Bayesian Model-Agnostic Meta-Learning with Matrix-Valued Kernels for Quality Estimation

Abiola Obamuyide[1] **Marina Fomicheva**[1] **Lucia Specia**[1,2]
[1]Department of Computer Science, University of Sheffield
[2]Department of Computing, Imperial College London
United Kingdom
{a.obamuyide,m.fomicheva,l.specia}@sheffield.ac.uk

Abstract

Most current quality estimation (QE) models for machine translation are trained and evaluated in a fully supervised setting requiring significant quantities of labelled training data. However, obtaining labelled data can be both expensive and time-consuming. In addition, the test data that a deployed QE model would be exposed to may differ from its training data in significant ways. In particular, training samples are often labelled by one or a small set of annotators, whose perceptions of translation quality and needs may differ substantially from those of end-users, who will employ predictions in practice. Thus, it is desirable to be able to adapt QE models efficiently to new user data with limited supervision data. To address these challenges, we propose a Bayesian meta-learning approach for adapting QE models to the needs and preferences of each user with limited supervision. To enhance performance, we further propose an extension to a state-of-the-art Bayesian meta-learning approach which utilizes a matrix-valued kernel for Bayesian meta-learning of quality estimation. Experiments on data with varying number of users and language characteristics demonstrates that the proposed Bayesian meta-learning approach delivers improved predictive performance in both limited and full supervision settings.

1 Introduction

Quality Estimation (QE) models aim to evaluate the output of Machine Translation (MT) systems at run-time, when no reference translations are available (Blatz et al., 2004; Specia et al., 2009). QE models can be applied for instance to improve translation productivity by selecting high-quality translations amongst several candidates. A number of approaches have been proposed for this task (Specia et al., 2009, 2015; Kim et al., 2017; Kepler et al., 2019; Ranasinghe et al., 2020), and a shared task yearly benchmarks proposed approaches (Fonseca et al., 2019; Specia et al., 2020).

Different users of MT output have varying quality needs and standards, depending for instance on the downstream task at hand, or the level of their knowledge of the languages involved. Thus, the perception of the quality of MT output can be subjective, and therefore the quality estimates obtained from a model trained on data from one set of users may not serve the needs of a different set of users. In order to be able to make the most of these models, it is thus desirable to be able to efficiently adapt them to the needs and preferences of the end-user and with as little supervision as possible. However, most existing QE models are trained and evaluated in a fully supervised setting which assumes access to substantial quantities of labelled supervision data, which may not be available and can be expensive and time-consuming to obtain.

In order to endow QE models with the ability to learn to adapt efficiently with limited supervision data, this work proposes a Bayesian meta-learning framework for the training and evaluation of QE models that are able to adapt to the needs of end-users with limited supervision data. We further improve the performance of Bayesian meta-learning for the task of quality estimation by extending the state-of-the-art Bayesian Model-Agnostic Meta-Learning (BMAML) approach of Kim et al. (2018) to utilize Stein Variational Gradient Descent (Liu and Wang, 2016) with matrix-valued kernels (Wang et al., 2019), and demonstrate that this leads to enhanced predictive performance in both limited and full supervision settings.

2 Background

2.1 Model-Agnostic Meta-Learning

The goal of meta-learning, also known as learning to learn (Schmidhuber, 1987; Thrun and Pratt,

Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pages 223–230
Bangkok, Thailand (Online), August 6, 2021.

1998), is to develop models that can learn more efficiently over time, by generalizing from knowledge of how to solve related tasks from a given distribution of tasks. Given a learner model $f_{\boldsymbol{w}}$, for instance a neural network parametrized by $\boldsymbol{w} \in \mathbb{R}^d$, and a distribution $p(\mathcal{T})$ over tasks $\mathcal{T}$, gradient-based model-agnostic meta-learning approaches such as *MAML* (Finn et al., 2017) seek to learn the parameters of the learner model which can be quickly adapted to new tasks sampled from the same distribution of tasks with limited supervision data.

