

BLEURT: Learning Robust Metrics for Text Generation

Thibault Sellam Dipanjan Das Ankur P. Parikh

Google Research

New York, NY

{tsellam, dipanjand, aparikh}@google.com

Abstract

Text generation has made significant advances in the last few years. Yet, evaluation metrics have lagged behind, as the most popular choices (e.g., BLEU and ROUGE) may correlate poorly with human judgments. We propose BLEURT, a learned evaluation metric based on BERT that can model human judgments with a few thousand possibly biased training examples. A key aspect of our approach is a novel pre-training scheme that uses millions of synthetic examples to help the model generalize. BLEURT provides state-of-the-art results on the last three years of the WMT Metrics shared task and the WebNLG Competition dataset. In contrast to a vanilla BERT-based approach, it yields superior results even when the training data is scarce and out-of-distribution.

1 Introduction

In the last few years, research in natural text generation (NLG) has made significant progress, driven largely by the neural encoder-decoder paradigm (Sutskever et al., 2014; Bahdanau et al., 2015) which can tackle a wide array of tasks including translation (Koehn, 2009), summarization (Mani, 1999; Chopra et al., 2016), structured-data-to-text generation (McKeown, 1992; Kukich, 1983; Wiseman et al., 2017) dialog (Smith and Hipp, 1994; Vinyals and Le, 2015) and image captioning (Fang et al., 2015). However, progress is increasingly impeded by the shortcomings of existing metrics (Wiseman et al., 2017; Ma et al., 2019; Tian et al., 2019).

Human evaluation is often the best indicator of the quality of a system. However, designing crowd sourcing experiments is an expensive and high-latency process, which does not easily fit in a daily model development pipeline. Therefore, NLG researchers commonly use *automatic*

evaluation metrics, which provide an acceptable proxy for quality and are very cheap to compute. This paper investigates sentence-level, reference-based metrics, which describe the extent to which a *candidate* sentence is similar to a *reference* one. The exact definition of similarity may range from string overlap to logical entailment.

The first generation of metrics relied on hand-crafted rules that measure the surface similarity between the sentences. To illustrate, BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), two popular metrics, rely on N-gram overlap. Because those metrics are only sensitive to lexical variation, they cannot appropriately reward semantic or syntactic variations of a given reference. Thus, they have been repeatedly shown to correlate poorly with human judgment, in particular when all the systems to compare have a similar level of accuracy (Liu et al., 2016; Novikova et al., 2017; Chaganty et al., 2018).

Increasingly, NLG researchers have addressed those problems by injecting *learned* components in their metrics. To illustrate, consider the WMT Metrics Shared Task, an annual benchmark in which translation metrics are compared on their ability to imitate human assessments. The last two years of the competition were largely dominated by neural net-based approaches, RUSE, YiSi and ESIM (Ma et al., 2018, 2019). Current approaches largely fall into two categories. *Fully learned metrics*, such as BEER, RUSE, and ESIM are trained end-to-end, and they typically rely on handcrafted features and/or learned embeddings. Conversely, *hybrid metrics*, such as YiSi and BERTscore combine trained elements, e.g., contextual embeddings, with handwritten logic, e.g., as token alignment rules. The first category typically offers great expressivity: if a training set of human ratings data is available, the metrics may take full advantage of it and fit the ratings distribution tightly. Fur-

thermore, learned metrics can be tuned to measure task-specific properties, such as fluency, faithfulness, grammar, or style. On the other hand, hybrid metrics offer robustness. They may provide better results when there is little to no training data, and they do not rely on the assumption that training and test data are identically distributed.

And indeed, the IID assumption is particularly problematic in NLG evaluation because of *domain drifts*, that have been the main target of the metrics literature, but also because of *quality drifts*: NLG systems tend to get better over time, and therefore a model trained on ratings data from 2015 may fail to distinguish top performing systems in 2019, especially for newer research tasks. An ideal learned metric would be able to both take full advantage of available ratings data for training, and be robust to distribution drifts, i.e., it should be able to *extrapolate*.

