SUE: Sparsity-based Uncertainty Estimation via Sparse Dictionary Learning

Tamás Ficsor and Gábor Berend
University of Szeged, Hungary
{ficsort,berendg}@inf.u-szeged.hu

Abstract

The growing deployment of deep learning models in real-world applications necessitates not only high predictive accuracy, but also mechanism to identify unreliable predictions, especially in high-stakes scenarios where decision risk must be minimized. Existing methods estimate uncertainty by leveraging predictive confidence (e.g., Softmax Response), structural characteristics of representation space (e.g., Mahalanobis distance), or stochastic variation in model outputs (e.g., Bayesian inference techniques such as Monte Carlo Dropout). In this work, we propose a novel uncertainty estimation (UE) framework based on sparse dictionary learning by identifying dictionary atoms associated with misclassified samples. We leverage pointwise mutual information (PMI) to quantify the association between sparse features and predictive failure. Our method – Sparsity-based Uncertainty Estimation (SUE) – is computationally efficient, offers interpretability via atom-level analysis of the dictionary, has no assumption about the class distribution (unlike Mahalanobis distance). We evaluated SUE on several NLU benchmarks (GLUE and ANLI tasks) and sentiment analysis benchmarks (Twitter, ParaDetox, and Jigsaw). In general, SUE outperforms or matches the performance of other methods. SUE performs particularly well when there is considerable uncertainty in the model, i.e., when the model lacks high precision.

1 Introduction

The application of language models (LMs) in real-world applications is growing rapidly across many domains, including but not limited to healthcare (Razzak et al., 2018), finance (Akoglu et al., 2024), law (Siino et al., 2025), and education (Xiao et al., 2023). Although these models achieve strong performance on various NLP tasks, they are inherently prone to errors (Nguyen and O'Connor, 2015). These errors can be caused by several factors, such as biased or noisy training data (Mukhoti et al., 2023), ambiguity in the task (such as opinions or sentiments), or limitations in the model's training process (Geiping et al., 2022).

A clear example of this is sentiment analysis, where the subjective nature of language can introduce high levels of uncertainty. To address this, it is crucial to identify uncertain instances and handle them differently, such as flagging them for human review, rather than treating all predictions as equally reliable (Geifman and El-Yaniv, 2017; Roberts, 2019).

Further examples of applications where UE plays an important role include clinical decision support, where incorrect predictions can harm patients; legal document analysis, where misinterpretations can lead to legal or compliance risks; content moderation, where errors can suppress valid speech or overlook harmful content; automated customer service, where wrong answers may cause user frustration or financial mistakes; and educational feedback systems, where misleading feedback can negatively impact student learning.

In such scenarios, selective classification (Geifman and El-Yaniv, 2017) can be applied, which allows models to refrain from making predictions when their confidence (or certainty) in a particular instance is insufficient. The option to refrain from predicting can consequently reduce the ratio of misclassified instances. This is typically achieved by associating a confidence (or uncertainty) score with each prediction and introducing a user-defined risk threshold that determines the subset of inputs for which the model's decisions are considered reliable.

By doing so, the system effectively balances coverage – the proportion of samples on which predictions are made – with the overall decision risk. Thus, we can defer the high-risk samples to an expert or inform the user about potential conse-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32901–32918
November 4-9, 2025

Input	Technical	Billing	Spam	Uncertainty
I was charged twice and now I can't log in.	48.4%	51.2%	0.4%	High
I am unable to log in to my account.	92.5%	5.7%	1.8%	Low
I cannot enter.	47.5%	0.7%	51.8%	High
Do you want to lose weight? Download this app now!	1.2%	0.1%	98.7%	Low

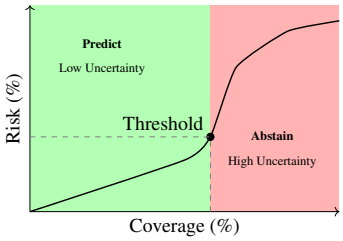


Figure 1: An example of selective classification where the task is to identify and categorize user issues. Selective classification enables a model to assess uncertainty of each sample and abstain from predictions it deems unreliable, allowing those cases to be handled by human reviewers.

quences. We demonstrate such a scenario within the context of selective classification in Figure 1, where the task is to identify and categorize user issues.

Several classical machine learning models, such as Gaussian processes and Bayesian models, inherently provide uncertainty estimates as part of their framework (Liu et al., 2020). In contrast, deep learning models lack this intrinsic capability, necessitating the development of auxiliary metrics to quantify predictive uncertainty. One of the simplest and most widely adopted method is the Softmax Response (SR; Geifman and El-Yaniv (2017)), which derives confidence scores directly from the output probabilities of the softmax layer.

However, it has been well documented that softmax probabilities are often poorly calibrated and tend to be overconfident (Guo et al., 2017). To address this limitation, alternative approaches have been proposed, such as leveraging the Mahalanobis Distance (MD; Lee et al. (2018a)) computed over the hidden representations of the network to provide an uncertainty estimate. Recently, several other methods based on Bayesian inference (BI; Shen et al. (2021)) have also been introduced, and further methods that rely on auxiliary models (Mukhoti et al., 2023) to obtain the desired estimates.

Nevertheless, each of the aforementioned methods presents specific limitations: SR tends to produce overconfident predictions; MD relies on the assumption of a predefined mean and covariance structure for each class; and BI methods require considerable computational overhead, which may be impractical for large-scale deep models.

In this work, we propose a novel framework for uncertainty estimation (UE), based on sparse dictionary learning. Our method imposes no assumptions regarding the spatial distribution of class-specific representations (such as mean or variance) and offers good scalability. Furthermore, the risk of overestimation can be mitigated through the application of stronger regularization or by reducing the number of dictionary atoms used during factorization. Our code is available on our GitHub repository[1].

2 Related Work

Uncertainty estimates in deep neural networks can be derived from the variance in their predictive responses. A natural way to capture this variability is through model ensembles, where each model or head provides an independent prediction for the same input sample (Lakshminarayanan et al., 2017). While ensemble methods are known for their uncertainty estimation capabilities, they introduce significant computational overhead due to the need for training and storing multiple models.

An alternative is to employ Bayesian inference techniques within a single model. A widely adopted method in this context is Monte Carlo Dropout (MC), where stochasticity is introduced during inference by enabling dropout within layers at evaluation time. By performing T stochastic forward passes on the same input, one can approximate the posterior predictive distribution and compute uncertainty estimates from the aggregated outputs. Common aggregation metrics include sampled maximum probability, predictive variance (Gal et al., 2017; Smith and Gal, 2018), and Bayesian Active Learning by Disagreement (Houlsby et al., 2011). Despite its practical appeal, MC Dropout remains computationally demanding at inference time, as it requires multiple forward passes per instance to obtain reliable uncertainty estimates.

A less computationally demanding alternative to ensemble or Bayesian approaches is to derive uncertainty metrics from already computed internal representations or to train a lightweight auxil-

[1] https://github.com/SzegedAI/sue

iary model on top of the model output. To extract meaningful information from the internal model states, one can utilize the Mahalanobis distance as a proxy for uncertainty, as it effectively captures the structure of hidden representations. This method has been successfully applied in various recent works (Lee et al., 2018b; Podolskiy et al., 2021; Vazhentsev et al., 2022, 2023), demonstrating its ability to identify out-of-distribution or low-confidence samples.

Alternatively, auxiliary models trained on the hidden states or prediction outputs offer a flexible and generalizable means of modeling uncertainty. These models can be calibrated to estimate uncertainty (Kendall and Gal, 2017; Kail et al., 2022).

