in the header represent the OOD dataset. We select with **bold** the best results w.r.t. standard deviation.

H Overall Comparison of OOD Detection Methods on the Question Answering Task

Figure 6 presents the mean ROC curves over 5 seeds for the T5 (t5-small-ssm-nq) model. The second dataset in the title of the figure represents OOD.

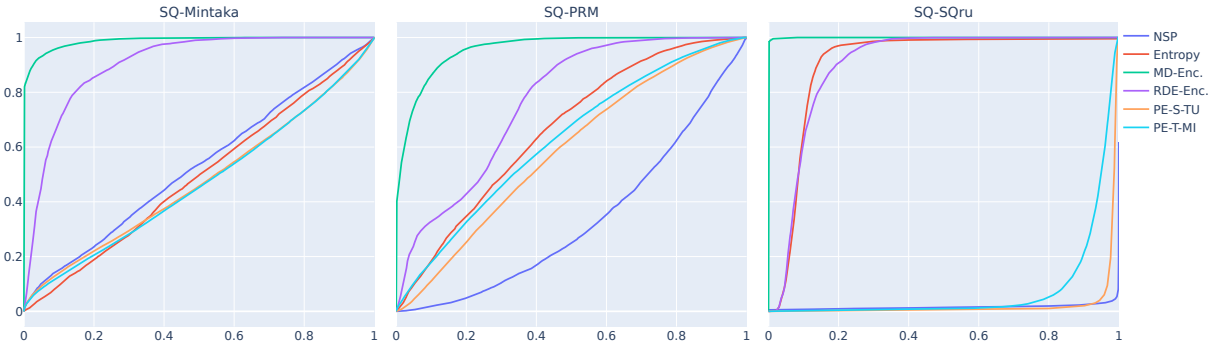

Figure 6: Average ROC curves in various configurations on the KBQA task for the selected UE methods. The first dataset in the title represent the ID dataset, the second is the OOD dataset.

UE Method	Simple Questions			RuBQ 2.0 En		
	Simple Questions Ru	Mintaka	PRM	RuBQ 2.0 Ru	Mintaka	PRM
MSP	0.48 ± 0.01	0.48 ± 0.01	0.51 ± 0.01	0.54 ± 0.01	0.53 ± 0.01	0.54 ± 0.01
NSP	0.53 ± 0.01	0.53 ± 0.01	0.33 ± 0.01	0.41 ± 0.01	0.62 ± 0.01	0.41 ± 0.01
Entropy	0.49 ± 0.00	0.49 ± 0.00	0.66 ± 0.01	0.64 ± 0.01	0.45 ± 0.01	0.64 ± 0.01
BLEUVAR	0.69 ± 0.00	0.50 ± 0.01	0.63 ± 0.00	0.68 ± 0.01	0.49 ± 0.00	0.59 ± 0.01
MD-Enc.	**1.00 ± 0.00**	**0.99 ± 0.00**	**0.96 ± 0.00**	**1.00 ± 0.00**	**0.87 ± 0.01**	**0.95 ± 0.00**
MD-Dec.	0.86 ± 0.00	0.71 ± 0.00	0.66 ± 0.00	0.85 ± 0.01	0.74 ± 0.01	0.65 ± 0.01
RDE-Enc.	0.90 ± 0.00	0.90 ± 0.00	0.77 ± 0.00	0.88 ± 0.00	0.75 ± 0.00	0.74 ± 0.01
RDE-Dec.	0.76 ± 0.01	0.44 ± 0.00	0.49 ± 0.00	0.97 ± 0.01	0.53 ± 0.01	0.60 ± 0.01
EP-S-TU	0.42 ± 0.00	0.47 ± 0.00	0.71 ± 0.01	0.41 ± 0.01	0.50 ± 0.01	0.66 ± 0.01
EP-S-RMI	0.03 ± 0.00	0.50 ± 0.00	0.57 ± 0.01	0.03 ± 0.00	0.54 ± 0.01	0.58 ± 0.01
EP-T-TU	0.70 ± 0.00	0.48 ± 0.01	0.67 ± 0.01	0.70 ± 0.01	0.47 ± 0.01	0.65 ± 0.01
EP-T-MI	0.27 ± 0.00	0.48 ± 0.01	0.69 ± 0.01	0.22 ± 0.00	0.47 ± 0.01	0.64 ± 0.01
EP-T-RMI	0.30 ± 0.00	0.46 ± 0.00	0.72 ± 0.01	0.24 ± 0.00	0.46 ± 0.01	0.67 ± 0.01
EP-T-DU	0.80 ± 0.00	0.48 ± 0.01	0.57 ± 0.01	0.82 ± 0.01	0.50 ± 0.01	0.59 ± 0.01
EP-T-EPKL	0.29 ± 0.00	0.46 ± 0.00	0.73 ± 0.01	0.23 ± 0.00	0.46 ± 0.01	0.67 ± 0.01
PE-S-TU	0.02 ± 0.00	0.48 ± 0.01	0.58 ± 0.01	0.02 ± 0.00	0.55 ± 0.01	0.57 ± 0.01
PE-S-RMI	0.32 ± 0.00	0.47 ± 0.00	0.71 ± 0.00	0.31 ± 0.01	0.49 ± 0.01	0.68 ± 0.01
PE-T-TU	0.15 ± 0.00	0.46 ± 0.00	0.64 ± 0.01	0.15 ± 0.01	0.51 ± 0.01	0.63 ± 0.01
PE-T-MI	0.07 ± 0.00	0.47 ± 0.00	0.62 ± 0.00	0.07 ± 0.00	0.52 ± 0.01	0.61 ± 0.01
PE-T-RMI	0.02 ± 0.00	0.48 ± 0.00	0.64 ± 0.00	0.02 ± 0.00	0.52 ± 0.01	0.59 ± 0.01
PE-T-DU	0.29 ± 0.00	0.46 ± 0.00	0.64 ± 0.01	0.30 ± 0.01	0.51 ± 0.01	0.63 ± 0.01
PE-T-EPKL	0.03 ± 0.00	0.48 ± 0.00	0.64 ± 0.00	0.02 ± 0.00	0.52 ± 0.01	0.59 ± 0.01

Table 18: Full results (AU-ROC↑) of OOD detection in QA obtained using t5-small-ssm-nq when SimpleQuestions / RuBQ2.0 En stand for the ID dataset. The dataset in the second line in the header represent the OOD dataset. Results are obtained by applying a bootstrap technique and averaging over 5 subsamples.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
5 (after conclusion)

☑ A2. Did you discuss any potential risks of your work?
5 (after conclusion)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Not applicable. Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
B 2.4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3; Appendices 3-5

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable in our case since we used public model checkpoints / datasets, released in HuggingFace repo.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.