Uncertainty Estimation and Reduction of Pre-trained Models for Text Regression

Yuxia Wang[◇] **Daniel Beck**[◇] **Timothy Baldwin**[◇] **Karin Verspoor**[†◇]
[◇] The University of Melbourne, Melbourne, Victoria, Australia
[†]RMIT University, Melbourne, Victoria, Australia
yuxiaw@student.unimelb.edu.au d.beck@unimelb.edu.au
tb@ldwin.net karin.verspoor@rmit.edu.au

Abstract

State-of-the-art classification and regression models are often not well calibrated, and cannot reliably provide uncertainty estimates, limiting their utility in safety-critical applications such as clinical decision-making. While recent work has focused on calibration of classifiers, there is almost no work in NLP on calibration in a regression setting. In this paper, we quantify the calibration of pre-trained language models for text regression, both intrinsically and extrinsically. We further apply uncertainty estimates to augment training data in low-resource domains. Our experiments on three regression tasks in both self-training and active-learning settings show that uncertainty estimation can be used to increase overall performance and enhance model generalization.

1 Introduction

Modern neural network models, particularly those based on pre-training and fine-tuning, have achieved impressive results across a broad spectrum of NLP tasks, in terms of evaluation metrics such as classification accuracy or F-score for classification tasks and mean squared error for regression tasks. However, the standard training regime fails to take model uncertainty into account, and tends to result in over-fitting and poor generalization, especially in limited training data situations.

In addition, these models have been empirically demonstrated to have poor calibration—the predictive probability does not reflect the true correctness likelihood, and they are generally over-confident when they make wrong predictions (Guo et al., 2017; Desai and Durrett, 2020; Jiang et al., 2020). Put differently, the models do not know what they don't know. This is particularly the case in low-resource settings. However, faithfully assessing the uncertainty of model predictions is as important as obtaining high accuracy in many safety-critical applications, such as autonomous driving or clinical decision support (Chen et al., 2020; Kendall and Gal, 2017; Davis et al., 2017). If models were able to more faithfully capture their lack of certainty when they make erroneous predictions, they could be used more reliably in critical decision-making contexts, and avoid catastrophic errors.

In the context of text regression, we aim to alleviate over-fitting and improve generalizability in low-resource settings by taking the uncertainty sourced from both the data and model into account. Specifically, we address: (1) data uncertainty by filtering noisy annotations from (either pseudo or gold) labeled data based on predictive confidence, preventing models from memorizing out-of-distribution examples; and (2) model uncertainty to accurately estimate both the target value and predictive confidence by uncertainty models, providing more reliable and interpretable predictions, meanwhile effectively supporting denoising in (1).

Uncertainty estimation has been extensively explored in the context of classification (Guo et al., 2017; Vaicenavicius et al., 2019; Desai and Durrett, 2020; Jiang et al., 2020), but is relatively unexplored for regression tasks, due to the complexities in dealing with a continuous target space. The output of a classifier passed through a $\mathrm{softmax}$ layer naturally provides a discrete probability distribution, while in a regression setting the output is a single numerical value.

We compare four well-studied techniques for uncertainty estimation, as applied to pre-trained language models (LMs): Gaussian processes (Shen et al., 2019; Camporeale and Carè, 2020),

Transactions of the Association for Computational Linguistics, vol. 10, pp. 680–696, 2022. https://doi.org/10.1162/tacl_a_00483
Action Editor: Dani Yogatama. Submission batch: 11/2021; Revision batch: 02/2022; Published 6/2022.

Bayesian linear regression (Hernández-Lobato and Adams, 2015), Bayes by backprop, and Monte Carlo (MC) dropout. To comprehensively assess uncertainty quality, we evaluate results intrinsically using various metrics, and extrinsically with several downstream experiments. Our analysis shows that predictions are highly uncertain and inaccurate in low-resource scenarios.

Two major types of uncertainty have been identified: *aleatoric uncertainty* captures noise inherent in the observations; and *epistemic uncertainty* accounts for uncertainty in the model, which can be explained away given enough data, compensating for limited knowledge (Kendall and Gal, 2017). In other words, uncertainty results primarily from noisy human annotations, insufficient labeled data, and out-of-domain text in practice (Glushkova et al., 2021). We therefore propose a simple method to filter noisy labels and select high-quality instances from an unlabeled data pool based on the predictive confidence, which on the one hand alleviates both aleatoric and epistemic uncertainty, and on the other hand, improves accuracy and generalization thanks to increased training data.

In this work, we explore how to estimate uncertainty in a regression setting with pre-trained language models, and evaluate estimation quality both intrinsically and extrinsically. Intrinsic uncertainty estimation provides the basis for our proposed data selection strategy: By filtering noise based on confidence thresholding, and mitigating exposure bias, our approach is shown to be effective at improving both performance and generalization in low-resource settings, in self-training, and active learning settings.

2 Background

We first review approaches for estimating the predictive uncertainty of deep neural networks (DNNs) in a regression setting, then methods for reducing uncertainty and improving generalization.

2.1 Uncertainty Estimation in DNNs

Bayesian Estimation Bayesian approaches provide a general framework for dealing with uncertainty estimation, for example, in the form of Gaussian processes (GPs: Camporeale and Carè, 2020; Shen et al., 2019) and Bayesian neural networks (Hernández-Lobato and Adams, 2015). However, prior work has either been based on hand-crafted features, or based on small-scale neural networks with only one or two hidden layers, which are far removed from modern pre-trained LMs. How to combine deterministic pre-trained LMs with Bayesian methods to achieve both high accuracy and accurate uncertainty estimation is an open problem, particularly in a regression setting.

While applying Bayesian estimation to all model parameters in large-scale LMs is theoretically possible, in practice it is prohibitively expensive in both model training and evaluation (Xue et al., 2021). Concretely, the true Bayesian posterior on the weights $P(\mathbf{w}|\mathcal{D})$ is generally approximated by variational inference, minimizing the KL divergence with a parameterized distribution $q(\mathbf{w}|\theta)$:

$$\begin{aligned}\theta^\star &= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta)\|P(\mathbf{w}|\mathcal{D})] \\ &= \arg\min_\theta \int q(\mathbf{w}|\theta)\log\frac{q(\mathbf{w}|\theta)}{P(\mathbf{w})P(\mathcal{D}|\mathbf{w})}\mathrm{d}\mathbf{w}\end{aligned}$$

Deriving uncertainty estimates by integrating over millions of model parameters, and initializing the prior distribution for each are both non-trivial. One simple strategy for combining them is Bayes by backprop (BBB: Blundell et al., 2015), whereby unbiased Monte Carlo gradients are minimized:

$$\sum_{i=1}^{n}\log q(\mathbf{w}^{(i)}|\theta) - \log P(\mathbf{w}^{(i)}) - \log P(\mathcal{D}|\mathbf{w}^{(i)})$$

where $\mathbf{w}^{(i)}$ denotes the ith Monte Carlo sample drawn from the variational posterior $q(\mathbf{w}^{(i)}|\theta)$.

Ensemble Estimation Another approach is to estimate uncertainty by ensemble, typically with MC-dropout (Gal and Ghahramani, 2016) and deep ensembles (Lakshminarayanan et al., 2017), which are agnostic to model structure.

MC-dropout casts dropout training in DNNs as approximate Bayesian inference in deep Gaussian processes. The predictive probability of the deep GP model (integrated with respect to the finite rank covariance function parameters $\mathbf{w}$) given precision parameter $\tau > 0$ is:

$$\begin{aligned}p(\mathbf{y}|\mathbf{x},\mathcal{D}) &= \int p(\mathbf{y}|\mathbf{x},\mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w} \\ p(\mathbf{y}|\mathbf{x},\mathbf{w}) &= \mathcal{N}(\mathbf{y};\hat{\mathbf{y}}(\mathbf{x},\mathbf{w}),\tau^{-1}\mathbf{I}_{\mathcal{D}})\end{aligned}$$

The dropout NNs are kept on during evaluation, without changing either the model or the optimization strategy. MC-dropout and its variants have been extensively used to estimate regression uncertainty due to their simplicity and scalability in implementation (Zelikman et al., 2020; Laves et al., 2020; Sicking et al., 2021).