In formal terms, these approaches seek parameters $\boldsymbol{w}$ that satisfy the meta-objective:

$$\min_{\boldsymbol{w}} \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})} \left[\mathcal{L}_{\mathcal{T}} \left(\mathcal{U}_k \left(\boldsymbol{w}; \mathcal{D}_{\mathcal{T}}\right)\right)\right], \tag{1}$$

where $\mathcal{L}_{\mathcal{T}}$ is the loss and $\mathcal{D}_{\mathcal{T}}$ is training data from task $\mathcal{T}$, and $\mathcal{U}_k$ denotes k steps of a gradient descent learning rule such as SGD.

Intuitively, the meta-objective explicitly encourages the model to learn model parameters that can be quickly adapted to achieve optimum predictive performance across all tasks using limited supervision data and with as few gradient descent steps as possible.

In order to account for uncertainty and improve robustness, Bayesian approaches to meta-learning have also been proposed (Kim et al., 2018; Finn et al., 2018; Ravi and Beatson, 2019; Wang et al., 2020; Nguyen et al., 2020). In contrast to their non-Bayesian counterparts which learn point estimates of the parameters, Bayesian meta-learning approaches learn a distribution over the parameters to further improve robustness in limited supervision settings.

2.2 Stein Variational Gradient Descent

Stein Variational Gradient Descent (SVGD)(Liu and Wang, 2016) is a Bayesian inference method which works by initializing a set of samples, also known as particles, from a simple distribution and iteratively updating the particles to match samples from a target distribution. Because its particle update rule is deterministic and differentiable, it can be used to perform Bayesian inference in the meta-learning inner loop, since the entire update process can still be differentiated through for gradient-based updates from the outer loop, for instance as was done in Kim et al. (2018).

In order to obtain N samples from a posterior $p(\boldsymbol{w})$, SVGD maintains N samples of model parameters, and iteratively transports the samples to match samples from the target distribution. Let the samples be represented by $W = \{\boldsymbol{w}^n\}_{n=1}^N$. At each successive iteration t, SVGD updates each sample with the following update rule:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha_t \phi\left(\boldsymbol{w}_t\right), \tag{2}$$

where $\phi\left(\boldsymbol{w}_t\right) =$

$$\frac{1}{N}\sum_{n=1}^{N}\left[k\left(\boldsymbol{w}_t^n, \boldsymbol{w}_t\right)\nabla_{\boldsymbol{w}_t^n}\log p\left(\boldsymbol{w}_t^n\right) + \nabla_{\boldsymbol{w}_t^n} k\left(\boldsymbol{w}_t^n, \boldsymbol{w}_t\right)\right], \tag{3}$$

α_t is a step-size parameter and $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a scalar-valued positive-definite kernel such as the Radial Basis Function (RBF) kernel. Intuitively, the first term in Equation 3 implies that a particle determines its update direction through a weighted aggregate of the gradients from the other particles, with the kernel distance between the particles serving as the weight. Thus, closer particles have more weight in the aggregate. The second term of the equation can be understood as a repulsive force that prevents the particles from collapsing to a single point. For the case when the number of particles is one, the SVGD update procedure reduces to standard gradient ascent on the objective $p(\boldsymbol{w})$ for any kernel with the property $\nabla_{\boldsymbol{w}} k\left(\boldsymbol{w}, \boldsymbol{w}\right) = 0$, such as the RBF kernel. SVGD has been applied in a wide range of settings, including reinforcement learning (Liu et al., 2017; Haarnoja et al., 2017), uncertainty quantification (Zhu and Zabaras, 2018), and online continual learning (Obamuyide et al., 2021).