Our insight is that it is possible to combine expressivity and robustness by pre-training a fully learned metric on large amounts of synthetic data, before fine-tuning it on human ratings. To this end, we introduce BLEURT,¹ a text generation metric based on BERT (Devlin et al., 2019). A key ingredient of BLEURT is a novel pre-training scheme, which uses random perturbations of Wikipedia sentences augmented with a diverse set of lexical and semantic-level supervision signals.

To demonstrate our approach, we train BLEURT for English and evaluate it under different generalization regimes. We first verify that it provides state-of-the-art results on all recent years of the WMT Metrics Shared task (2017 to 2019, to-English language pairs). We then stress-test its ability to cope with quality drifts with a synthetic benchmark based on WMT 2017. Finally, we show that it can easily adapt to a different domain with three tasks from a data-to-text dataset, WebNLG 2017 (Gardent et al., 2017). Ablations show that our synthetic pretraining scheme increases performance in the IID setting, and is critical to ensure robustness when the training data is scarce, skewed, or out-of-domain.

The code and pre-trained models are available online².

¹Bilingual Evaluation Understudy with Representations from Transformers. We refer the intrigued reader to Papineni et al. 2002 for a justification of the term *understudy*.

²<http://github.com/google-research/bleurt>

2 Preliminaries

Define $\mathbf{x} = (x_1, \dots, x_r)$ to be the reference sentence of length r where each x_i is a token and let $\tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_p)$ be a prediction sentence of length p . Let $\{(\mathbf{x}_i, \tilde{\mathbf{x}}_i, y_i)\}_{i=1}^N$ be a training dataset of size N where $y_i \in \mathbb{R}$ is the human rating that indicates how good $\tilde{\mathbf{x}}_i$ is with respect to \mathbf{x}_i . Given the training data, our goal is to learn a function $f : (\mathbf{x}, \tilde{\mathbf{x}}) \rightarrow y$ that predicts the human rating.

3 Fine-Tuning BERT for Quality Evaluation

Given the small amounts of rating data available, it is natural to leverage unsupervised representations for this task. In our model, we use BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), which is an unsupervised technique that learns contextualized representations of sequences of text. Given \mathbf{x} and $\tilde{\mathbf{x}}$, BERT is a Transformer (Vaswani et al., 2017) that returns a sequence of contextualized vectors:

$$\mathbf{v}_{[\text{CLS}]}, \mathbf{v}_{x_1}, \dots, \mathbf{v}_{x_r}, \mathbf{v}_1, \dots, \mathbf{v}_{\tilde{x}_p} = \text{BERT}(\mathbf{x}, \tilde{\mathbf{x}})$$

where $\mathbf{v}_{[\text{CLS}]}$ is the representation for the special [CLS] token. As described by Devlin et al. (2019), we add a linear layer on top of the [CLS] vector to predict the rating:

$$\hat{y} = f(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbf{W} \tilde{\mathbf{v}}_{[\text{CLS}]} + \mathbf{b}$$

where \mathbf{W} and \mathbf{b} are the weight matrix and bias vector respectively. Both the above linear layer as well as the BERT parameters are trained (i.e. fine-tuned) on the supervised data which typically numbers in a few thousand examples. We use the regression loss $\ell_{\text{supervised}} = \frac{1}{N} \sum_{n=1}^N \|y_i - \hat{y}\|^2$.

Although this approach is quite straightforward, we will show in Section 5 that it gives state-of-the-art results on WMT Metrics Shared Task 17-19, which makes it a high-performing evaluation metric. However, fine-tuning BERT requires a sizable amount of IID data, which is less than ideal for a metric that should generalize to a variety of tasks and model drift.