The deep ensemble approach trains multiple copies of the variance networks from different network initializations to estimate predictive distributions. It operates similarly to sub-networks of MC dropout, but is computationally more expensive due to the need to train multiple models. Additionally, the need to split the training data into multiple folds to train different networks exacerbates overfitting in small-data scenarios. Given our specific focus on low-data scenarios, we focus exclusively on MC dropout in this paper.

The only work we are aware of for estimating uncertainty with transformers in a regression setting is Glushkova et al. (2021), who use ensemble estimation of uncertainty for machine translation quality evaluation, comparing the translated sentence with a reference translation. In contrast, we experiment in a cross-lingual setting, comparing a source sentence and its translation directly.

2.2 Selecting Clean Instances

To reduce the uncertainty from both data and model, we draw on approaches that can filter noisy labels from labeled data, and select clean instances from unlabeled data, thus eliminating aleatoric uncertainty, and reducing epistemic uncertainty due to the enhanced knowledge learned from the augmented data. In brief, we need a method to distinguish noisy and clean labels.

It has been shown that in data augmentation, self-training, and zero-shot learning, using the right sampling strategy is critical (Thakur et al., 2020; Wang et al., 2020c). However, previous work has mainly focused on label distribution balance, and lexical and semantic similarity, but not uncertainty.

In this work, we propose a simple method leveraging predictive confidence, to select high-quality instances, which is related to uncertainty-based sampling in active learning (Settles, 2009). However, most work in active learning has focused on classification rather than regression, either extracting the least probable or the most informative examples with large entropy (Settles and Craven, 2008; Pinsler et al., 2019; Radmard et al., 2021).

Our approach also has a similar flavor to self-paced curricular learning (Bengio et al., 2009; Kumar et al., 2010; Wan et al., 2020), in which the aim is to choose ‘‘hard’’ examples and gradually increase the difficulty of learning content, differing from the criteria in our setting—‘‘clean’’ ones.

According to a recent review of uncertainty estimation for DNNs (Abdar et al., 2020), there is little work on using aleatoric uncertainty for denoising and sampling in NLP tasks. The most relevant work is that by Miok et al. (2020), who aims to guide the annotation process for the binary classification task of hate speech detection.

3 Tasks and Notation

In this paper, we consider text regression across three separate tasks, and a total of 10 datasets.

Tasks **STS**: Semantic textual similarity assesses the degree of semantic equivalence between two pieces of text (Corley and Mihalcea, 2005). The aim is to predict a similarity score for a sentence pair $(S1, S2)$, generally in the range $[0, 5]$, where 0 indicates complete dissimilarity and 5 indicates equivalence in meaning. As an example:

S1: *Total minutes spent in timed codes: 10 mins.*
S2: *Total minutes spent in timed codes: 33 mins.*

might be labeled 4, as the two texts differ only in very specific content (underlined).

SA: Sentiment analysis rating involves predicting a sentiment score for a review S, in the range 1 (extremely negative) to 5 (extremely positive).

DA: Machine translation quality estimation, based on the direct assessment approach (Graham et al., 2017), aims to predict a normalised quality score for text pair $(S1, S2)$, where $S2$ is machine translated from $S1$. As such, it is similar to STS, but differs in that it is cross-lingual.

Notation and Assumptions Throughout this paper, raw examples, column vectors, and matrices are denoted in lower-case italics, bold, and upper-case italics, respectively (e.g., x, $\mathbf{x}$, and X). $\boldsymbol{\theta}_{encoder}$ and $\boldsymbol{\theta}_{reg}$ represent parameters of the encoder and task-specific regression layers, and $f(\boldsymbol{\theta}, \cdot)$ refers to the whole model. Take a dataset $\mathcal{D} = \{(x_1, y_1), (x_i, y_i), \cdots, (x_N, y_N)\}$, where (x_i, y_i) is the ith instance, $y_i \in \mathbb{R}$, and $\mathbf{x}_i = s(\boldsymbol{\theta}_{encoder}, x_i)$ is the hidden state of x_i. The

Dataset	Size (train, test, dev)	Range	Domain
STS-B (2017)	5749, 1379, 1500	[0, 5]	general
MedSTS (2018)	750, 318, —	[0, 5]	clinical
N2C2-STS (2019)	1642, 412, —	[0, 5]	clinical
BIOSSES (2017)	100, —, —	[0, 4]	biomedical
EBMSASS (2019)	700, 300, —	[1, 5]	biomedical
Yelp (2018)	7000, 1500, 1500,	[1,5]	product
PeerRead (2018)	713, 290, —	[1,5]	paper
WMT en-zh (2020)	7000, 1000, 1000	[0, 100]	high-resource
WMT ru-en (2020)	7000, 1000, 1000	[0, 100]	medium-resource
WMT si-en (2020)	7000, 1000, 1000	[0, 100]	low-resource

Table 1: STS/SA rating/QE-DA datasets. Train, Test, Dev Size = number of text pairs, range = label range. In practice, QE-DA is normalised by z-score.

loss function is the empirical risk of the mean square error (MSE): $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{\theta}, x_i) - y_i)^2$

Datasets We evaluate on different-sized datasets across various domains for STS and SA, and three same-sized datasets for DA, summarized in Table 1.

For STS, we use: (1) one large-scale general dataset, STS-B (Cer et al., 2017); (2) two small clinical data sets, MedSTS (Wang et al., 2018) and N2C2-STS (Wang et al., 2020a); and (3) two small biomedical data sets, BIOSSES (Soğancıoğlu et al., 2017) and EBMSASS (Hassanzadeh et al., 2019), each of which is 5-way annotated.

For SA, we use: (1) a large-scale product review dataset, Yelp (Sabnis, 2018); and (2) a small paper review rating dataset, PeerRead (Kang et al., 2018), augmented with 399 Spanish paper reviews (Keith et al., 2017) machine-translated into English.

For DA, we use the three language pairs from WMT2020 (Lucia et al., 2020), en-zh, ru-en, and si-en, corresponding to high-, medium-, and low-resource settings in terms of the source language.

4 Method

In this section, we first introduce approaches for estimating regression uncertainty based on pre-trained LMs, then propose a simple method to sample ''clean'' instances from unlabeled data to augment training data based on predictive uncertainty. The proposed methods can be applied equally in semi-supervised and unsupervised settings (including active learning and self-learning).

4.1 Bayesian Regression using LMs

We investigate two alternatives to combine pre-trained transformer LMs with Bayesian estimation, either in a pipeline approach, or end-to-end. Figure 1 provides an overview.

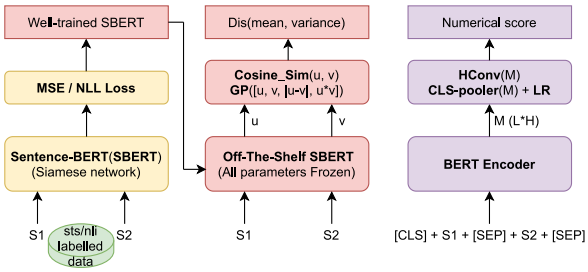

Figure 1: Overview of pipeline and end-to-end training workflow. left: SBERT is fine-tuned separately with STS/NLI labeled data using MSE/NLL loss; middle: well-trained SBERT provides off-the-shelf sentence embeddings to GP/Cosine similarity. End-to-end (right): under MC-dropout, keep dropout on in inference; in BBB, parameters of LR/HConv are stochastic variables.

Pipeline Training To estimate probability distributions for the regression task of document quality assessment, Shen et al. (2019) used a Gaussian process (GP) with Radial Basis Function (RBF) kernel function over hand-crafted features. We build off this in applying Bayesian linear regression and sparse GP regression to pre-trained sentence encoders, such as Sentence-BERT (SBERT; Reimers and Gurevych, 2019). For text input x, we generate $\mathbf{x} = s(\boldsymbol{\theta}_{encoder}, x) \in \mathbb{R}^d$. In this way, we leverage contextualized sentence representations, while avoiding the complexity of estimating uncertainty directly from a large-scale Bayesian neural network.