2.3 Stein Variational Gradient Descent with Matrix-Valued Kernels

Let $\mathcal{H}_k$ denote a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with kernel k. Wang et al. (2019) observed that the original SVGD as proposed in Liu and Wang (2016) searches for the optimal update direction ϕ in RKHS $\mathcal{H}_k^d = \mathcal{H}_k \times \cdots \times \mathcal{H}_k$, a product of d copies of RKHS of scalar-valued functions, which does not allow the encoding of any potential correlations between different co-ordinates of ϕ. Wang et al. (2019) proposed *Matrix-SVGD*, which addressed this limitation by replacing $\mathcal{H}_k^d$ with a more general RKHS of vector-valued functions (also known as vector-valued RKHS), which uses *matrix-valued* positive-definite kernels to specify rich correlation structures between the different co-ordinates. Concretely, Equation 3 as used in SVGD is replaced with Equation 4:

$$\phi(\boldsymbol{w}_t) = \frac{1}{N}\sum_{n=1}^{N}\left[\boldsymbol{K}(\boldsymbol{w}_t, \boldsymbol{w}_t^n)\nabla_{\boldsymbol{w}_t^n}\log p(\boldsymbol{w}_t^n) + \boldsymbol{K}(\boldsymbol{w}_t, \boldsymbol{w}_t^n)\nabla_{\boldsymbol{w}_t^n}\right], \tag{4}$$

where $\boldsymbol{K} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^{d\times d}$ is now a matrix-valued kernel, and $\boldsymbol{K}(\cdot, \boldsymbol{w})\nabla_{\boldsymbol{w}}$ is formally defined as the product of matrix $\boldsymbol{K}(\cdot, \boldsymbol{w})$ with vector $\nabla_{\boldsymbol{w}}$. The ℓ-th element of $\boldsymbol{K}(\cdot, \boldsymbol{w})\nabla_{\boldsymbol{w}}$ is computed as:

$$(\boldsymbol{K}(\cdot, \boldsymbol{w})\nabla_{\boldsymbol{w}})_\ell = \sum_{m=1}^{d}\nabla_{w^m}K_{\ell,m}(\cdot, \boldsymbol{w}), \tag{5}$$

where $K_{\ell,m}(\boldsymbol{w}, \boldsymbol{w}')$ represents the (ℓ, m)-element of matrix $\boldsymbol{K}(\boldsymbol{w}, \boldsymbol{w}')$ and w^m the m-element of $\boldsymbol{w}$.

Importantly, the advantage of *Matrix-SVGD* over the original SVGD algorithm is that it allows us to pre-condition SVGD by constructing a proper matrix kernel which incorporates the pre-conditioning information, in order to accelerate exploration and convergence.

2.4 Bayesian Model-Agnostic Meta-Learning

Kim et al. (2018) proposed a Bayesian Model-Agnostic Meta-Learning (BMAML) algorithm which learns a distribution over parameters which, when given data from a new task, can be adapted quickly to a task-specific distribution using SVGD updates as defined in Equation 3. Thus, BMAML as proposed in Kim et al. (2018) makes use of scalar-valued kernels for SVGD updates, which (as discussed earlier) does not allow the encoding of potential correlations between different parameter co-ordinates for effective optimization, a limitation which we next address.

3 Bayesian Model-Agnostic Meta-Learning with Matrix-SVGD

In this work we propose to improve the predictive performance of BMAML for quality estimation with the use of the Matrix-SVGD, which uses matrix-valued kernels for more effective parameter updates, in place of the original SVGD algorithm used in Kim et al. (2018). As pre-conditioning information, we use $\boldsymbol{P}$, the average of the Fisher information matrix of the particles:

$$\boldsymbol{P} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{F}(\boldsymbol{w}_n), \tag{6}$$

where $\boldsymbol{F}(\boldsymbol{w}_n)$ is the Fisher information matrix for particle $\boldsymbol{w}_n$. The matrix-valued kernel is then computed as:

$$\boldsymbol{K}_{\boldsymbol{P}}(\boldsymbol{w}, \boldsymbol{w}') = \boldsymbol{P}^{-1}\exp\left(-\frac{1}{2h}\left\|\boldsymbol{w} - \boldsymbol{w}'\right\|_{\boldsymbol{P}}^2\right), \tag{7}$$

where $\|\boldsymbol{w} - \boldsymbol{w}'\|_{\boldsymbol{P}}^2 := (\boldsymbol{w} - \boldsymbol{w}')^\top \boldsymbol{P}(\boldsymbol{w} - \boldsymbol{w}')$ and h is a bandwidth parameter.