3 Background

In this section, we provide a detailed overview of the UE methods evaluated in this study. Given our focus on sequence classification tasks, all methods operate exclusively on the representation of the `[CLS]` token or the corresponding output logits produced by the model.

3.1 Softmax Response (SR)

Softmax Response (Geifman and El-Yaniv, 2017) relies on the class probabilities generated by the softmax layer in the final classification head. It serves as a simple, yet effective UE baseline. The underlying intuition is that lower maximum softmax probabilities correspond to higher uncertainty in the model prediction. The uncertainty estimate is formally defined as

$$\mathcal{U}_{\text{SR}} = 1 - \max_{c \in C} p\left(y = c \mid x\right),$$

where C denotes the set of all possible classes and $p\left(y = c \mid x\right)$ represents the softmax probability assigned to class c given input x.

3.2 Shannon Entropy (SE)

Shannon Entropy provides a natural and well-established measure of uncertainty, reflecting the amount of unpredictability in a probability distribution. In the context of classification, predictive entropy quantifies the spread of the model's output distribution over the class labels. A high-entropy output indicates greater uncertainty, whereas low entropy reflects confident predictions concentrated on a single class (Malinin and Gales, 2018). SE is formally defined as

$$\mathcal{U}_{\text{SE}} = -\sum_{c \in C} p\left(y = c \mid x\right) \; log\, p\left(y = c \mid x\right)$$

3.3 Mahalanobis Distance (MD)

Mahalanobis distance (MD) is a specialized metric for measuring the distance between points in Euclidean space. In contrast to the Euclidean distance, MD considers the structure of the feature space. Hence, it is a suitable metric for UE that has been used in several studies already (Lee et al., 2018b; Podolskiy et al., 2021; Vazhentsev et al., 2022, 2023). MD can be formalized as

$$\mathcal{U}_{\text{MD}} = \min_{c \in C} \left(h - \mu_c\right) \Sigma^{-1} \left(h - \mu_c\right),$$

where h is the hidden state corresponding to the `[CLS]` token of the input sequence, μ_c is the mean of all vectors associated with class c, and Σ^{-1} is the inverse of the covariance matrix of the train set. Mahalanobis++ (MD++; Müller and Hein, 2025) is such a recent extension of the vanilla MD, which applies unit normalization to the input vectors before calculating the Mahalanobis distance.

3.4 Monte Carlo Dropout (MC)

In recent years, approaches based on Bayesian Inference become widely used for uncertainty estimation. During MC, we perform inference with dropout enabled for T steps. Similar to Vazhentsev et al. (2022), we are going to conduct experiments with the following estimation methods:

Sampled Maximum Probability (SMP):

$$\mathcal{U}_{\text{SMP}} = 1 - \max_{c \in C} \frac{1}{T} \sum_{t=1}^{T} p\left(y = c \mid x_t\right),$$

where $p\left(y = c \mid x_t\right)$ denotes the probability of class c given input x at stochastic step t.

Probability Variance (PV; Gal et al. (2017); Smith and Gal (2018)):

$$\mathcal{U}_{\text{PV}} = \frac{1}{C} \sum_{c=1}^{C} \left(\frac{1}{T} \sum_{t=1}^{T} \left(p\left(y = c \mid x_t\right) - \overline{p^c}\right) \right)$$

where $\overline{p^c} = \frac{1}{T} \sum_t p\left(y = c \mid x_t\right)$.

Bayesian Active Learning by Disagreement (BALD; Houlsby et al., 2011):

$$\mathcal{U}_{\text{BALD}} = -\sum_{c=1}^{C} \overline{p^c} \log \overline{p^c} + \frac{1}{T} \sum_{c,t} p\left(y = c \mid x_t\right) \log p\left(y = c \mid x_t\right).$$

4 Sparsity-based Uncertainty Estimation (SUE)

In this section, we introduce our approach for obtaining UE scores. SUE consists of the following key steps: (1) constructing sparse representations, (2) measuring co-occurrence with pointwise mutual information, and (3) quantifying uncertainty estimate. We summarize the key steps of SUE in Figure 2. SUE is inspired by (Berend, 2020), which demonstrated the utility of relying on PMI statistics of sparse coding-derived features for the task of word sense disambiguation.

4.1 Determining Sparse Representation

A key component of our approach is the use of dictionary learning to represent hidden states in a compact and interpretable manner. Instead of working directly with the dense `[CLS]` embeddings, we decompose them into a sparse linear combination of learned dictionary atoms. Intuitively, the dictionary atoms are fundamental building blocks that capture the most salient and recurring structures in the representation space.

Formally, given the input matrix $X \in \mathbb{R}^{N \times d}$, where each row corresponds to the `[CLS]` token embedding of a sample, we learn the dictionary matrix $D \in \mathbb{R}^{K \times d}$ and the sparse coefficients $\alpha \in \mathbb{R}^{N \times K}$ by solving the following optimization problem (Mairal et al., 2009):

$$\min_{\alpha, D} \frac{1}{2} \|X - \alpha D\|_F^2 + \lambda \|\alpha\|_1, \qquad (1)$$

where K is the number of dictionary atoms, and λ is the regularization coefficient of the sparsity-inducing regularization term $\|\alpha\|_1$. Due to the regularization, only a small fraction of the values in α is positive, making the sparse representations more interpretable. To improve stability, we apply ℓ_1-normalization to each row of X prior to factorization, following a common practice during dictionary learning. We train D on a *validation split disjoint from the model's training data*. At test time, the dictionary D remains fixed, and we infer only the coefficients α_{test}. This separation yields dictionary atoms in D with improved ability to generalize.

An important design choice is which samples to use for training the dictionary matrix. Using all inputs might introduce noise from ambiguous samples, while class-specific dictionaries may underrepresent minority classes. To balance these issues, we only use the correctly classified samples for constructing D, encouraging the dictionary atoms to reflect reliable patterns. We only rely on the set of filtered samples during dictionary learning, and we rely on all samples – regardless of the prediction's correctness – once the dictionary is fixed.

4.2 Identifying Dictionary Atoms Correlated with Prediction Uncertainty

The central idea of our method is to link sparse representations with the reliability of model predictions. Specifically, we aim to identify dictionary atoms that frequently co-occur with misclassified instances. These "uncertain atoms" serve as indicators of prediction failures, and by quantifying their contribution we can derive an uncertainty score.

We measure this association using *Pointwise Mutual Information (PMI)*, which captures how strongly the activation of a given atom correlates with a correct or incorrect classification. Intuitively, PMI highlights atoms that appear disproportionately often in failures compared to what would be expected by chance.