Bayesian Linear Regression: The prior distribution of a Bayesian linear layer with parameters $\mathbf{w}$ and b is set to be a Gaussian distribution:

$$\hat{y} = \mathbf{w}^\intercal \cdot \mathbf{x} + b + \varepsilon \quad (1)$$

$$\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I}); b \sim \mathcal{N}(\mu, 1) \quad (2)$$

where $\hat{y}$ is the approximated value and ε is the observation noise, which is assumed to be an independent and identically distributed random variable $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

Gaussian Processes (GPs) (Rusmassen and Williams, 2005) are a natural way to generalize the concept of a multivariate normal distribution determined by a mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, to describe a real-valued function. They provide a mathematically elegant framework for Bayesian inference and offer principled uncertainty estimates for regression problems with

a closed-form posterior (Leibfried et al., 2020). Given $(\mathbf{x}_i, y_i)$, $y_i = f(\mathbf{x}_i) + \varepsilon_i$, where $f(\cdot)$ is a real-valued function with input $\mathbf{x}_i$ that is sampled from a GP, and where ε_i are scalar independent and identically distributed random variables corresponding to observation noise.

The prior on data generation can be encapsulated in the distribution of $f(\cdot)$. We assume that $f(\cdot)$ is distributed according to a GP, that is,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))) \tag{3}$$

where $m(\mathbf{x})$ is a mean function, and $k(\mathbf{x}, \mathbf{x}')$ is a covariance or kernel function, corresponding to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of a multivariate normal distribution. Following common practice, we fix the mean function to zero, and use a RBF as the kernel function (Preoţiuc-Pietro and Cohn, 2013; Beck et al., 2014; Bitvai and Cohn, 2015; Shen et al., 2019).

Computing the exact posterior requires the storage and inversion of an $(N \times N)$ matrix, which is quadratic in the amount of training data N and has cubic computational complexity, both of which are infeasible for large datasets. Thus we use sparse GPs, which approximate an exact GP by using a small set of latent inducing points (Titsias, 2009), learned by variational inference.

End-to-end Training Rather than pre-training a LM and task-specific model separately, Xue et al. (2021) jointly trained them by only applying Bayesian estimation to a subset of the model parameters. This requires training entirely from scratch, while we seek to leverage pre-trained LMs. We apply Bayesian inference to task-specific layers, keeping parameters of the LM deterministic and making task-specialised parameters stochastic during fine-tuning. Importantly, being deterministic is not equivalent to being frozen: Parameters are updated as in non-Bayesian optimization, rather than kept fixed during back-propagation.

To increase randomness, we evaluate on two task-specific networks with more stochastic parameters than a single-layer linear regression network used in Pipeline Training, as detailed below.

Bayesian Two-layer MLP: The linear regression layers take the hidden state $\mathbf{h} \in \mathbb{R}^d$, through a two-layer MLP with $\tanh$ activation function:

$$\mathbf{h}' = \tanh(\mathbf{W}\mathbf{h} + \mathbf{b}); \hat{y} = \mathbf{w}^T\mathbf{h}' + b \tag{4}$$

where $\hat{y}$ is the approximated score, and $\mathbf{W} \in \mathbb{R}^{d\times d}$, $\mathbf{b}, \mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are trainable parameters.

Bayesian Hierarchical Convolution: Drawing on the finding that a hierarchical convolution neural network (HConv) is effective in low-resource settings (Wang et al., 2020b), and that increasing the capacity of task-specific layers can boost performance (Chung et al., 2020), we train a large-capacity network as follows. HConv is structured as a two-layer convolutional network, with kernel size $k = 2, 3, 4$ in the first layer and $k = 2$ in the second (Wang et al., 2020b). The prior distributions of the weights and bias are based on Eq. (2) for Bayesian inference, and the inference method follows Bayes by Backprop (Blundell et al., 2015).

4.2 Predictive Uncertainty-based Sampling

Given a pre-trained uncertainty model $f(\boldsymbol{\theta}, \cdot)$, and a (large-scale) unlabeled data pool $\mathcal{D}_u = \{x_1, x_2, \cdots, x_i, \cdots, x_U\}$, the distribution of the predicted y_i for input x_i is:

$$P(y_i) = f_{\boldsymbol{\theta}}(x_i) \sim \mathcal{N}(\mu_i, \sigma_i) \tag{5}$$

where μ_i and σ_i are the mean and standard deviation of the normal distribution of y_i.

Our aim is to sample a subset $\mathcal{D}'_u$ from $\mathcal{D}_u$ in which the uncertainty model is expected to be sufficiently confident in predicting $\mathcal{D}'_u$, that is have a confidence interval as narrow as possible under a given confidence level. For example, under 99% confidence, the confidence interval $[\mu_i - 2.58\sigma_i, \mu_i + 2.58\sigma_i]$ is expected to be narrow. Put differently, the distribution is concentrated around the mean with small standard deviation.

Based on this, we propose a simple instance selection method based on predictive uncertainty. For each instance x_i in $\mathcal{D}_u$, if $\sigma_i < \tau$, select x_i; $\mathcal{D}'_u \leftarrow x_i$. The threshold τ is a global hyperparameter tuned over the validation set, or in the case of self-training and active learning, using a heuristic strategy.[1]

The strategy is based on the observation that the model can generally predict precisely for

[1]We also experimented with a strategy for tuning τ based on the principle of discarding the majority so that remaining examples are as clean as possible. Specifically, we set τ to the marginal value corresponding to the left boundary of the peak of the std probability distribution, but found little difference in results, so omit it from the paper.

instances of extreme polarity, such as labels in the ranges $[0, 1]$ and $[4, 5]$ for STS. We posit that cases whose predictive uncertainty is at the same level as these well-predicted examples are also predicted accurately. Formally, after inference, the unlabeled data pool is $\mathcal{D}_u = \{(x_i, \mu_i, \sigma_i)\}$, $i \in [1, U]$, where U is the number of unlabeled instances. The standard deviation of all well-predicted examples can be vectorized as $\boldsymbol{\sigma} = [\sigma_i]$, where σ_i is the std whose μ_i is at an extremum, such as $0 \leq \mu_i \leq 1$ or $4 \leq \mu_i \leq 5$ for STS. We then set $\tau = \text{mean}(\boldsymbol{\sigma})$.

5 Uncertainty Evaluation Metrics

Evaluating uncertainty estimates of predictions is challenging in a regression setting, as the ''ground truth'' uncertainty is usually not available (Lakshminarayanan et al., 2017). To evaluate model predictions, we consider four metrics.

Pearson Correlation: It is vital to assess the predictive accuracy of the system, regardless of the uncertainty estimate. We use Pearson correlation r to evaluate the correlation between the system's average predictions and ground truth quality scores.

Calibration Error (CAL): One way to understand if models can be trusted is by analysing whether they are calibrated. Gneiting et al. (2007) defined calibration in a regression setting as the asymptotic consistency between the probabilistic forecasts F_i and the true data-generating distributions G_i, with the index i referring to each example.

Practically, F_i is the cumulative probability distribution $P(Y \leq y_i)$, G_i is generally estimated by empirical distribution functions based on the observations only. So calibration measures if the predictive confidence estimates are aligned with the empirical correctness likelihoods. Given a confidence level p_j, the empirical accuracy is calculated:

$$\hat{p_j} = \frac{\sum_{i=1}^{n} \mathbb{I}\left\{y_i \leq {F_i}^{-1}(p_j)\right\}}{n}$$

where ${F_i}^{-1}$ is used to denote the quantile function ${F_i}^{-1}(p) = \inf\{y : p \leq F_i(y)\}$, that is mapping from [0,1] $\rightarrow$ Y. The expected calibration error $\text{cal} = \sum_{j=1}^{m} w_j \cdot (p_j - \hat{p_j})^2$, with m confidence levels $0 \leq p_1 < \cdots < p_m \leq 1$, is the distance of predictive confidence away from the empirical accuracy.

Negative Log-Probability Density (NLPD) complements CAL's equal treatment to over- and under-confidence. It penalises over-confidence more strongly through logarithmic scaling: $L_{\text{NLPD}} = -\frac{1}{n}\sum_{i=1}^{n} \log p(y_i = t_i|\mathbf{x}_i))$, favouring under-confident ones. In Gaussian predictive distributions with mean m_i and variance v_i, the NLPD loss incurred for predicting at input x_i with true associated target t_i is given by:

$$L_{\text{NLPD}} = \frac{1}{2n}\sum_{i=1}^{n}\left[\log v_i + \frac{(t_i - m_i)^2}{v_i}\right]$$

Sharpness (SHP): The metrics above do not account for the concentration of the predictive distributions, which generally favours predictors that produce wide and uninformative confidence intervals. To guarantee useful uncertainty estimation, confidence intervals should not only be calibrated, but also sharp and ''tight'' around the predicted value. The numerical width of prediction intervals (Gneiting et al., 2007; Song et al., 2019) and the mean of variance (Kuleshov et al., 2018; Zelikman et al., 2020) are often used to quantify sharpness. We apply the latter in our work, with a lower score implying higher sharpness.