The full algorithm, which we refer to as *Matrix-BMAML*, is outlined in Algorithm 1. We use machine translation quality estimation as a case study in this work, and so assume access to a distribution of quality estimation tasks $p(T)$ (each QE task can be a QE user/annotator/post-editor with their corresponding data), and a quality estimation model f_W parameterized by W, though the approach can also be applied to other natural language processing or computer vision tasks.

Algorithm 1 Bayesian Model-Agnostic Meta-Learning with Matrix-SVGD

Require: Distribution of QE tasks $p(\mathcal{T})$
Require: QE model f_W, Number of update steps K
Require: Learning rates α, β

1: Initialize W
2: **while** not done **do**
3: Sample batch of QE tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for each** $\mathcal{T}_i$ **do**
5: Sample $\mathcal{D}_{\mathcal{T}_i}^{\text{train}}$ from $\mathcal{T}_i^{train}$
6: Sample $\mathcal{D}_{\mathcal{T}_i}^{\text{val}}$ from $\mathcal{T}_i^{val}$
7: $W_0^i \leftarrow W$
8: **for** k = 1,..K **do**
9: W_k^i = Matrix-SVGD($W_{k-1}^i; \mathcal{D}_{\mathcal{T}_i}^{\text{train}}, \alpha$)
10: **end for**
11: **end for**
12: $W \leftarrow W - \beta\nabla_W \sum_{\mathcal{T}_i\sim p(\mathcal{T})}\mathcal{L}\left(f_{W_K^i}; \mathcal{D}_{\mathcal{T}_i}^{\text{val}}\right)$
13: **end while**

We first initialize the parameters of the quality estimation model (line 1). Then in each iteration, we sample a batch of QE tasks (line 3), and for each QE task, we sample instances from its training and validation sets (lines 4-6). Thereafter, task-specific parameters are initialized from the model's parameters (line 7), and then updated with K steps of Matrix-SVGD (using Equations (2) and (4) to (7)) (lines 8-10). At the end of each iteration, a meta-update is performed on the model's parameters W.

4 Experiments and Results

We conduct experiments in two settings: in a limited supervision setting, where we provide all models access to only a limited number of training instances per QE task; and in a full-supervision setting, where we provide the models with access to all available training instances for each QE task.

PE ID	Train	Dev	Test
PE1	1440	360	200
PE2	2160	540	300
PE3	1444	361	195
PE4	1834	459	244
PE5	4866	1217	617
PE6	1677	420	203
PE7	1567	392	241
Total	14988	3749	2000

(a) QT21 en-lv (nmt)

PE ID	Train	Dev	Test
PE1	9952	2488	559
PE2	3445	862	193
PE3	8770	2193	537
PE4	4579	1145	276
PE5	7651	1913	435
Total	34397	8601	2000

(b) QT21 en-cs (smt)

Table 1: Number of instances per QE Task/Post Editor (PE) for the QT21 dataset.

The QT21 Dataset We evaluate our approach with the publicly available **QT21** (Specia et al., 2017), a large-scale dataset containing translations from both statistical (smt) and neural (nmt) machine translation systems in multiple language directions [1]. This is the largest dataset with annotator information available. We make use of data from the English-Latvian (en-lv) and English-Czech (en-cs) language directions. The languages were chosen as they contain the largest number of annotators. Each instance in the dataset is a tuple of source sentence, its machine translation, the corresponding post-edited translation by a professional translator (post-editor), a reference translation and other information such as (anonymized) post-editor identifier. We construct a QE dataset from this corpus by computing the HTER (Snover et al., 2006) values between each source sentence and its post-edited translation. We thereafter split the data into train, dev and test splits for each post-editor, which constitutes a QE task. A breakdown of the number of train, dev and test instances per QE task/post-editor is available in Table 1.