Formally, we construct a co-occurrence matrix $C \in \mathbb{R}^{K \times 2}$ between the activation of each dictionary atom and the binary correctness label $L \in \{0, 1\}^N$ (with 0 and 1 denoting incorrect and correct classification, respectively):

$$C_{k,l} = \sum_{n=1}^{N} \mathbb{I}\left[\alpha^{(n,k)} \neq 0 \wedge L^{(n)} = l\right], \qquad (2)$$

where $\mathbb{I}[\cdot]$ is the indicator function, $\alpha^{(n,k)}$ denotes the coefficient of the dictionary atom k in the sparse decomposition for sample n, and $L^{(n)}$ indicates the correctness of the prediction for sample n. From C, we calculate the maximum likelihood estimates of the joint probability $P(k, l)$ and the marginals $P(k)$ and $P(l)$. The PMI between the coefficient for the dictionary atom k being non-zero and the

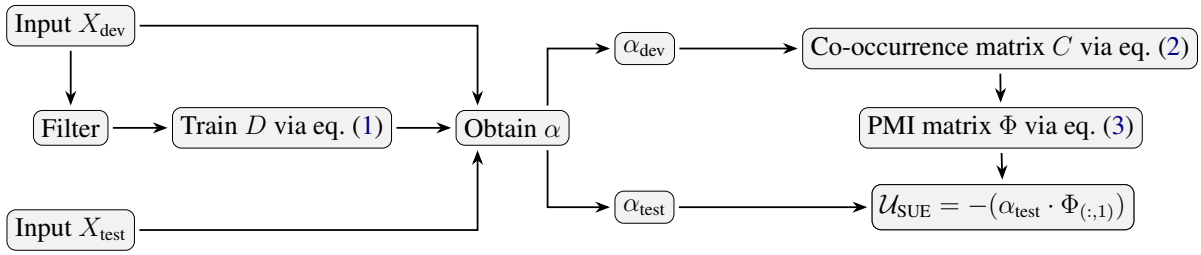


Figure 2: An overview of our approach. X denotes the collection of `[CLS]` tokens, α and D refers to the matrix of sparse representations and the dictionary matrix, respectively. The matrix C includes the co-occurrence statistics between the active (non-zero) elements from α and the correctness of the model predictions. Φ denotes the Pointwise Mutual Information (PMI) matrix, and $\mathcal{U}_{\text{SUE}}$ is the final uncertainty estimation metric. In the figure, *Filter* refers to a special operation that selects from X only those samples that were correctly classified by the model.

correctness label l is then defined as:

$$\Phi_{k,l} = \log \frac{P(k,l)}{P(k)P(l)}. \tag{3}$$

To estimate the uncertainty of a test instance i, we aggregate the PMI scores of its active atoms using the corresponding sparse representation $\alpha_{\text{test}}^{(i)}$:

$$U_{\text{SUE}}^{(i)} = -\left(\alpha_{\text{test}}^{(i)} \cdot \Phi_{(:,1)}\right), \tag{4}$$

where $\Phi_{(:,1)}$ denotes the PMI values associated with correctly classified samples. The negative sign ensures that lower scores correspond to higher classification confidence, or alternatively, higher values of (4) indicate higher prediction uncertainty.

This formulation yields an interpretable uncertainty estimate: the contribution of each dictionary atom to U_{SUE} can be directly inspected, allowing us to trace uncertainty back to specific building blocks of the representation space.

5 Experimental Setup

Models: We evaluated each method using fine-tuned BERT-Base and Large (Devlin et al., 2019), and RoBERTa-Base (Liu et al., 2019) models across a range of natural language understanding and sentiment classification tasks. We link the fine-tuned model checkpoints on our Github repository[1] alongside with our source code. The classification performance of these checkpoints is presented in Table 1. While ANLI may look weak in terms of accuracy, it is important to remember that it was intentionally designed to be challenging. All of the checkpoints perform above the majority and random baseline of 0.33 for each task. However, these results are not directly comparable to the official metrics, since we are using a reduced set of data points, a detail we are going to discuss in the following paragraphs. We provide the fine-tuning hyperparameters in Appendix A.

	BERT-Base	BERT-Large	RoBERTa-Base
ParaDetox	0.97(±0.00)	0.97(±0.00)	0.98(±0.00)
Twitter	0.90(±0.00)	0.88(±0.05)	0.90(±0.00)
Jigsaw	0.93(±0.00)	0.92(±0.01)	0.92(±0.01)
ANLI-R1	0.39(±0.01)	0.39(±0.01)	0.37(±0.04)
ANLI-R2	0.40(±0.01)	0.36(±0.02)	0.36(±0.03)
ANLI-R3	0.38(±0.02)	0.38(±0.02)	0.35(±0.02)

Table 1: The performance of models on the test split.

Datasets: We evaluated our UE approaches on three datasets: ParaDetox (Logacheva et al., 2022), Twitter Sentiment (Davidson et al., 2017), and the Jigsaw Toxic Comment Classification Challenge dataset (cjadams et al., 2017), all binarized for consistency, following (Vazhentsev et al., 2023).

To further assess our method on harder tasks, we also include experiments on Adversarial NLI (ANLI; Nie et al. 2020; Williams et al. 2022). Additionally, we conducted experiments using BERT-Large using the following datasets from the GLUE benchmark (Wang et al., 2018): the Corpus of Linguistic Acceptability (CoLA; Warstadt et al. 2019), the Microsoft Research Paraphrase Corpus (MRPC; Dolan and Brockett 2005), the Question Natural Language Inference (QNLI; Rajpurkar et al. 2016), the Stanford Sentiment Treebank (SST-2; Socher et al. 2013), and the Quora Question Pairs (QQP; Iyer et al. 2017) dataset. We provide further information on the datasets in Appendix A.

Metrics: To evaluate the performance of different UE methods, we adopt the Excess Area Under the Risk-Coverage Curve (eAU-RCC; Geifman et al., 2019). eAU-RCC is based on the Risk-Coverage Curve (RCC; El-Yaniv and Wiener, 2010), which is meant to assess the quality of a selective classifier, where the model is allowed to abstain from predictions considered as uncertain.

RCC measures the extent to which the risk accumulates as more samples – ranked by decreasing confidence – are included in the prediction set. eAU-RCC extends RRC by quantifying the additional risk that incurs due to suboptimal uncertainty ranking compared to an oracle

$$\text{eAU-RC} = \int_0^1 \left(R(c) - R^*(c)\right)\, dc,$$

where $R(c)$ is the empirical risk at coverage level $c \in [0, 1]$, and $R^*(c)$ is the oracle (optimal) risk at coverage level c. Lower eAU-RCC values indicate better uncertainty estimation, as they reflect lower excess risk across different coverage levels.

While some tasks are typically evaluated using specialized metrics (e.g., Matthew's Correlation Coefficient for CoLA), we report accuracy across all datasets to maintain consistency with our uncertainty estimation framework. This is because the empirical risk metric used in Risk-Coverage Curves is based on the zero-one loss.

Hyperparameters: We selected the final hyperparameters based on a validation set performance specific to each task.

As for the MC Dropout-related hyperparameters, we chose $T \in \{10, 25, 50\}$ stochastic steps. Following the findings of Shelmanov et al. (2021), we enabled dropout in *all* layers of our transformer model during inference time.

Related to the dictionary learning component of SUE, we experimented with the regularization coefficient $\lambda \in \{0.01, 0.02, 0.04, 0.06, 0.08, 0.1\}$ and the number of dictionary atoms $K \in \{256, 512, 768\}$. Additionally, for non-GLUE tasks, we selected 8,000 samples as a calibration set that we further split into two disjoint parts with a split ratio of $sr \in \{0.2, 0.4, 0.6, 0.8\}$. For a certain split ratio, we selected the given fraction of calibration set samples for performing dictionary learning, and used the remaining samples of the calibration set for calculating the co-occurrence matrix and the PMI statistics.

6 Results

6.1 Analyzing the Effects of Hyperparameters

We next analyze the effect of our hyperparameters, that is, the fraction of calibration set used for dictionary learning, the choice of regularization strength λ, and the number of dictionary atoms K.