To interpret mixed results, for example when a model attains the best sharpness but with infinitely large NLPD, we suggest that Pearson correlation (r) has primacy, followed by CAL and NLPD, then SHP. That is, when models have comparable r, the comparison of CAL/NLPD is more meaningful, and if those are also similar, SHP should be considered; otherwise, it's largely meaningless.

6 Evaluation of Uncertainty Estimation

We expect that the incorporation of uncertainty estimation should not harm predictive performance compared to point estimation without uncertainty, in both in- and out-of-domain scenarios. Additionally, uncertainty estimates should reflect ''what the model does not know'', making it possible to determine whether a prediction can be trusted based on the output distribution. This is quantified intrinsically with CAL and NLPD (the lower, the better), and extrinsically via instance selection in Section 7.

6.1 Experimental Setup

Pipeline Training: We use SBERT as an off-the-shelf sentence encoder. We fine-tune SBERT

separately over each STS corpus based on the pre-trained `bert-base-nli-mean-tokens`, using the same configuration as the original paper (4 epochs with training batch size of 16). For the cross-lingual DA task, we use *distiluse-base-multilingual-cased-v1*.

To represent a sentence pair $(S1, S2)$ using SBERT, we use the concatenation of the embeddings $u \oplus v$, along with their absolute difference $|u - v|$ and element-wise multiplication $v \times t$. ''SBERT Bayesian LR'' and ''SBERT Sparse GP Regression'' indicates that features are fed into Bayesian LR and sparse GP regression, respectively, implemented in pyro.[2]

End-to-End Training: We apply pre-trained BERT as the LM encoder (Devlin et al., 2019), using `bert-base-uncased` for monolingual tasks and `bert-base-multilingual-cased` for cross-lingual tasks. The input format is `[CLS] S1 [SEP] S2 [SEP]` for text pair $(S1, S2)$, and `[CLS] S [SEP]` for a single text S. BERT Bayesian LR and BERT Bayesian ConvLR denote task-specific networks based on a two-layer MLP and HConv, respectively, implemented based on the *Huggingface Transformer* framework and blitz for BBB estimation (Esposito, 2020).

MC-Dropout: We apply MC-dropout to base models BERT LR and BERT ConvLR, with dropout rate = 0.1 and 30 iterations of sampling.[3]

Point Estimation: In addition to the uncertainty estimation approaches, we also compare with four non-Bayesian methods: (1) cosine similarity; (2) optimization of deterministic LR with SBERT (SBERT LR); (3) fine-tuned BERT LR; and (4) fine-tuned BERT ConvLR.

Training Configuration: The maximum sequence length is set to 128 for STS and DA, and 256 for SA. The learning rate (lr), training batch size, and training epochs are optimized over the validation set. In the situation that a validation set is not available (i.e., EBMSASS and MedSTS), we provisionally split the training data into 80%:20% training:dev data, and tune hyperparameters over the dev data. We then retrain the model over the full training dataset, and evaluate on the test set. Tuned hyperparameter settings of the pipeline are shown in Table 3. End-to-end is based on grid-searching over [8, 16, 32] $\times$ [1e-5, 2e-5] $\times$ [1, 2, 3, $\cdots$ 10] for batch size, lr, and epochs, respectively. Generally, the best setting is batch size = 16, lr = 2e-5, and epochs = 3, although BERT ConvLR based on BBB requires more epochs to converge. Further details of the training regimen and hyperparameter settings are provided in our Github repository.[4]

[2] `https://pyro.ai/`.

[3] No significant difference was observed when sampling 20, 30, 40, or 50 times, so we report only on 30.

6.2 Sentence-Pair STS

In this section, we compare the various uncertainty estimation approaches from Section 4.1 over STS, in terms of correlation and the metrics for uncertainty estimation, aiming to empirically establish:

1. Which uncertainty estimation strategy is most accurate, most calibrated, and sharpest?
2. Which method performs best in out-of-domain settings?

6.2.1 In-Domain Performance

To observe the influence of data size and domain distribution on uncertainty estimation, we experiment over the large-scale general-domain STS-B, in addition to the smaller-scale domain-specific MedSTS (clinical domain) and EBMSASS (biomedical domain) datasets. There are three main findings from the results in Table 2.

Uncertainty models do not degrade accuracy. With SBERT, GP-based models have higher correlation than either cosine similarity or LR. In the case of BERT, estimation by MC-dropout is competitive with corresponding point estimates. Thus, they have comparable raw performance, in addition to providing uncertainty estimates.

End-to-end training based on BERT results in higher correlation and narrower confidence intervals, but poorer calibration and NLPD. Results over the three datasets show that end-to-end training based on BERT overall performs much better than pipeline training using SBERT, but BERT-based models are poorly calibrated compared to SBERT-based Bayesian linear regression and sparse GP regression using fixed sentence features (as can be seen in the higher NLPD numbers for BERT-based models). This is consistent with prior work (Guo et al., 2017).

MC-dropout is superior to BBB inference, and sparse GP regression performs better than SBERT Bayesian LR, regardless of data size

[4] `https://github.com/yuxiaw/Uncertainty-regression`.

	STS-B test				EBMSASS test				MedSTS test				Yelp test			
	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓
SBERT Cosine similarity	0.842	N/A	N/A	N/A	0.773	N/A	N/A	N/A	0.784	N/A	N/A	N/A	—	N/A	N/A	N/A
SBERT LR	0.835	N/A	N/A	N/A	0.743	N/A	N/A	N/A	0.776	N/A	N/A	N/A	0.666	N/A	N/A	N/A
SBERT Bayesian LR	0.810	**0.046**	0.648	1.632	0.688	0.443	1.095	2.156	0.740	0.101	0.801	2.092	0.671	**0.019**	**0.447**	0.753
SBERT Sparse GP Regression	0.847	0.065	**0.614**	1.621	0.788	0.195	0.541	1.627	0.781	**0.073**	**0.499**	1.453	0.689	0.049	0.573	1.507
BERT LR	**0.868**	N/A	N/A	N/A	0.914	N/A	N/A	N/A	0.858	N/A	N/A	N/A	0.826	N/A	N/A	N/A
BERT ConvLR	0.855	N/A	N/A	N/A	**0.922**	N/A	N/A	N/A	0.846	N/A	N/A	N/A	0.822	N/A	N/A	N/A
BERT Bayesian LR (BBB)	0.848	0.521	$+\infty$	**0.005**	0.914	0.669	1177.2	**0.005**	0.848	0.514	6594.3	**0.006**	**0.827**	0.531	3908.6	**0.083**
BERT Bayesian ConvLR (BBB)	0.849	0.495	2061.0	0.015	0.898	0.618	327.3	0.010	0.835	0.506	1037.2	0.017	0.797	1.513	119.2	0.089
BERT LR MC dropout	0.868	0.181	4.659	0.215	0.921	**0.054**	**0.036**	0.140	**0.859**	0.163	4.118	0.168	0.827	0.267	7.285	0.153
BERT ConvLR MC dropout	0.855	0.202	5.830	0.209	0.922	0.093	2.137	0.085	0.852	0.219	6.402	0.146	0.823	0.291	8.214	0.150

Table 2: Correlation r and uncertainty prediction quality metrics (CAL, NLPD, and SHP) on three STS datasets (STS-B, EBMSASS, and MedSTS) and a SA rating dataset (Yelp), with SBERT and BERT sentence embeddings with various task-specific layers: Cosine similarity = calculate cosine similarity between vectors representing S1 and S2; LR = single-layer linear regression; Bayesian LR = Bayesian linear regression; and Sparse GP Regression = Sparse Gaussian process regression. N/A indicates that the method doesn't produce an uncertainty estimate to apply the given metric to.

	LR		Bayes LR		GP Reg	
	lr	epoch	lr	epoch	lr	epoch
STS-B	0.1	100	0.01	2500	0.1	25
EBMSASS	0.1	15	0.01	10000	0.1	25
MedSTS	0.1	100	0.01	8500	0.1	25
Yelp	0.1	600	0.01	2500	0.1	25
en-zh	0.1	50	0.03	300	0.1	200
ru-en	0.1	40	0.03	400	0.1	200
si-en	0.1	199	0.03	300	0.1	1000

Table 3: Learning rate (lr) and training epochs (epoch) for pipeline training based on SBERT.

and domain. Under both BERT LR and ConvLR, MC-dropout achieves higher or equal correlation, and much lower CAL and NLPD than BBB in end-to-end training. Among methods based on SBERT, sparse GP regression requires many fewer iterations to converge, and outperforms Bayesian LR in correlation and NLPD, and is comparable for CAL and SHP.