5 QE Model

The quality estimation model used by all methods is based on multi-lingual DistilBERT (Sanh et al., 2019), a smaller version of multi-lingual BERT (Devlin et al., 2019) trained with knowledge distillation (Buciluǎ et al., 2006; Hinton et al., 2015). It accepts as input the source and machine translation outputs concatenated as a single text, separated by a '[SEP]' token and prepended with a '[CLS]' token. The representation of the '[CLS]' token is then passed to a linear layer to predict HTER (Snover et al., 2006) values as regression targets.

Benchmark Approaches We compare the proposed approach with the following: MTL-PRETRAIN is a baseline trained in classic multi-task fashion for multiple epochs using data from all QE tasks. It is thereafter fine-tuned using each QE task's training data before making predictions on its test set, in a similar fashion as the meta-learning approaches; REPTILE (Nichol and Schulman, 2018); Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017); implicit Model-Agnostic Meta-Learning (iMAML) (Rajeswaran et al., 2019); Amortized Bayesian Meta-Learning (ABML) (Ravi and Beatson, 2019); and BMAML (Kim et al., 2018), a state-of-the-art Bayesian meta-learning method.

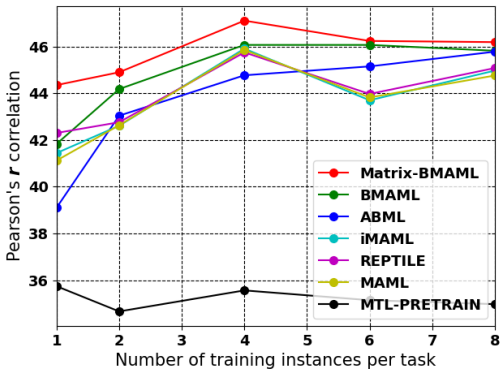
Evaluation We report Pearson's r correlation scores and Mean Absolute Error (MAE) between model output and gold labels, both standard evaluation metrics in QE.

Each experiment is repeated across five (5) different random seeds, and we report the average.

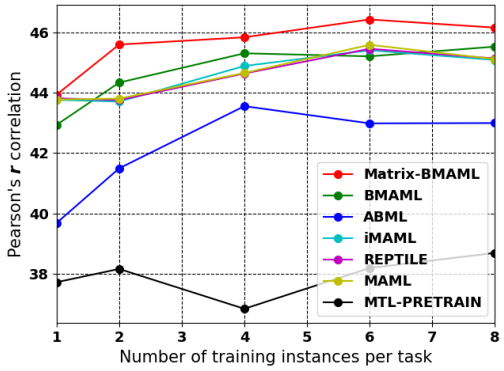
5.1 Limited Supervision Results

Results obtained in a setting where all approaches have access to only very limited training instances is presented in Figure 1. As expected, training with classic multi-task learning and then fine-tuning on the training data of each QE task (*MTL-PRETRAIN*) results in very poor performance on both datasets. This result is consistent with the results observed in Finn et al. (2017), since classic multi-task learning does not have any explicit objective that encourages the model to learn how to learn with limited supervision data. In contrast, all meta-learning approaches obtain consistent improvements over the *MTL-PRETRAIN* baseline. We find that in general, our approach (*Matrix-BMAML*) obtains marked performance improvements over the other Bayesian and non-Bayesian meta-learning approaches. This demonstrates the importance of incorporating pre-conditioning information through matrix-valued kernels for more ef-

[1] http://www.qt21.eu/resources/data/



(a)



(b)

Figure 1: Results obtained using limited training instances for each task on the (a) *en-lv* and (b) *en-cs* quality estimation datasets.

fective SVGD updates in Bayesian model-agnostic meta-learning.

5.2 Full Supervision Results

Method	en-lv		en-cs	
	Pearson ↑	MAE ↓	Pearson ↑	MAE ↓
MTL-PRETRAIN	0.4505	0.1936	0.4473	0.1711
MAML	0.5239	0.1590	0.4894	0.1611
REPTILE	0.5237	0.1591	0.5037	0.1605
iMAML	0.5254	0.1588	0.5036	0.1605
ABML	0.5196	0.1600	0.4807	0.1620
BMAML	0.5295	**0.1585**	0.4963	0.1606
Matrix-BMAML	**0.5377**	0.1588	**0.5202**	**0.1566**

Table 2: Comparison with existing approaches.