4 Pre-Training on Synthetic Data

The key aspect of our approach is a pre-training technique that we use to “warm up” BERT before fine-tuning on rating data.³ We generate a large

³To clarify, our pre-training scheme is an addition, not a replacement to BERT’s initial training (Devlin et al., 2019) and happens after it.

number of of synthetic reference-candidate pairs (z, \tilde{z}) , and we train BERT on several lexical- and semantic-level supervision signals with a multi-task loss. As our experiments will show, BLEURT generalizes much better after this phase, especially with incomplete training data.

Any pre-training approach requires a dataset and a set of pre-training tasks. Ideally, the setup should resemble the final NLG evaluation task, i.e., the sentence pairs should be distributed similarly and the pre-training signals should correlate with human ratings. Unfortunately, we cannot have access to the NLG models that we will evaluate in the future. Therefore, we optimized our scheme for generality, with three requirements. (1) The set of reference sentences should be large and diverse, so that BLEURT can cope with a wide range of NLG domains and tasks. (2) The sentence pairs should contain a wide variety of lexical, syntactic, and semantic dissimilarities. The aim here is to anticipate all variations that an NLG system may produce, e.g., phrase substitution, paraphrases, noise, or omissions. (3) The pre-training objectives should effectively capture those phenomena, so that BLEURT can learn to identify them. The following sections present our approach.

4.1 Generating Sentence Pairs

One way to expose BLEURT to a wide variety of sentence differences is to use existing sentence pairs datasets (Bowman et al., 2015; Williams et al., 2018; Wang et al., 2019). These sets are a rich source of related sentences, but they may fail to capture the errors and alterations that NLG systems produce (e.g., omissions, repetitions, non-sensical substitutions). We opted for an automatic approach instead, that can be scaled arbitrarily and at little cost: we generate synthetic sentence pairs (z, \tilde{z}) by randomly perturbing 1.8 million segments z from Wikipedia. We use three techniques: mask-filling with BERT, backtranslation, and randomly dropping out words. We obtain about 6.5 million perturbations \tilde{z} . Let us describe those techniques.

Mask-filling with BERT: BERT’s initial training task is to fill gaps (i.e., masked tokens) in tokenized sentences. We leverage this functionality by inserting masks at random positions in the Wikipedia sentences, and fill them with the language model. Thus, we introduce lexical alter-

ations while maintaining the fluency of the sentence. We use two masking strategies—we either introduce the masks at random positions in the sentences, or we create contiguous sequences of masked tokens. More details are provided in the Appendix.

Backtranslation: We generate paraphrases and perturbations with backtranslation, that is, round trips from English to another language and then back to English with a translation model (Bannard and Callison-Burch, 2005; Ganitkevitch et al., 2013; Sennrich et al., 2016). Our primary aim is to create variants of the reference sentence that preserves semantics. Additionally, we use the mispredictions of the backtranslation models as a source of realistic alterations.

Dropping words: We found it useful in our experiments to randomly drop words from the synthetic examples above to create other examples. This method prepares BLEURT for “pathological” behaviors or NLG systems, e.g., void predictions, or sentence truncation.

4.2 Pre-Training Signals

The next step is to augment each sentence pair (z, \tilde{z}) with a set of pre-training signals $\{\tau_k\}$, where τ_k is the target vector of pre-training task k . Good pre-training signals should capture a wide variety of lexical and semantic differences. They should also be cheap to obtain, so that the approach can scale to large amounts of synthetic data. The following section presents our 9 pre-training tasks, summarized in Table 1. Additional implementation details are in the Appendix.

Automatic Metrics: We create three signals τ_{BLEU} , τ_{ROUGE} , and $\tau_{\text{BERTscore}}$ with sentence BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BERTscore (Zhang et al., 2020) respectively (we use precision, recall and F-score for the latter two).