6.2.2 Out-of-Domain Performance

Apart from in-domain evaluation, out-of-domain performance is also an important concern. We expect that a model trained on domain A will generate more uncertain predictions on domain B, with lower correlation, larger CAL and NLPD, and a wider confidence interval (Lakshminarayanan et al., 2017). Given two models trained on domain A with similar point-estimate performance on domain B, that is competitive r, the model with the lower NLPD is arguably the better model, as this indicates that the model gives sharper distributions when the prediction is correct, and flatter ones when wrong.

Using models fine-tuned over the general-domain STS-B, we evaluate on the biomedical EBMSASS and clinical MedSTS test sets. In contrast with the results in Table 2, in which models have been fine-tuned with in-domain labeled data, Table 4 shows a steep decline in r of more than 10 points on average for EBMSASS, and 7 for MedSTS. Meanwhile, both CAL and NLPD increase by a large margin.

MC-dropout is not always best. Interestingly, we find that BERT Bayesian LR performs well in this setting, obtaining the highest correlation and smallest SHP on EBMSASS and PeerRead. This suggests that BERT Bayesian LR has better generalizability over these two domains, but the substantially higher NLPD also reveals that its predictions are over-confident. By and large, MC-dropout stably offers accurate and calibrated predictions in out-of-domain settings. ConvLR in particular outperforms Bayesian inference across all metrics.

BERT ConvLR tends to be inferior to BERT LR in the out-of-domain setting. We speculate this is because of its smaller capacity to memorize task-specific knowledge, as eight layers of the BERT encoder are frozen in BERT ConvLR.

6.3 Single-sentence Sentiment Rating

We perform in-domain SA evaluation on Yelp, and out-of-domain evaluation by applying the fine-tuned Yelp model to PeerRead test data. We find:

Fine-tuned sentence embeddings are vital to the performance of pipeline uncertainty estimation. As shown in Table 2, performance over Yelp, EBMSASS, and MedSTS based on SBERT

	EBMSASS test				MedSTS test				PeerRead test			
	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓
SBERT Cosine similarity	0.716	N/A	N/A	N/A	0.731	N/A	N/A	N/A	—	N/A	N/A	N/A
SBERT LR	0.696	N/A	N/A	N/A	0.718	N/A	N/A	N/A	0.256	N/A	N/A	N/A
SBERT Bayesian LR	0.684	**0.091**	**0.400**	1.325	0.672	**0.038**	**0.568**	1.506	0.241	0.116	1.018	1.245
SBERT Sparse GP Regression	0.726	0.211	0.586	1.609	0.723	0.129	0.634	1.604	0.427	**0.021**	**0.771**	1.339
BERT LR	0.838	N/A	N/A	N/A	0.786	N/A	N/A	N/A	0.669	N/A	N/A	N/A
BERT ConvLR	0.806	N/A	N/A	N/A	0.776	N/A	N/A	N/A	0.627	N/A	N/A	N/A
BERT Bayesian LR	**0.867**	0.625	5165	**0.005**	0.768	0.619	11081	**0.005**	**0.694**	0.522	7606.	**0.009**
BERT Bayesian ConvLR	0.811	0.714	1043.	0.011	0.770	0.523	1527.	0.017	0.608	0.990	189.0	0.086
BERT LR MC dropout	0.838	0.280	3.517	0.137	**0.795**	0.199	5.060	0.188	0.676	0.400	21.75	0.160
BERT ConvLR MC dropout	0.814	0.194	4.649	0.153	0.788	0.240	8.447	0.158	0.635	0.456	36.37	0.138

Table 4: Results on EBMSASS, MedSTS and PeerRead test sets using models trained on general-purpose STS-B and Yelp for STS and SA, respectively.

	en-zh test				ru-en test				si-en test			
	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓
SBERT Cosine similarity	0.115	N/A	N/A	N/A	0.428	N/A	N/A	N/A	0.097	N/A	N/A	N/A
SBERT LR	0.270	N/A	N/A	N/A	0.616	N/A	N/A	N/A	0.397	N/A	N/A	N/A
SBERT Bayesian LR	0.280	**0.025**	0.155	0.908	0.625	0.013	0.223	0.771	0.371	0.013	0.193	0.934
SBERT Sparse GP Regression	0.384	0.026	**0.143**	0.892	0.626	**0.007**	**0.207**	0.776	0.366	**0.010**	**0.191**	0.931
BERT LR	0.395	N/A	N/A	N/A	0.621	N/A	N/A	N/A	0.504	N/A	N/A	N/A
BERT ConvLR	0.436	N/A	N/A	N/A	0.641	N/A	N/A	N/A	0.524	N/A	N/A	N/A
BERT Bayesian LR	0.385	0.726	11600	**0.005**	0.644	0.515	11666	**0.005**	0.506	0.568	10971	**0.005**
BERT Bayesian ConvLR	0.378	1.780	683.7	0.066	0.609	1.775	723.4	0.069	0.503	1.758	638.5	0.059
BERT LR MC dropout	0.407	0.250	9.216	0.190	0.637	0.315	17.00	0.126	0.527	0.178	6.578	0.200
BERT ConvLR MC dropout	**0.441**	0.268	13.33	0.127	**0.649**	0.333	22.28	0.106	**0.530**	0.275	10.19	0.133

Table 5: Results for DA-style quality estimation over the three WMT language pairs.

is substantially worse than with BERT. We speculate this is due to poor feature representations. That is, on the STS task, we continue to fine-tune sentence embeddings over each STS dataset. As a result of being unable to fine-tune SBERT on SA (as there is no paired data), the representations for Yelp are pre-trained using SNLI only, which is neither task- nor domain-specific. Compared with the similarly sized STS-B where embeddings are fine-tuned, the performance gap for Yelp between SBERT and BERT is more than 0.15, but less than 0.02 for STS-B. Equally, though we fine-tune SBERT for EBMSASS and MedSTS, each has fewer than 1k training instances. Poor domain-specific sentence embeddings result in gaps of 0.15 and 0.07.

Meanwhile, for SBERT in the upper half of Table 4, the out-of-domain correlation on PeerRead is extremely poor; the gap of 6 points on EBMSASS and MedSTS relative to in-domain results (0.78 in Table 2) further confirms our hypothesis.

LR outperforms ConvLR in out-of-domain SA. In both point and Bayesian estimates, ConvLR performs better than LR (Table 4), similar to STS.

6.4 Cross-lingual Sentence-pair DA

We evaluate on machine translation quality estimation (QE) over three language pairs using DA, using 7,000 training instances in each case. The results are shown in Table 5. We first observe that using embeddings directly from pretrained SBERT with cosine similarity underperforms other methods that involve fine-tuning.

Traditional Bayesian LR and GP models achieve results competitive with deep uncertainty models when the input sentence embedding is expressive enough, and with smaller CAL and NLPD. Related uncertainty prediction work (Glushkova et al., 2021) argued that GPs are not competitive or easy to integrate with current neural architectures. In contrast, our results demonstrate that GPs can achieve comparable results to deep neural networks, while also being better calibrated.