Table 2 presents results obtained when the approaches are given access to all available training data for each QE task. We can observe that *Matrix-BMAML* obtained the best MAE on the *en-cs* dataset, and the best Pearson's correlation on both datasets, which again demonstrates the effectiveness of our approach in this setting.

6 Conclusions

We proposed a Bayesian meta-learning framework for adapting machine translation quality estimation models to the quality needs and preferences of each user with limited supervision data. We further extend a state-of-the-art Bayesian meta-learning method with the use of matrix-valued kernels, which enables the incorporation of preconditioning information for more effective SVGD updates. Using data from two language directions, we demonstrate improved predictive performance in both limited and full-supervision settings over recent state-of-the-art Bayesian and non-Bayesian meta-learning methods.

Acknowledgements

This work was supported by funding from the Bergamot project (EU H2020 grant no. 825303).

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Chelsea Finn, Kelvin Xu, and S. Levine. 2018. Probabilistic model-agnostic meta-learning. In *Advances In Neural Information Processing Systems*.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 562–568. Association for Computational Linguistics.

Taesup Kim, Jaesik Yoon, O. Dia, S. Kim, Yoshua Bengio, and Sungjin Ahn. 2018. Bayesian model-agnostic meta-learning. In *Advances In Neural Information Processing Systems*.

Qiang Liu and Dilin Wang. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems 29*, pages 2378–2386. Curran Associates, Inc.

Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. 2017. Stein variational policy gradient. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press.

Cuong Nguyen, Thanh-Toan Do, and Gustavo Carneiro. 2020. Uncertainty in model-agnostic meta-learning using variational inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3090–3100.

Alex Nichol and John Schulman. 2018. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(2):1.

Abiola Obamuyide, Marina Fomicheva, and Lucia Specia. 2021. Continual quality estimation with online bayesian meta-learning. In *Proceedings of the Association for Computational Linguistics*.

Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 113–124.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest at wmt2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Sachin Ravi and Alex Beatson. 2019. Amortized bayesian meta-learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Jurgen Schmidhuber. 1987. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, MA.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Kim Harris, Aljoscha Burchardt, Marco Turchi, Matteo Negri, and Inguna Skadina. 2017. Translation quality and productivity: A study on rich morphology languages. In *Machine Translation Summit XVI*, pages 55–71.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, System Demonstrations*, pages 115–120. The Association for Computer Linguistics.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating

the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Sebastian Thrun and Lorien Pratt. 1998. Learning to Learn: Introduction and Overview. In *Learning to Learn*, pages 3–17. Springer US, Boston, MA.

Dilin Wang, Ziyang Tang, C. Bajaj, and Qiang Liu. 2019. Stein variational gradient descent with matrix-valued kernels. *Advances in neural information processing systems*, 32:7834–7844.

Zhenyi Wang, Yang Zhao, Ping Yu, Ruiyi Zhang, and Changyou Chen. 2020. Bayesian meta sampling for fast uncertainty adaptation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Yinhao Zhu and Nicholas Zabaras. 2018. Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys.*, 366:415–447.

A Additional Experimental Details

Hyper-parameter	Value
Learning rate	3e-5
Mini-batch size	16
Max. sequence length	100

Table 3: Hyper-parameter values for all compared approaches

All compared approaches have a run time of about two hours on average. Each model was implemented as a linear layer on top of multilingual DistilBERT (Sanh et al., 2019), which has a total of 134M parameters. [2]

For the evaluation metrics, Pearson r correlation and MAE, we use open-source implementations available in SciPy [3] and scikit-learn [4] libraries respectively.

All models make use of the same values for hyper-parameters such as learning rate and batch size, selected by manual search in initial experiments. These are provided in Table 3.

[2]https://huggingface.co/distilbert-base-multilingual-cased

[3]https://www.scipy.org

[4]https://scikit-learn.org