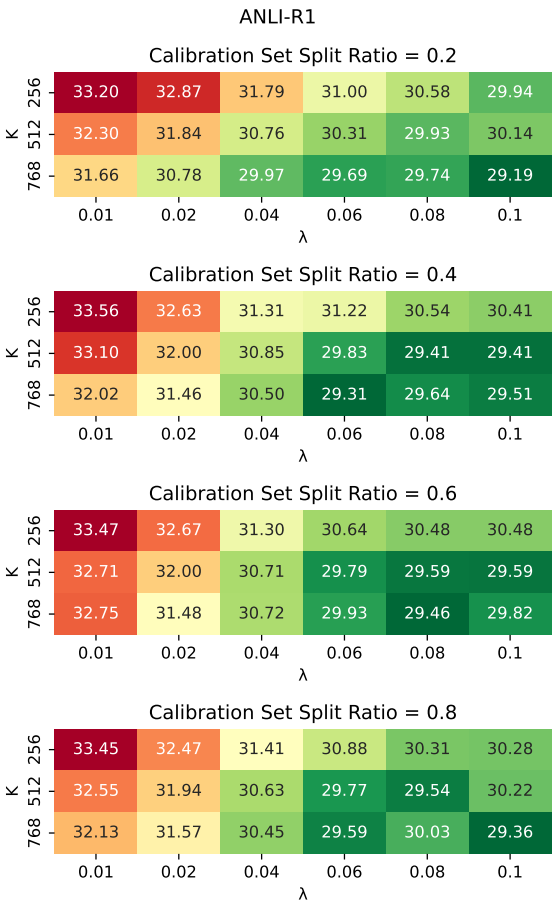


Figure 3: The effect of the percentage of samples used from the calibration set during dictionary learning on BERT-Base model with respect to the regularization coefficient (λ) and the number of dictionary atoms (K).

First, we illustrate the joint effect of choosing our hyperparameters in Figure 3, evaluated on ANLI-R1 with fine-tuned BERT-Base models. While the proportion of calibration set used for dictionary learning did not influence the eAU-RCC scores substantially, the choice of λ and K had a more pronounced effect.

To better understand the joint effects of λ and K, we provide further results for their different combinations in Figure 4, where we fixed the fraction of calibration set samples used for dictionary learning to 20%, and allocated the remaining 80% of the calibration set for calculating the matrix of PMI values in Φ. We observe that on simpler tasks (ParaDetox, Twitter, Jigsaw), the effect of regularization differs compared to harder tasks (ANLI). In the case of simpler tasks, recurring atoms may overfit to the number of samples – that is, fewer atoms are sufficient – while for harder tasks, we can extract more unique atoms.

	ParaDetox	Twitter	Jigsaw	ANLI-R1	ANLI-R2	ANLI-R3
			BERT-Base			
SE	0.37(±0.04)	1.84(±0.15)	0.65(±0.02)	37.06(±1.28)	39.85(±1.10)	36.88(±0.71)
SR	0.37(±0.04)	1.85(±0.15)	0.65(±0.02)	37.00(±1.32)	39.98(±1.03)	37.06(±0.84)
MD	0.52(±0.11)	4.00(±0.18)	5.91(±4.24)	46.62(±3.24)	43.68(±1.23)	44.64(±1.92)
MD++	0.54(±0.08)	3.88(±0.16)	4.87(±3.05)	46.51(±2.79)	43.79(±1.24)	43.85(±1.17)
MC-SMP	0.30(±0.06)	**1.73(±0.12)**	0.55(±0.02)	36.70(±1.57)	39.41(±1.10)	36.91(±0.61)
MC-PV	**0.27(±0.04)**	1.88(±0.09)	0.58(±0.03)	38.52(±1.40)	40.11(±0.97)	38.88(±0.91)
MC-BALD	0.29(±0.04)	2.06(±0.06)	0.67(±0.04)	38.91(±1.38)	40.39(±1.14)	39.51(±0.93)
SUE	0.46(±0.14)	1.95(±0.23)	**0.40(±0.05)**	**29.55(±2.11)**	**31.65(±0.96)**	**32.76(±1.13)**
			BERT-Large			
SE	0.45(±0.24)	2.18(±0.89)	0.67(±0.05)	38.81(±1.98)	41.70(±2.18)	39.19(±1.89)
SR	0.45(±0.24)	**2.17(±0.90)**	0.67(±0.05)	38.72(±1.92)	41.73(±2.40)	39.24(±2.10)
MD	0.85(±0.34)	9.82(±11.83)	5.08(±4.67)	47.20(±3.95)	44.41(±1.97)	43.26(±2.50)
MD++	0.75(±0.30)	9.71(±12.00)	5.56(±5.62)	45.26(±4.62)	43.91(±2.60)	43.07(±2.51)
MC-SMP	0.33(±0.14)	2.24(±1.11)	0.55(±0.06)	38.84(±1.81)	40.75(±1.30)	38.99(±1.90)
MC-PV	**0.31(±0.12)**	3.05(±2.33)	0.69(±0.13)	42.24(±1.28)	41.32(±1.06)	41.01(±2.74)
MC-BALD	0.34(±0.12)	4.34(±4.48)	0.89(±0.20)	43.13(±1.50)	41.85(±1.14)	41.12(±2.81)
SUE	0.67(±0.21)	2.50(±1.11)	**0.44(±0.10)**	**30.65(±2.27)**	**37.75(±3.86)**	**32.15(±2.08)**
			RoBERTa-Base			
SE	0.17(±0.03)	2.03(±0.21)	0.64(±0.09)	40.21(±4.64)	40.75(±1.82)	**39.79(±3.07)**
SR	0.17(±0.03)	2.04(±0.21)	0.64(±0.09)	39.67(±4.01)	40.51(±1.52)	40.61(±2.42)
MD	0.49(±0.12)	3.76(±0.50)	6.45(±3.36)	45.47(±1.40)	43.53(±1.65)	44.50(±2.25)
MD++	0.28(±0.06)	3.48(±0.25)	3.91(±1.83)	45.24(±1.98)	42.96(±1.64)	43.89(±2.85)
MC-SMP	**0.12(±0.01)**	**1.92(±0.15)**	0.51(±0.06)	40.35(±4.65)	41.76(±2.05)	41.24(±3.10)
MC-PV	0.14(±0.02)	2.27(±0.25)	0.64(±0.09)	41.13(±2.17)	42.66(±1.60)	42.13(±3.76)
MC-BALD	0.17(±0.02)	2.55(±0.38)	0.84(±0.17)	41.74(±2.02)	42.94(±1.55)	42.21(±3.68)
SUE	0.38(±0.29)	2.18(±0.17)	**0.38(±0.03)**	**35.73(±10.84)**	**39.33(±4.77)**	40.02(±5.50)

Table 2: eAU-RCC scores (multiplied by 100) for each task-method pair, with lower scores indicating better UE performance. The best results are in bold and the second best results are underlined. For the individual MC-* approaches, we report the best result that we obtained over the different choices of T. We include the detailed MC-* results that we obtained for the different values of T in Table 7.

Compared to the theoretical recommendation $\lambda = 1.2/\sqrt{d}$, which is approximately 0.0433 for BERT-Base ($d = 768$), we observe only minor differences in outcomes. This makes it a reasonable initial value for λ. However, selecting the number of atoms requires further investigation, and care should be taken to avoid overfitting. As a general rule, we suggest fixing λ to the theoretical value and choosing the number of atoms according to the number of available samples and the difficulty of the task at hand. Results for other model–task pairs are presented in Appendix A, showing similar trends.

We relied on the validation set performance when selecting hyperparameters. In order to assess the sensitivity to hyperparameter choice, we provide the paired validation and test set performances for all tested hyperparameter combinations for BERT-Base in Figure 5. We can see that for the individual tasks, there is low variability in the performances along both axes and that the hyperparameters that perform well on the validation set also perform well on the test set, indicating the robustness of SUE to hyperparameter choices. We report similar plots indicating the robustness of SUE to hyperparameter choices when used in conjunction with other models in Appendix B.