	N2C2-STS test			MedSTS test			PeerRead test		
	r_1 / r_2 ↑	CAL ↓	NLPD ↓	r_1 / r_2 ↑	CAL ↓	NLPD ↓	r_1 / r_2 ↑	CAL ↓	NLPD ↓
Semi-supervised:									
BERT LR	0.853 / 0.857	**0.384**	**6.571**	0.858 / 0.859	**0.158**	**3.903**	0.686 / 0.686	0.370	**15.95**
+$\mathcal{D}_u$	0.861 / 0.862	0.511	9.232	0.860 / 0.861	0.224	5.267	0.655 / 0.656	0.394	19.26
+$\mathcal{D}'_u$	0.860 / 0.864	0.493	8.476	**0.863 / 0.866**	0.181	4.758	**0.720 / 0.720**	**0.340**	19.89
BERT ConvLR	0.874 / 0.875	0.509	11.51	0.846 / 0.853	0.201	5.968	0.691 / 0.692	0.346	16.98
+$\mathcal{D}_u$	**0.875** / 0.876	0.522	13.50	0.846 / 0.855	0.215	6.403	0.671 / 0.683	0.453	25.50
+$\mathcal{D}'_u$	**0.875 / 0.879**	0.535	11.44	0.857 / 0.864	0.222	6.129	0.699 / 0.697	0.374	21.78
Zero-shot:									
BERT LR	0.682 / 0.663	**0.568**	**17.08**	0.786 / 0.795	**0.199**	**5.060**	0.669 / 0.676	**0.400**	**21.75**
+ $\mathcal{D}_u$	0.687 / 0.673	0.624	40.10	0.796 / 0.797	0.266	11.94	0.023 / 0.006	1.728	387.3
+ $\mathcal{D}'_u$	0.743 / 0.729	0.630	23.67	0.793 / 0.792	0.296	8.907	**0.678 / 0.675**	0.495	52.72
BERT ConvLR	0.728 / 0.722	0.612	21.06	0.776 / 0.788	0.240	8.447	0.627 / 0.635	0.456	36.37
+ $\mathcal{D}_u$	0.746 / 0.737	0.653	47.68	0.790 / 0.794	0.332	16.45	0.138 / 0.119	1.748	546.1
+ $\mathcal{D}'_u$	**0.763** / 0.**748**	0.628	40.32	**0.809 / 0.810**	0.303	15.26	0.656 / 0.659	0.483	57.77

Table 6: Results on three low-resource regression datasets: clinical STS: MedSTS, N2C2-STS, and PeerRead. r_1 are the results without MC-dropout, while r_2, CAL, and NLPD are based on applying 30 iterations of MC-dropout. There are two setups: (1) semi-supervised (upper half) = domain gold-labeled data is available; and (2) zero-shot (bottom half). In each case, $\mathcal{D}_u$ = unlabeled data pool selected based on the model probability; $\mathcal{D}'_u$ = unlabeled data pool selected based on hyperparameter τ over the predicted std; and row 1,4,7,10 = baseline for each setting.

ConvLR consistently outperforms LR for BERT-based models. In the cross-lingual scenario, SBERT models have smaller CAL and NLPD, and larger SHP, analogous to the monolingual setting.

7 Instance Selection Through Uncertainty

In self-training, a model is first trained using labeled data, then used to predict labels for unlabeled data instances. Instances with higher-probability predictions are then adopted as pseudo-labels, and used to re-train the model in conjunction with the labeled training data. Active learning is similar, expect that instances are selected for explicit human labelling rather than pseudo-labeled, often based on estimates of model confidence or uncertainty. In both tasks, accurate estimation of labelling (un)certainty is critical.

In this section, we evaluate the uncertainty-based instance selection method from Section 4.2 in the settings of self-training and active learning, over the tasks of STS, SA rating, and cross-lingual DA.

7.1 Self-training STS and SA

In self-training, we experiment in both semi-supervised (limited gold-standard training data) and zero-shot scenarios, over three low-resource datasets: MedSTS, N2C2-STS, and PeerRead.

Experimental Setup: As we require high correlation to ensure high-quality pseudo-labels, and lower CAL and NLPD to guarantee that predictions are neither over- nor under-confident, we employ MC-dropout over LR and ConvLR. Additionally, to alleviate domain data sparsity, we first fine-tune the regressor on two general datasets—STS-B for STS and Yelp for SA (general-purpose STS/SA)—also providing the proxy for the zero-shot setting. We continue to fine-tune on domain training data in the semi-supervised scenario, and predict (μ, σ) for $\mathcal{D}_u$ by applying dropout 30 times. All results in Table 6 are obtained using train batch size = 16, learning rate = 2e-5, and training epochs = 3.

Unlabeled Data Pool: For clinical STS, we extract sentences from MIMIC-III covering topics of medication, diagnosis, follow-up instructions, and test, then synthetically balance across each unit score interval, resulting in 1,534 sentence pairs, which we denote as $\mathcal{D}_u$. For PeerRead, we use 1,014 reviews from ICLR 2017 without labels as $\mathcal{D}_u$. To expand $\mathcal{D}_u$ in the zero-shot setting, we remove the gold-standard labels and integrate the resulting unlabeled data into $\mathcal{D}_u$.

Results and Analysis: As seen in Table 6, semi-supervision improves correlation, at the cost of being more uncertain and miscalibrated, with larger CAL and NLPD. Predictive confidence threshold selection can further improve the accuracy. It also effectively calibrates the model, resulting in much lower CAL and NLPD, compared with directly incorporating unlabeled data (''+$\mathcal{D}_u$'').

In the zero-shot setting, CAL and NLPD increase for all tasks under both LR and ConvLR with $\mathcal{D}_u$, making predictions less reliable, especially for PeerRead where the model totally collapses. This matches our intuition that the distribution of the pseudo-labeled data differs from the true distribution, and that learning from this data impedes the model. This problem is alleviated by retaining only the highly confident subset $\mathcal{D}'_u$, as its distribution is closer to the gold-standard for well-calibrated models. This is also consistent with the observation that CAL and NLPD in the zero-shot setting are much larger than in the semi-supervised setting, as the latter benefits from the guidance of the gold-standard distribution.

Note that if we merely assess the model with Pearson correlation as in most previous work, we can only observe the improvement due to data augmentation, neglecting the risk of the model being more miscalibrated, and producing less reliable predictions. Further, CAL and NLPD are useful metrics to evaluate the effectiveness of the data sampling strategy used in self-training.

7.2 Cross-lingual DA

We evaluate self-training and active-learning on DA-based machine translation quality estimation using BERT LR.

Experimental Setup: We use three language pairs: WMT 2020 DA en-zh, ru-en, and si-en, in each case splitting the original 7k training instances into a training set $\mathcal{D}$ of 3k instances and 4k unlabeled data pool $\mathcal{D}_u$, keeping the original validation and test sets. The lr is set to 2e-5, and training epochs and batch size are tuned by grid search over the validation set based on the range [1,2,3,4,5] $\times$ [16, 32]. Other settings follow STS and SA above, but without a general-purpose base model. As a baseline, we use $\mathcal{D}$ fine-tuned on the validation set, and evaluate the best configuration on test.

	en-zh (high)		ru-en (medium)		si-en (low)	
	$r_{dev}\uparrow$	$r_{test}\uparrow$	$r_{dev}\uparrow$	$r_{test}\uparrow$	$r_{dev}\uparrow$	$r_{test}\uparrow$
Baseline	0.407	0.374	0.592	0.599	0.427	0.478
+ pseudo $\mathcal{D}_u$	0.434	0.400	0.604	0.619	0.449	0.488
+ $\mathcal{D}'_u$	0.438	0.404	0.606	0.603	0.443	0.482
+ $\mathcal{D}'_u \cup \mathcal{D}'_a$	0.445	**0.422**	**0.615**	**0.628**	**0.466**	0.496
+ gold $\mathcal{D}_u$	**0.453**	0.395	0.600	0.621	**0.466**	**0.504**

Table 7: Results for DA-based quality estimation in WMT 2020 (dev/test) for three language pairs: en-zh, ru-en and si-en. Baseline = training with 3,000 gold-labeled instances. Row ''+ $\mathcal{D}'_u \cup \mathcal{D}'_a$'' is active learning.

Results and Analysis: As shown in Table 7, directly incorporating pseudo $\mathcal{D}_u$ substantially outperforms baselines for all three language pairs. This differs from the results for STS and SA in the semi-supervised setting, but is consistent with the results in the zero-shot setting. It indicates that a high-performance model requires high-quality data to further gain improvements; lower-quality models are more tolerant to lower data quality.

We select the most confident 1,904, 1,985, and 2,462 instances with $\tau = 0.15$, 0.13 and 0.19 for en-zh, ru-en and si-en, respectively. Equal or higher performance is achieved when this subset of instances is added to the training data, as compared to the complete $\mathcal{D}_u$.

Simulating active learning, we also explore the annotation of $\mathcal{D}_u - \mathcal{D}'_u$ with human gold scores, i.e. $\mathcal{D}'_a$. The results show that with $\mathcal{D}'_u \cup \mathcal{D}'_a$, our model achieves results competitive with using all of $\mathcal{D}_u$ with gold labels. This reveals that it is not necessary to annotate the entire dataset, but we can focus on the subset where the model is not confident. In this way, data annotation is more efficient, and models generalize better over unseen data.