Backtranslation Likelihood: The idea behind this signal is to leverage existing translation models to measure semantic equivalence. Given a pair (z, \tilde{z}) , this training signal measures the probability that \tilde{z} is a backtranslation of z , $P(\tilde{z}|z)$, normalized by the length of \tilde{z} . Let $P_{\text{en} \rightarrow \text{fr}}(z_{\text{fr}}|z)$ be a translation model that assigns probabilities to French sentences z_{fr} conditioned on English sentences z and let $P_{\text{fr} \rightarrow \text{en}}(z|z_{\text{fr}})$ be a translation model that assigns probabilities to English

Task Type	Pre-training Signals	Loss Type
BLEU	τ_{BLEU}	Regression
ROUGE	$\tau_{\text{ROUGE}} = (\tau_{\text{ROUGE-P}}, \tau_{\text{ROUGE-R}}, \tau_{\text{ROUGE-F}})$	Regression
BERTscore	$\tau_{\text{BERTscore}} = (\tau_{\text{BERTscore-P}}, \tau_{\text{BERTscore-R}}, \tau_{\text{BERTscore-F}})$	Regression
Backtrans. likelihood	$\tau_{\text{en-fr}, z \tilde{z}}, \tau_{\text{en-fr}, \tilde{z} z}, \tau_{\text{en-de}, z \tilde{z}}, \tau_{\text{en-de}, \tilde{z} z}$	Regression
Entailment	$\tau_{\text{entail}} = (\tau_{\text{Entail}}, \tau_{\text{Contradict}}, \tau_{\text{Neutral}})$	Multiclass
Backtrans. flag	$\tau_{\text{backtran_flag}}$	Multiclass

Table 1: Our pre-training signals.

sentences given french sentences. If $|\tilde{z}|$ is the number of tokens in \tilde{z} , we define our score as $\tau_{\text{en-fr}, \tilde{z}|z} = \frac{\log P(\tilde{z}|z)}{|\tilde{z}|}$, with:

$$P(\tilde{z}|z) = \sum_{z_{\text{fr}}} P_{\text{fr} \rightarrow \text{en}}(\tilde{z}|z_{\text{fr}}) P_{\text{en} \rightarrow \text{fr}}(z_{\text{fr}}|z)$$

Because computing the summation over all possible French sentences is intractable, we approximate the sum using $z_{\text{fr}}^* = \arg \max P_{\text{en} \rightarrow \text{fr}}(z_{\text{fr}}|z)$ and we assume that $P_{\text{en} \rightarrow \text{fr}}(z_{\text{fr}}^*|z) \approx 1$:

$$P(\tilde{z}|z) \approx P_{\text{fr} \rightarrow \text{en}}(\tilde{z}|z_{\text{fr}}^*)$$

We can trivially reverse the procedure to compute $P(z|\tilde{z})$, thus we create 4 pre-training signals $\tau_{\text{en-fr}, z|\tilde{z}}, \tau_{\text{en-fr}, \tilde{z}|z}, \tau_{\text{en-de}, z|\tilde{z}}, \tau_{\text{en-de}, \tilde{z}|z}$ with two pairs of languages ($\text{en} \leftrightarrow \text{de}$ and $\text{en} \leftrightarrow \text{fr}$) in both directions.

Textual Entailment: The signal τ_{entail} expresses whether z entails or contradicts \tilde{z} using a classifier. We report the probability of three labels: *Entail*, *Contradict*, and *Neutral*, using BERT fine-tuned on an entailment dataset, MNLI (Devlin et al., 2019; Williams et al., 2018).

Backtranslation flag: The signal $\tau_{\text{backtran_flag}}$ is a Boolean that indicates whether the perturbation was generated with backtranslation or with mask-filling.

4.3 Modeling

For each pre-training task, our model uses either a regression or a classification loss. We then aggregate the task-level losses with a weighted sum.