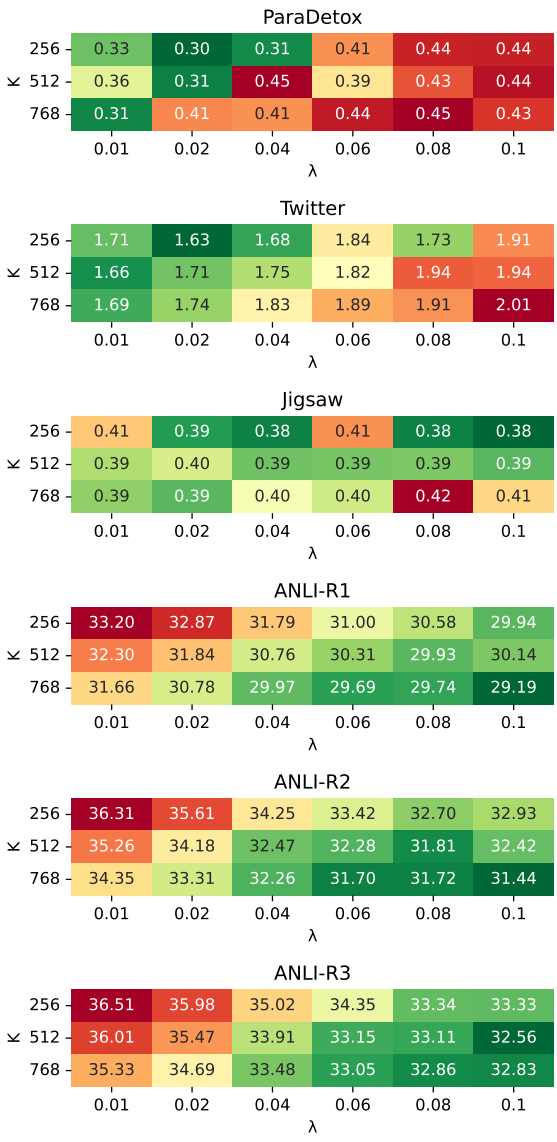


Figure 4: The effect of sparse factorization on BERT-Base model with respect to the regularization coefficient (λ) and the number of dictionary atoms (K) on every task.

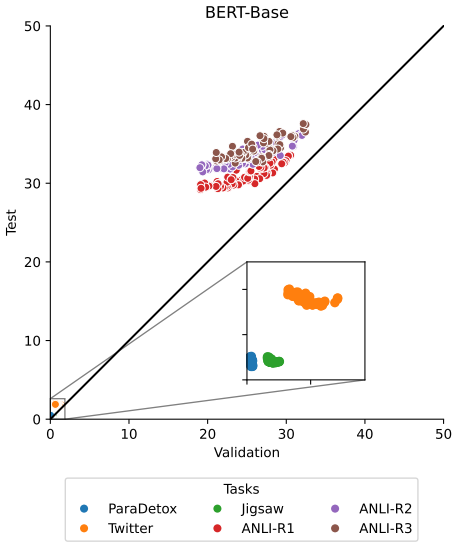


Figure 5: Relationship of development and test set eAU-RCC scores when applying SUE with different hyperparameter choices for fine-tuned BERT-base models.

	CoLA	MRPC	QNLI	SST-2	QQP
SR	4.04	17.04	11.80	1.04	7.39
SE	4.17	17.04	11.80	1.04	7.39
MD	8.25	28.85	13.73	9.05	10.27
MD++	21.27	35.64	13.51	5.55	9.79
MC-SMP	4.51	29.59	8.10	0.97	7.81
MC-PV	4.17	30.47	8.11	**0.90**	6.26
MC-BALD	4.38	31.01	8.14	**0.90**	7.73
SUE	**4.00**	**11.64**	**7.66**	1.24	**5.59**

Table 3: eAU-RCC scores (multiplied by 100) for each GLUE task on BERT-large models. Each MC method were evaluated with $T = 25$ steps.

6.2 Quantitative Results

In this section, we compare the performance of various UE methods across a range of sequence classification tasks. Table 2 presents the eAU-RCC scores, where lower values indicate better uncertainty estimation (i.e., lower risk at increasing coverage levels).

In general, we observe that baseline methods such as SE, SR, and MD(++) perform consistently worse than more advanced approaches. MC based methods achieve competitive results on simpler tasks like ParaDetox and Twitter, but their effectiveness diminishes on more challenging reasoning benchmarks such as ANLI.

By contrast, our proposed SUE demonstrates robust performance across all tasks and models, often achieving the best or second-best scores. Notably, SUE substantially outperforms alternative methods on Jigsaw and ANLI, indicating its strength in handling harder tasks. These results highlight SUE's stability and efficiency as a reliable alternative to sampling-heavy approaches, particularly in scenarios where traditional MC methods become computationally expensive.

Similar trends can be seen on GLUE as well which can be seen in Table 3. These results are limited to BERT-large models only. The corresponding model performances can be seen in the Appendix (Table 5), which shows that SUE only performs worse on SST-2.

These results suggest that SUE is particularly well-suited for tasks with higher overall uncertainty, i.e., tasks where the model has lower precision. In general, the harder the task, the more likely SUE is to outperform other methods – a trend consistently observed across GLUE, ANLI, and sentiment classification benchmarks.

6.3 Interpretability of SUE

To better understand the behavior of SUE, we inspected the distribution of the uncertainty scores of ParaDetox. During that inspection, we had a look at the most confident test samples according to their SUE score. We further filtered these instances to keep only those for which have been misclassified. These samples can be seen in Table 4.

Some errors can be easily traced back to simple mistakes in the dataset, such as the second example in Table 4, where a similar sentence appears twice, with the only difference being that the word *"generation"* was replaced with *"gen"* which – according to the annotation – changes the label to toxic. There are several other examples where no or unsubstantial modifications were made to the sentence. Additionally, other samples exhibit simple ambiguities, which may stem from a lack of context or differences in personal tolerance.

We can further explore why SUE assigns high confidence to certain misclassified samples. Figure 6 visualizes the contribution of each atom contribute towards the final uncertainty estimate. These visualizations correspond to the same samples listed in Table 4, shown in the same order.

In the plot, each bar represents the influence of a dictionary atom. Red indicates a contribution towards confidence, while blue indicates a contribution towards uncertainty. This visualization can be interpreted as a force plot: red bars push the estimate towards higher confidence (left), and blue bars push towards greater uncertainty (right). The final SUE score is determined by the point at which these opposing forces meet.

By observing Figure 6, we can make a few remarks about the samples. The first six samples were chosen because they contain no uncertain atoms, while the remaining samples have only one or two atoms that push them toward uncertainty.

Text
but screw everything else they do, especially canadian content.
i had the new generation rondo as a rental and i agree, its crap.
damn straight!! dd when you said it to me on saturday night i lost my mind
lol rt when bored all the stupid things come to your head
we are not totally stupid, we know she works for cnn.
oh shit eddie izzard this night i forget
vice just went full - retard.
sometimes, i just sit here on twitter, thinking i am not stupid.
eh! whoresnops for life! thanks for the warning.

Table 4: Misclassified instances of ParaDetox with high confidence according to the SUE scores. All instances have a *neutral* ground truth label.

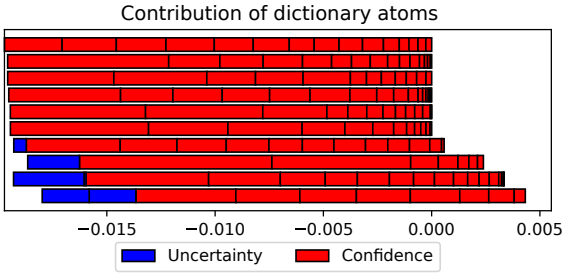


Figure 6: We present the contribution of each atom towards the final SUE score. Each row in the plot corresponds to one of the samples listed in Table 4, presented in the same order.