8 Analysis

In this section, we conduct further analysis to better understand the results of the experiments.

Qualitative Comparison: In both in-domain and out-of-domain evaluation, end-to-end training based on BERT, particularly BBB estimation, obtains much larger NLPD than pipeline training based on SBERT, especially GP regression. We speculate that end-to-end uncertainty models are confident for both correct and incorrect

	STS-B test				EBMSASS test				MedSTS test				Yelp test			
	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓	$r\uparrow$	CAL ↓	NLPD ↓	SHP↓
simCSE Cosine	0.833	N/A	N/A	N/A	0.700	N/A	N/A	N/A	0.696	N/A	N/A	N/A	—	N/A	N/A	N/A
simCSE LR	0.849	N/A	N/A	N/A	0.703	N/A	N/A	N/A	0.675	N/A	N/A	N/A	0.688	N/A	N/A	N/A
simCSE Bayes LR	0.850	0.051	0.381	0.891	0.738	0.048	0.102	0.900	0.693	0.002	0.295	0.885	0.668	0.005	0.377	0.846
simCSE Sparse GP	0.853	0.002	0.368	0.960	0.757	0.210	0.218	0.962	0.694	0.034	0.346	0.960	0.681	0.004	0.360	0.880

Table 8: Pipeline model results for the simCSE sentence encoder (Gao et al., 2021).

	SBERT Sparse GP	BERT Bayesian LR (BBB)
Incorrect Predictions: **S1:** *You will want to clean the area first.* **S2:** *You will also want to remove the seeds.* **Gold score = 0**		
Prediction:	2.22 ± 1.62	1.95 ± 0.0037
Correct Predictions: **S1:** *He was referring to ..., ... last Sunday.* **S2:** *Next week, ... Sunday ..., will take up his position.* **Gold score = 4**		
Prediction:	3.89 ± 1.58	4.14 ± 0.0056

Table 9: Predictions for two STS-B examples by GP regression and BBB.

predictions, i.e. have small variance over all instances, thus resulting in the smaller SHP and larger NLPD. Meanwhile, models with extremely small NLPD are less confident in inaccurate predictions, and might also be under-confident in correct predictions.

We score sentence pairs in the STS-B test set using BERT Bayesian LR (BBB) and SBERT GP.[5] Overall, the incorrect predictions (> 1 from the true score) by BBB have a much smaller variance compared to those predicted by GP. For correct predictions (≤ 1 of the true score), BBB has a higher variance than for incorrect predictions, which is counter-intuitive. Though the std for SBERT GP regression on correct predictions is much larger than BBB, it's slightly less than that for incorrect ones. This fits the expectation that when a model is good at uncertainty prediction, the model should be more confident for correct predictions than incorrect ones. Examples where both models are correct and incorrect are presented in Table 9.

The near-zero variance of BBB (0.005 on average) results in infinite NLPD because of the element $\frac{(t_i - m_i)^2}{v_i}$ in the NLPD formula. Larger SHP of GP tends to produce smaller NLPD in spite of being under-confident on correct cases—the variance of 1.57 is much larger than the true gap of 0.01. So NLPD is not a perfect metric, favouring under-confident models. We therefore suggest a metric priority order of r, CAL, NLPD and SHP.

Impact of Sentence Embedding: The quality of sentence embeddings is critical for uncertainty training, affecting not only the correlation, but also the uncertainty metrics. Instead of SBERT, we also experimented with simCSE, the current state-of-the-art sentence encoder (Gao et al., 2021). We train three pipeline models with STS-B training data based on *sup-simcse-bert-base-uncased*, using the same settings as the first row of Table 3, and evaluate on the STS-B, ENMSASS, and MedSTS test sets. In Table 8, contrasting with the results in Table 2 for STS-B and Yelp, and results in Table 4 for EBMSASS and MedSTS, the correlation improves for all datasets other than MedSTS, and CAL and NLPD drop. This suggests that better sentence encoders boost pipeline performance.

High-disagreement Label Detection: A natural question to ask in the instance selection is what types of instances are selected and discarded, and how this correlates with the underlying label uncertainty in the data. When models are well-calibrated, the predicted variance will reflect the true label uncertainty, both aleatoric and epistemic. As such, if we select instances with smaller variance, we are effectively filtering out instances with higher inherent label uncertainty, as should be reflected in the labels assigned by independent annotators. We verify this hypothesis below.

We apply the model fine-tuned on STS-B over BIOSSES and EBMSASS (1000 instances each), for which five raw annotations for each instance can be accessed to approximate an empirical label distribution. KL-Divergence (KL) is used to measure the distance between the predicted and empirical probability. In Table 10, the trend in KL values on the two datasets is consistent with CAL/NLPD across all estimation methods,

[5] These two were chosen because they have similar r, but one has the largest NLPD and the other has the smallest.

	EBMSASS			BIOSSES		
	$r\uparrow$	CAL / NLPD $\downarrow$	KL1 / KL2 $\downarrow$	$r\uparrow$	CAL / NLPD $\downarrow$	KL1 / KL2 $\downarrow$
LR MC	0.828	0.236 / **3.319**	**8.75 / 1.23**	**0.870**	**0.250 / 4.488**	**8.82 / 1.54**
ConvLR MC	0.806	**0.201** / 4.668	12.74 / 1.46	0.823	0.304 / 12.59	19.64 / 2.06
LR BBB	**0.854**	0.633 / 5351.	16297 / 5.00	0.836	0.530 / 11972	16598 / 4.90
ConvLR BBB	0.806	0.736 / 1091.	2373.7 / 4.13	0.804	0.923 / 2076.	2631.2 / 5.01

Table 10: Intrinsic metrics results on EBMSASS 1000 and BIOSSES based on a model trained on STS-B. KL1 = KL-Divergence$(p\|q)$, KL2 = KL-Divergence$(q\|p)$: p = gold empirical distribution; q = predicted distribution.

indirectly suggesting that CAL and NLPD are effective metrics in the absence of empirical label distributions.

Do large-variance instances selected by strategies in Section 4.2 overlap with high-disagreement instances? Without a ground truth of high-disagreement annotations, they are identified by two steps iteratively: (1) select labels whose std is greater than α, beginning from 0.3; and (2) manually check whether for all selected instances, at least two out of the five annotations differ from the others by ≥ 1.0; if not $\alpha+=0.1$, otherwise end. This results in 137 and 31 label disagreements when $\alpha = 0.5$ and 0.4, for EBMSASS and BIOSSES, respectively.

Using BERT LR MC-dropout, a learned threshold of $\tau = 0.162$ results in Acc = 0.48, F1 = 0.28 at high-disagreement label detection on EBMSASS. For BIOSSES, $\tau = 0.1$ leads to Acc = 0.37, F1 = 0.48. Under ConvLR MC, EBMSASS has Acc = 0.46, F1 = 0.31 as $\tau = 0.124$; BIOSSES: $\tau = 0.157$ with Acc = 0.45, F1 = 0.48.

As such, high-disagreement labels can be detected by the large-variance criterion, obtaining Acc = 0.44, F1 = 0.39 on average. This is not good as a binary classifier, since regarding all instances as the majority-class ‘‘clean’’ performs better. But in our context, it is effective as a data augmentation strategy—selecting clean examples from an out-of-domain corpus. Detecting noisy labels is not just a binary classification task requiring high accuracy, but critical to recognize and filter noisy instances from a whole training corpus, even at the cost of removing clean labels.

9 Conclusion

We comprehensively investigated a range of uncertainty estimation methods over different regression tasks, using pre-trained language models. Bayesian linear regression and sparse Gaussian process regression based on fixed features obtain lower calibration error and NLPD compared with fine-tuning large-capacity deep networks end-to-end, but are inferior in terms of correlation. When embeddings are sufficiently expressive, they are comparable in performance to deep uncertainty models.

To reduce uncertainty resulting from noisy labels and limited labeled data in specific domains, we proposed a simple instance selection method based on uncertainty model predictive confidence. This approach demonstrated consistent performance improvements on three regression tasks in both self-training and active-learning settings, underscoring its effectiveness and generalizability.

Acknowledgments

We thank the anonymous reviewers and three editors for their helpful comments. Yuxia Wang is supported by scholarships from the University of Melbourne and China Scholarship Council (CSC).