Let τ_k describe the target vector for each task, e.g., the probabilities for the classes *Entail*, *Contradict*, *Neutral*, or the precision, recall, and F-score for ROUGE. If τ_k is a regression task, then the loss used is the ℓ_2 loss i.e. $\ell_k = \|\tau_k - \hat{\tau}_k\|_2^2 / |\tau_k|$ where $|\tau_k|$ is the dimension of τ_k and $\hat{\tau}_k$ is computed by using a task-specific linear layer on top of the [CLS] embedding: $\hat{\tau}_k =$

$\mathbf{W}_{\tau_k} \tilde{\mathbf{v}}_{[\text{CLS}]} + \mathbf{b}_{\tau_k}$. If τ_k is a classification task, we use a separate linear layer to predict a logit for each class c : $\hat{\tau}_{kc} = \mathbf{W}_{\tau_{kc}} \tilde{\mathbf{v}}_{[\text{CLS}]} + \mathbf{b}_{\tau_{kc}}$, and we use the multiclass cross-entropy loss. We define our aggregate pre-training loss function as follows:

$$\ell_{\text{pre-training}} = \frac{1}{M} \sum_{m=1}^M \sum_{k=1}^K \gamma_k \ell_k(\tau_k^m, \hat{\tau}_k^m) \quad (1)$$

where τ_k^m is the target vector for example m , M is number of synthetic examples, and γ_k are hyperparameter weights obtained with grid search (more details in the Appendix).

5 Experiments

In this section, we report our experimental results for two tasks, translation and data-to-text. First, we benchmark BLEURT against existing text generation metrics on the last 3 years of the WMT Metrics Shared Task (Bojar et al., 2017). We then evaluate its robustness to quality drifts with a series of synthetic datasets based on WMT17. We test BLEURT’s ability to adapt to different tasks with the WebNLG 2017 Challenge Dataset (Gardent et al., 2017). Finally, we measure the contribution of each pre-training task with ablation experiments.

Our Models: Unless specified otherwise, all BLEURT models are trained in three steps: regular BERT pre-training (Devlin et al., 2019), pre-training on synthetic data (as explained in Section 4), and fine-tuning on task-specific ratings (translation and/or data-to-text). We experiment with two versions of BLEURT, BLEURT and BLEURTbase, respectively based on BERT-Large (24 layers, 1024 hidden units, 16 heads) and BERT-Base (12 layers, 768 hidden units, 12 heads) (Devlin et al., 2019), both uncased. We use batch size 32, learning rate 1e-5, and 800,000 steps for pre-training and 40,000 steps for fine-tuning. We provide the full detail of our training setup in the Appendix.

model	cs-en τ / r	de-en τ / r	fi-en τ / r	lv-en τ / r	ru-en τ / r	tr-en τ / r	zh-en τ / r	avg τ / r
sentBLEU	29.6 / 43.2	28.9 / 42.2	38.6 / 56.0	23.9 / 38.2	34.3 / 47.7	34.3 / 54.0	37.4 / 51.3	32.4 / 47.5
MoverScore	47.6 / 67.0	51.2 / 70.8	NA	NA	53.4 / 73.8	56.1 / 76.2	53.1 / 74.4	52.3 / 72.4
BERTscore w/ BERT	48.0 / 66.6	50.3 / 70.1	61.4 / 81.4	51.6 / 72.3	53.7 / 73.0	55.6 / 76.0	52.2 / 73.1	53.3 / 73.2
BERTscore w/ roBERTa	54.2 / 72.6	56.9 / 76.0	64.8 / 83.2	56.2 / 75.7	57.2 / 75.2	57.9 / 76.1	58.8 / 78.9	58.0 / 76.8
chrF++	35.0 / 52.3	36.5 / 53.4	47.5 / 67.8	33.3 / 52.0	41.5 / 58.8	43.2 / 61.4	40.5 / 59.3	39.6 / 57.9
BEER	34.0 / 51.1	36.1 / 53.0	48.3 / 68.1	32.8 / 51.5	40.2 / 57.7	42.8 / 60.0	39.5 / 58.2	39.1 / 57.1
BLEURTbase -pre	51.5 / 68.2	52.0 / 70.7	66.6 / 85.1	60.8 / 80.5	57.5 / 77.7	56.9 / 76.0	52.1 / 72.1	56.8 / 75.8
BLEURTbase	55.7 / 73.4	56.3 / 75.7	68.0 / 86.8	64.7 / 83.3	60.1 / 80.1	62.4 / 81.7	59.5 / 80.5	61.0 / 80.2
BLEURT -pre	56.0 / 74.7	57.1 / 75.7	67.2 / 86.1	62.3 / 81.7	58.4 / 78.3	61.6 / 81.4	55.9 / 76.5	59.8 / 79.2
BLEURT	59.3 / 77.3	59.9 / 79.2	69.5 / 87.8	64.4 / 83.5	61.3 / 81.1	62.9 / 82.4	60.2 / 81.4	62.5 / 81.8