7 Conclusions

In this work, we introduced SUE, a sparse coding-based uncertainty estimation method and evaluated it on sequence classification tasks. By leveraging sparse representations of the final hidden states of transformer models, our approach effectively captures meaningful patterns aligned with the model's confidence. Through extensive experiments on GLUE, ANLI and Sentiment benchmarks, we demonstrated that SUE consistently outperforms classical confidence-based methods such as Softmax Response and Shannon Entropy, as well as MC dropout variants. This is particularly the case in scenarios where model precision is low and uncertainty estimation becomes critical. Our method yields the best overall performance in terms of eAU-RCC, and offers more stable and interpretable risk-coverage behavior.

We complemented our quantitative evaluation with qualitative analyzes, revealing how atoms contribute toward the final uncertainty scores, helping the identification of those cases where high or low model prediction confidence is unwarranted. Additionally, we linked the structure of the learned PMI matrix to downstream estimation quality, offering insight into potential failure cases. Overall, our results suggest that sparse representations provide a powerful and interpretable foundation for uncertainty estimation.

Limitations

Our approach relies on sparse dictionary learning, which requires setting the hyperparameters related to the number of dictionary atoms (K) and the sparsity-inducing regularization coefficient (λ). However, our ablation study on the choice of these hyperparameters showed little variability in performance, and selecting theoretical values was sufficient to achieve the expected outcomes. We also note that alternative uncertainty estimators likewise involve hyperparameters in one form or another.

Acknowledgments

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. The research received additional support from the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory and project no. 2024-1.2.3-HU-RIZONT-2024-00017, which has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the 2024-1.2.3-HU-RIZONT funding scheme. Additionally, we are grateful for the possibility to use ELKH Cloud (see Héder et al., 2022; https://science-cloud.hu/) which helped us achieve the results published in this paper.

References

Leman Akoglu, Nitesh V. Chawla, Josep Domingo-Ferrer, Eren Kurshan, Senthil Kumar, Vidyut M. Naware, José A. Rodríguez-Serrano, Isha Chaturvedi, Saurabh Nagrecha, Mahashweta Das, and Tanveer A. Faruquie. 2024. Machine learning in finance. In *KDD*, page 6703. ACM.

Gábor Berend. 2020. Sparsity makes sense: Word sense disambiguation using sparse contextualized word representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8498–8508, Online. Association for Computational Linguistics.

cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. 2017. Toxic comment classification challenge. `https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge`. Kaggle.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):512–515.

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

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

Ran El-Yaniv and Yair Wiener. 2010. On the foundations of noise-free selective classification. *J. Mach. Learn. Res.*, 11:1605–1641.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. 2019. Bias-reduced uncertainty estimation for deep neural classifiers. In *ICLR (Poster)*. OpenReview.net.

Jonas Geiping, Micah Goldblum, Phillip Pope, Michael Moeller, and Tom Goldstein. 2022. Stochastic training is not necessary for generalization. In *ICLR*. OpenReview.net.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *roceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1321–1330. PMLR.

Mihály Héder, Ernő Rigó, Dorottya Medgyesi, Róbert Lovas, Szabolcs Tenczer, Ferenc Török, Attila Farkas, Márk Emődi, József Kadlecsik, György Mező, Ádám Pintér, and Péter Kacsuk. 2022. The past, present and future of the ELKH cloud. *Információs Társadalom*, 22(2):128.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745.

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

Roman Kail, Kirill Fedyanin, Nikita Muravev, Alexey Zaytsev, and Maxim Panov. 2022. Scaleface: Uncertainty-aware deep metric learning. *CoRR*, abs/2209.01880.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, pages 5574–5584.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, pages 6402–6413.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018a. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018b. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, pages 7167–7177.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Advances in Neural Information Processing Systems*, volume 33, pages 7498–7512. Curran Associates, Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. Cite arxiv:1907.11692.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland. Association for Computational Linguistics.

Julien Mairal, Francis R. Bach, Jean Ponce, and Guillermo Sapiro. 2009. Online dictionary learning for sparse coding. In *ICML*, volume 382 of *ACM International Conference Proceeding Series*, pages 689–696. ACM.

Andrey Malinin and Mark Gales. 2018. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. 2023. Deep deterministic uncertainty: A new simple baseline. In *CVPR*, pages 24384–24394. IEEE.

Maximilian Müller and Matthias Hein. 2025. Mahalanobis++: Improving OOD detection via feature normalization. In *Forty-second International Conference on Machine Learning*.

Khanh Nguyen and Brendan O'Connor. 2015. Posterior calibration and exploratory analysis for natural language processing models. In *EMNLP*, pages 1587–1598. The Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13675–13682.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, pages 2383–2392. Association for Computational Linguistics.

Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. 2018. *Deep Learning for Medical Image Processing: Overview, Challenges and the Future*, pages 323–350. Springer International Publishing, Cham.

Sarah T. Roberts. 2019. *Behind the Screen: Content Moderation in the Shadows of Social Media*. Yale University Press.

Artem Shelmanov, Evgenii Tsymbalov, Dmitri Puzyrev, Kirill Fedyanin, Alexander Panchenko, and Maxim Panov. 2021. How certain is your Transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1833–1840, Online. Association for Computational Linguistics.

Yilin Shen, Yen-Chang Hsu, Avik Ray, and Hongxia Jin. 2021. Enhancing the generalization for intent classification and out-of-domain detection in SLU. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2443–2453, Online. Association for Computational Linguistics.

Marco Siino, Mariana Falco, Daniele Croce, and Paolo Rosso. 2025. Exploring llms applications in law: A literature review on current legal nlp approaches. *IEEE Access*, 13:18253–18276.

Lewis Smith and Yarin Gal. 2018. Understanding measures of uncertainty for adversarial example detection. In *UAI*, pages 560–569. AUAI Press.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, Manvel Avetisian, and Leonid Zhukov. 2022. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8237–8252, Dublin, Ireland. Association for Computational Linguistics.

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023. Hybrid uncertainty quantification for selective text classification in ambiguous tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681, Toronto, Canada. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Tristan Thrush, and Douwe Kiela. 2022. Anlizing the adversarial natural language inference dataset.

Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023. Evaluating reading comprehension exercises generated by LLMs: A showcase of ChatGPT in education applications. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 610–625, Toronto, Canada. Association for Computational Linguistics.

A Models and Datasets

For our ANLI and sentiment classification models, we considered the following hyperparamters:

- Learning rate: $\{1e-5, 2e-5, 5e-5, 1e-6, 2e-6, 5e-6\}$,
- Batch size: $\{8, 16, 32\}$,
- Weight decay: $\{0, 0.1, 0.01\}$.

	$\lvert Dev\rvert$	$\lvert Test\rvert$	Test Accuracy
CoLA	417	626	84.85
MRPC	163	245	87.99
QNLI	2,185	3,278	92.23
SST-2	348	524	93.46
QQP	16,172	24,258	91.07

Table 5: Basic statistics of the datasets and model performance. The test accuracy is the classification accuracy of the model that we evaluate from an UE perspective.

The optimal hyperparameters are saved along the checkpoints and can be seen on their corresponding repository on Huggingface. In case of GLUE, we rely on well established, publicly available checkpoints[2] where we opted to use BERT-Large models.

Each model relies on a subset of the samples, which varies on a task-by-task bases:

ANLI: For each subset (R1, R2, R3), we take 8,000 samples for training and 8,000 for calibration from the official training set. The development and test sets stay the same.