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, and Vladimir Makarenkov, and Saeid Nahav. 2020. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *arXiv preprint arXiv:2011.06225*. https://doi.org/10.1016/j.inffus.2021.05.008

Daniel Beck, Trevor Cohn, and Lucia Specia. 2014. Joint emotion analysis via multi-task Gaussian processes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1798–1803. https://doi.org/10.3115/v1/D14-1190

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, volume 382, pages 41–48. ACM. https://doi.org/10.1145/1553374.1553380

Zsolt Bitvai and Trevor Cohn. 2015. Predicting peer-to-peer loan rates using Bayesian non-linear regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, pages 1613–1622. http://proceedings.mlr.press/v37/blundell15.pdf.

Enrico Camporeale and Algo Carè. 2020. Estimation of accurate and calibrated uncertainties in deterministic models. *CoRR*, abs/2003.05103.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14. Vancouver, Canada. https://www.aclweb.org/anthology/S17-2001.

Chacha Chen, Junjie Liang, Fenglong Ma, Lucas M. Glass, Jimeng Sun, and Cao Xiao. 2020. Unite: Uncertainty-based health risk prediction leveraging multi-sourced data. *arXiv preprint arXiv:2010.11389*. https://doi.org/10.1145/3442381.3450087

Hyung Won Chung, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2020. Rethinking embedding coupling in pre-trained language models. *arXiv preprint arXiv: 2010.12821*.

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18. https://doi.org/10.3115/1631862.1631865

Sharon E. Davis, Thomas A. Lasko, Guanhua Chen, Edward D. Siew, and Michael E. Matheny. 2017. Calibration drift in regression and machine learning models for acute kidney injury. *Journal of the American Medical Informatics Association*, 24(6):1052–1061. https://doi.org/10.1093/jamia/ocx030, PubMed: 28379439

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*. https://doi.org/10.18653/v1/2020.emnlp-main.21

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. https://www.aclweb.org/anthology/N19-1423.

Piero Esposito. 2020. BLiTZ – Bayesian Layers in Torch Zoo (a Bayesian deep learing library for Torch). https://github.com/piEsposito/blitz-bayesian-deep-learning/.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. http://proceedings.mlr.press/v48/gal16.pdf.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-aware machine translation evaluation. *CoRR*, abs/2109.06352. https://doi.org/10.18653/v1/2021.findings-emnlp.330

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. 2007. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268. https://doi.org/10.1111/j.1467-9868.2007.00587.x

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone? *Natural Language Engineering*, 23(1):3–30. https://doi.org/10.1017/S1351324915000339

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

Hamed Hassanzadeh, Anthony Nguyen, and Karin Verspoor. 2019. Quantifying semantic similarity of clinical evidence in the biomedical literature to facilitate related evidence synthesis. *Journal of Biomedical Informatics*. https://doi.org/10.1016/j.jbi.2019.103321, PubMed: 31676460

José Miguel Hernández-Lobato and Ryan Adams. 2015. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869. http://proceedings.mlr.press/v37/hernandez-lobatoc15.pdf.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8423–438. https://doi.org/10.1162/tacl_a_00324

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1149

Brian Keith, Exequiel Fuentes, and Claudio Meneses. 2017. A hybrid approach for sentiment analysis applied to paper. In *Proceedings of ACM SIGKDD Conference*.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584. https://proceedings.neurips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. 2018. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804. http://proceedings.mlr.press/v80/kuleshov18a/kuleshov18a.pdf.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 1. https://papers.nips.cc/paper/2010/file/e57c6b956a6521b28495f2886ca0977a-Paper.pdf.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*. https://arxiv.org/pdf/1612.01474.pdf

Max-Heinrich Laves, Sontje Ihler, Jacob F. Fast, Lüder A. Kahrs, and Tobias Ortmaier. 2020. Well-calibrated regression uncertainty in medical imaging with deep learning. In *Medical Imaging with Deep Learning*, pages 393–412. http://proceedings.mlr.press/v121/laves20a/laves20a.pdf.

Felix Leibfried, Vincent Dutordoir, S. T. John, and Nicolas Durrande. 2020. A tutorial on sparse gaussian processes and variational inference. *arXiv preprint arXiv:2012.13962*.

Specia Lucia, Fomicheva Marina, Blain Frédéric, Guzmán Paco, Chaudhary Vishrav, Fonseca Erick, and Martins André. 2020. WMT 2020 quality estimation dataset. https://www.statmt.org/wmt20/qualityestimation-task.html.

Kristian Miok, Gregor Pirs, and Marko Robnik-Sikonja. 2020. Bayesian methods for semi-supervised text annotation. *arXiv preprint arXiv:2010.14872*.

Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. 2019. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems*, volume 32, pages 6359–6370. https://proceedings.neurips.cc/paper/2019/file/84c2d4860a0fc27bcf854c444fb8b400-Paper.pdf.

Daniel Preoţiuc-Pietro and Trevor Cohn. 2013. A temporal model of text periodicities using Gaussian processes. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 977–988.

Puria Radmard, Yassir Fathullah, and Aldo Lipani. 2021. Subsequence based deep active learning for named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 4310–4321. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.332

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

C. Rusmassen and C. Williams. 2005. Gaussian process for machine learning. https://doi.org/10.7551/mitpress/3206.001.0001

Omkar Sabnis. 2018. Yelp review dataset. https://www.kaggle.com/omkarsabnis/yelp-reviews-dataset

Burr Settles. 2009. Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences. http://burrsettles.com/pub/settles.activelearning.pdf.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25–27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1070–1079. ACL. https://doi.org/10.3115/1613715.1613855

Aili Shen, Daniel Beck, Bahar Salehi, Jianzhong Qi, and Timothy Baldwin. 2019. Modelling uncertainty in collaborative document quality assessment. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 191–201, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-5525

Joachim Sicking, Maram Akila, Maximilian Pintz, Tim Wirtz, Asja Fischer, and Stefan Wrobel. 2021. A novel regression loss for non-parametric uncertainty optimization. *arXiv preprint arXiv:2101.02726*.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: A semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58. https://doi.org/10.1093/bioinformatics/btx238, PubMed: 28881973

Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. 2019. Distribution calibration for regression. In *International Conference on Machine Learning*, pages 5897–5906. http://proceedings.mlr.press/v97/song19a/song19a.pdf

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240*. https://doi.org/10.18653/v1/2021.naacl-main.28

Michalis Titsias. 2009. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. http://proceedings.mlr.press/v5/titsias09a/titsias09a.pdf.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. 2019. Evaluating model calibration in classification. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3459–3467. http://proceedings.mlr.press/v89/vaicenavicius19a/vaicenavicius19a.pdf.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020*, pages 1074–1080. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.80

Yanshan Wang, Naveed Afzal, Sunyang Fu, Liwei Wang, Feichen Shen, Majid Rastegar-Mojarad, and Hongfang Liu. 2018. MedSTS: A resource for clinical semantic textual similarity. *Language Resources and Evaluation*, pages 1–16. https://doi.org/10.1007/s10579-018-9431-1

Yanshan Wang, Sunyang Fu, Feichen Shen, Sam Henry, Ozlem Uzuner, and Hongfang Liu. 2020a. The 2019 n2c2/OHNLP track on clinical semantic textual similarity: Overview. *JMIR Medical Informatics*, 8(11). https://doi.org/10.2196/23375

Yuxia Wang, Fei Liu, Karin Verspoor, and Timothy Baldwin. 2020b. Evaluating the utility of model configurations and data augmentation on clinical semantic textual similarity. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 105–111, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.bionlp-1.11

Yuxia Wang, Karin Verspoor, and Timothy Baldwin. 2020c. Learning from unlabeled data for clinical semantic textual similarity. In *Proceedings of the 3rd Clinical NLP Workshop*, Online. EMNLP. https://doi.org/10.18653/v1/2020.clinicalnlp-1.25

Boyang Xue, Jianwei Yu, Junhao Xu, Shansong Liu, Shoukang Hu, Zi Ye, Mengzhe Geng, Xunying Liu, and Helen Meng. 2021. Bayesian transformer language models for speech recognition. *arXiv preprint arXiv:2102.04754*. https://doi.org/10.1109/ICASSP39728.2021.9414046

Eric Zelikman, Christopher Healy, Sharon Zhou, and Anand Avati. 2020. Crude: Calibrating regression uncertainty distributions empirically. *arXiv preprint arXiv:2005.12496*.