ParaDetox: This dataset has about 19.7K samples in a single split. We use 8,000 for training, 8,000 for calibration, 1,000 for development, and the rest for testing.

Twitter: Following the same setup as ParaDetox, we use 8,000 for training, 8,000 for calibration, 1,000 for development, and the rest for testing.

Jigsaw: From the official training set, we use 8,000 samples for training and 8,000 for calibration. From the official test set, we take 1,000 for development and 3,000 for testing.

We use the train set to fine-tune the models, the calibration set to make further experiments on the hyperparameter space of UE methods, validation set for model selection, and test for final evaluation.

For the GLUE tasks, we use more constrained splits, since some of the GLUE tasks include much fewer samples compared to the sentiment analysis tasks that we experimented with. We left the train split intact and made our development and test splits from the official development split in a 40%-60% ratio. The statistics and model performances are provided in Table 5.

[2] https://huggingface.co/yoshitomo-matsubara

B Sparse Hyperparameter Choice

We present all hyperparameter combinations (Model, Task, Number of samples, Number of atoms, Regularization strength) in three figures: Figure 7 for BERT-Base, Figure 8 for BERT-Large, and Figure 9 for RoBERTa-Base.

Across models, we observe similar behavior, suggesting that the task has a stronger influence on our UE score than the choice of model. We also confirm our earlier finding that the number of dictionary atoms has little effect, and that the theoretical choice of λ tends to be a consistently good option.

C Interpretability

In Figure 11a, we observe that traditional confidence-based approaches such as Softmax Response (SR) and Shannon Entropy (SE) show significant risk variance at low coverage, especially when abstaining only from the most confident samples. These early fluctuations suggest that a few misclassified instances are mistakenly considered high-confidence, although their absolute number remains small. We can observer this same phenomenon where SUE exhibits this behavior on MRPC (see Figure 11b).

To better understand this behavior on MRPC, we inspected the distribution of the uncertainty scores. During that inspection, we had a look at the top-25% most confident test samples according to their SUE score. We further filtered these instances to keep only those for which the predicted label did not match the expected ground truth label. We provide these instances in Table 6, ordered by decreasing confidence under SUE. Manual inspection confirms that several of these were mislabeled or ambiguous, further explaining the early rise in risk under SUE in MRPC.

Figure 7: Effect of parameters on the final score with BERT-Base.

Sentence	Expected label
But I would rather be talking about high standards than low standards . " " I would rather be talking about positive numbers rather than negative .	Equivalent
" Overwhelmingly the Windows brand really resonated with them . " " Windows was the part of the experience that really resonated with people . "	Equivalent
" They 've been in the stores for over six weeks , " says Carney . The quarterlies usually stay in stores for between six to eight weeks , " Carney added .	Equivalent
Its closest living relatives are a family frogs called sooglossidae that are found only in the Seychelles in the Indian Ocean . Its closest relative is found in the Seychelles Archipelago , near Madagascar in the Indian Ocean .	Equivalent
About 10 percent of high school and 16 percent of elementary students must be proficient at math . In math , 16 percent of elementary and middle school students and 9.6 percent of high school students must be proficient .	Equivalent
The additional contribution brings total U.S. food aid to North Korea this year to 100,000 tonnes . The donation of 60,000 tons brings the total of U.S. contributions for the year to 100,000 .	Equivalent

Table 6: Misclassified instances of MRPC with high confidence by SUE scores.

Figure 8: Effect of parameters on the final score with BERT-Large.

Figure 9: Effect of parameters on the final score with RoBERTa-Base.

	ParaDetox	Twitter	Jigsaw	ANLI-R1	ANLI-R2	ANLI-R3
			BERT-Base			
SE	0.37(±0.04)	1.84(±0.15)	0.65(±0.02)	37.06(±1.28)	39.85(±1.10)	36.88(±0.71)
SR	0.37(±0.04)	1.85(±0.15)	0.65(±0.02)	37.00(±1.32)	39.98(±1.03)	37.06(±0.84)
MD	0.52(±0.11)	4.00(±0.18)	5.91(±4.24)	46.62(±3.24)	43.68(±1.23)	44.64(±1.92)
MD++	0.54(±0.08)	3.88(±0.16)	4.87(±3.05)	46.51(±2.79)	43.79(±1.24)	43.85(±1.17)
MC-SMP ($T = 10$)	0.31(±0.04)	**1.73(±0.12)**	0.56(±0.02)	36.74(±1.59)	39.55(±1.19)	36.94(±0.71)
MC-SMP ($T = 25$)	0.31(±0.06)	1.74(±0.12)	0.56(±0.02)	36.70(±1.57)	39.41(±1.10)	37.00(±0.59)
MC-SMP ($T = 50$)	0.30(±0.06)	1.75(±0.13)	0.55(±0.02)	36.71(±1.53)	39.44(±1.07)	36.91(±0.61)
MC-PV ($T = 10$)	0.30(±0.04)	1.95(±0.10)	0.61(±0.03)	39.45(±1.23)	40.11(±0.97)	39.42(±0.94)
MC-PV ($T = 25$)	0.28(±0.04)	1.91(±0.12)	0.59(±0.02)	38.81(±1.09)	40.25(±0.94)	39.18(±0.65)
MC-VP ($T = 50$)	**0.27(±0.04)**	1.88(±0.09)	0.58(±0.03)	38.52(±1.40)	40.33(±0.66)	38.88(±0.91)
MC-BALD ($T = 10$)	0.32(±0.04)	2.22(±0.13)	0.73(±0.04)	39.90(±1.10)	40.39(±1.14)	40.15(±0.93)
MC-BALD ($T = 25$)	0.29(±0.04)	2.12(±0.11)	0.69(±0.04)	39.29(±1.01)	40.40(±1.12)	39.82(±0.73)
MC-BALD ($T = 50$)	0.29(±0.04)	2.06(±0.06)	0.67(±0.04)	38.91(±1.38)	40.44(±0.83)	39.51(±0.93)
SUE	0.46(±0.14)	1.95(±0.23)	**0.40(±0.05)**	**29.55(±2.11)**	**31.65(±0.96)**	**32.76(±1.13)**
			BERT-Large			
SE	0.45(±0.24)	2.18(±0.89)	0.67(±0.05)	38.81(±1.98)	41.70(±2.18)	39.19(±1.89)
SR	0.45(±0.24)	**2.17(±0.90)**	0.67(±0.05)	38.72(±1.92)	41.73(±2.40)	39.24(±2.10)
MD	0.85(±0.34)	9.82(±11.83)	5.08(±4.67)	47.20(±3.95)	44.41(±1.97)	43.26(±2.50)
MD++	0.75(±0.30)	9.71(±12.00)	5.56(±5.62)	45.26(±4.62)	43.91(±2.60)	43.07(±2.51)
MC-SMP ($T = 10$)	0.35(±0.17)	2.33(±1.25)	0.56(±0.05)	38.84(±1.81)	40.75(±1.30)	39.03(±1.95)
MC-SMP ($T = 25$)	0.34(±0.16)	2.29(±1.19)	0.55(±0.06)	38.91(±1.75)	41.02(±1.51)	39.05(±1.88)
MC-SMP ($T = 50$)	0.33(±0.14)	2.24(±1.11)	0.55(±0.06)	38.94(±1.74)	40.81(±1.35)	38.99(±1.90)
MC-PV ($T = 10$)	0.34(±0.15)	3.19(±2.49)	0.71(±0.13)	42.24(±1.28)	41.86(±1.45)	41.01(±2.74)
MC-PV ($T = 25$)	0.34(±0.15)	3.09(±2.40)	0.69(±0.13)	42.26(±1.12)	41.37(±1.14)	41.11(±2.70)
MC-VP ($T = 50$)	**0.31(±0.12)**	3.05(±2.33)	0.69(±0.13)	42.25(±1.08)	41.32(±1.06)	41.03(±2.63)
MC-BALD ($T = 10$)	0.40(±0.17)	5.34(±6.21)	0.92(±0.21)	43.13(±1.50)	42.37(±1.58)	41.12(±2.81)
MC-BALD ($T = 25$)	0.38(±0.16)	4.95(±5.64)	0.89(±0.20)	43.20(±1.36)	42.00(±1.30)	41.25(±2.85)
MC-BALD ($T = 50$)	0.34(±0.12)	4.34(±4.48)	0.89(±0.20)	43.25(±1.42)	41.85(±1.14)	41.13(±2.73)
SUE	0.67(±0.21)	2.50(±1.11)	**0.44(±0.10)**	**30.65(±2.27)**	**37.75(±3.86)**	**32.15(±2.08)**
			RoBERTa-Base			
SE	0.17(±0.03)	2.03(±0.21)	0.64(±0.09)	40.21(±4.64)	40.75(±1.82)	**39.79(±3.07)**
SR	0.17(±0.03)	2.04(±0.21)	0.64(±0.09)	39.67(±4.01)	40.51(±1.52)	40.61(±2.42)
MD	0.49(±0.12)	3.76(±0.50)	6.45(±3.36)	45.47(±1.40)	43.53(±1.65)	44.50(±2.25)
MD++	0.28(±0.06)	3.48(±0.25)	3.91(±1.83)	45.24(±1.98)	42.96(±1.64)	43.89(±2.85)
MC-SMP ($T = 10$)	0.14(±0.02)	1.94(±0.15)	0.52(±0.06)	40.89(±5.51)	42.15(±2.49)	41.76(±3.59)
MC-SMP ($T = 25$)	0.13(±0.03)	**1.92(±0.16)**	0.51(±0.06)	40.70(±5.02)	42.12(±2.62)	41.70(±3.51)
MC-SMP ($T = 50$)	**0.12(±0.01)**	**1.92(±0.15)**	0.51(±0.06)	40.35(±4.65)	41.76(±2.05)	41.24(±3.10)
MC-PV ($T = 10$)	0.17(±0.04)	2.37(±0.21)	0.68(±0.11)	41.13(±2.17)	42.66(±1.60)	42.99(±3.80)
MC-PV ($T = 25$)	0.16(±0.04)	2.29(±0.23)	0.65(±0.10)	42.02(±2.94)	43.11(±2.56)	42.13(±3.76)
MC-VP ($T = 50$)	0.14(±0.02)	2.27(±0.25)	0.64(±0.09)	42.05(±3.31)	43.14(±2.91)	42.58(±4.24)
MC-BALD ($T = 10$)	0.21(±0.04)	2.72(±0.33)	0.95(±0.23)	41.74(±2.02)	42.94(±1.55)	43.23(±3.54)
MC-BALD ($T = 25$)	0.19(±0.04)	2.58(±0.35)	0.86(±0.18)	42.58(±2.53)	43.50(±2.45)	42.21(±3.68)
MC-BALD ($T = 50$)	0.17(±0.02)	2.55(±0.38)	0.84(±0.17)	42.64(±2.93)	43.50(±2.78)	42.67(±4.16)
SUE	0.38(±0.29)	2.18(±0.17)	**0.38(±0.03)**	**35.73(±10.84)**	**39.33(±4.77)**	40.02(±5.50)

Table 7: eAU-RCC scores (multiplied by 100) for each task-method pair, lower the score indicates better UE performance. We indicate the best performing approach with bold text and the second one with underline.

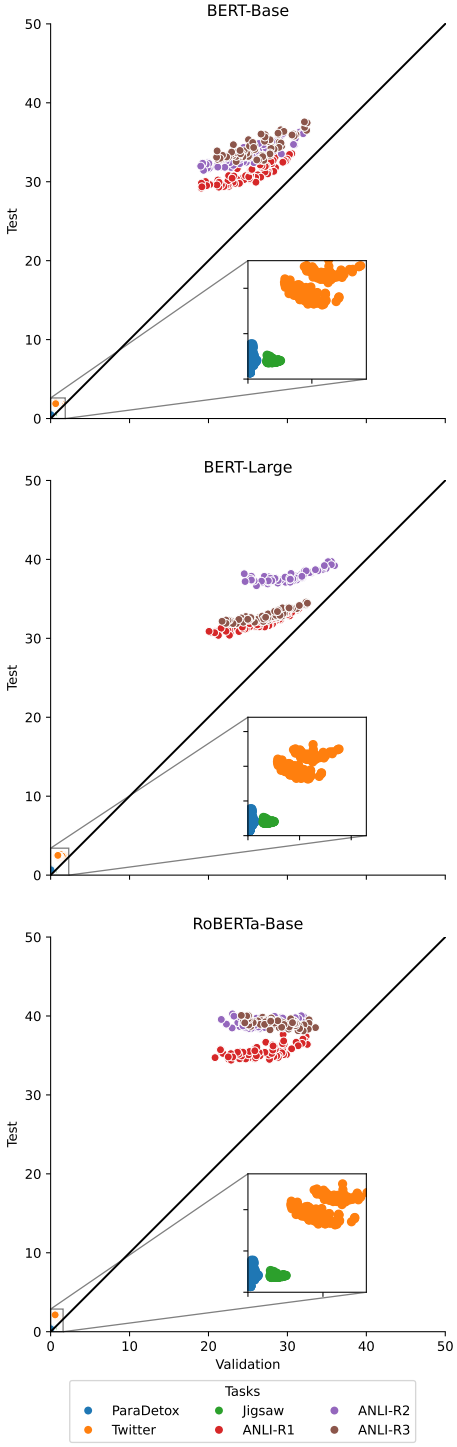
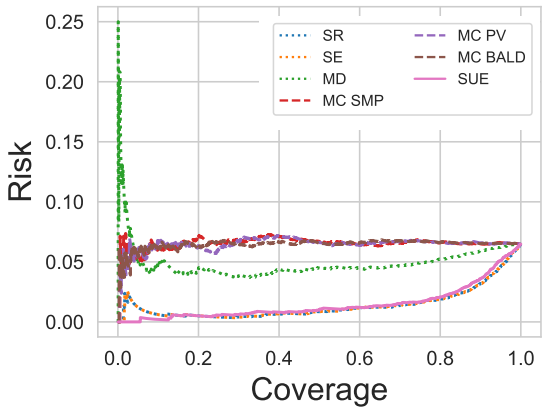
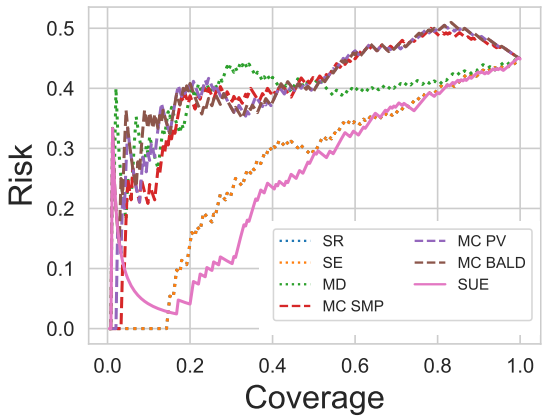


Figure 10: Relationship of test and development eAUROCC score when applying SUE.



(a) Twitter



(b) MRPC

Figure 11: Risk-Coverage Curve of two tasks.