Table 2: Agreement with human ratings on the WMT17 Metrics Shared Task. The metrics are Kendall Tau (τ) and the Pearson correlation (r , the official metric of the shared task), divided by 100.

model	cs-en τ / DA	de-en τ / DA	et-en τ / DA	fi-en τ / DA	ru-en τ / DA	tr-en τ / DA	zh-en τ / DA	avg τ / DA
sentBLEU	20.0 / 22.5	31.6 / 41.5	26.0 / 28.2	17.1 / 15.6	20.5 / 22.4	22.9 / 13.6	21.6 / 17.6	22.8 / 23.2
BERTscore w/ BERT	29.5 / 40.0	39.9 / 53.8	34.7 / 39.0	26.0 / 29.7	27.8 / 34.7	31.7 / 27.5	27.5 / 25.2	31.0 / 35.7
BERTscore w/ roBERTa	31.2 / 41.1	42.2 / 55.5	37.0 / 40.3	27.8 / 30.8	30.2 / 35.4	32.8 / 30.2	29.2 / 26.3	32.9 / 37.1
Meteor++	22.4 / 26.8	34.7 / 45.7	29.7 / 32.9	21.6 / 20.6	22.8 / 25.3	27.3 / 20.4	23.6 / 17.5*	26.0 / 27.0
RUSE	27.0 / 34.5	36.1 / 49.8	32.9 / 36.8	25.5 / 27.5	25.0 / 31.1	29.1 / 25.9	24.6 / 21.5*	28.6 / 32.4
Yisi1	23.5 / 31.7	35.5 / 48.8	30.2 / 35.1	21.5 / 23.1	23.3 / 30.0	26.8 / 23.4	23.1 / 20.9	26.3 / 30.4
Yisi1 SRL 18	23.3 / 31.5	34.3 / 48.3	29.8 / 34.5	21.2 / 23.7	22.6 / 30.6	26.1 / 23.3	22.9 / 20.7	25.7 / 30.4
BLEURTbase -pre	33.0 / 39.0	41.5 / 54.6	38.2 / 39.6	30.7 / 31.1	30.7 / 34.9	32.9 / 29.8	28.3 / 25.6	33.6 / 36.4
BLEURTbase	34.5 / 42.9	43.5 / 55.6	39.2 / 40.5	31.5 / 30.9	31.0 / 35.7	35.0 / 29.4	29.6 / 26.9	34.9 / 37.4
BLEURT -pre	34.5 / 42.1	42.7 / 55.4	39.2 / 40.6	31.4 / 31.6	31.4 / 34.2	33.4 / 29.3	28.9 / 25.6	34.5 / 37.0
BLEURT	35.6 / 42.3	44.2 / 56.7	40.0 / 41.4	32.1 / 32.5	31.9 / 36.0	35.5 / 31.5	29.7 / 26.0	35.6 / 38.1

Table 3: Agreement with human ratings on the WMT18 Metrics Shared Task. The metrics are Kendall Tau (τ) and WMT’s Direct Assessment metrics divided by 100. The star * indicates results that are more than 0.2 percentage points away from the official WMT results (up to 0.4 percentage points away).

5.1 WMT Metrics Shared Task

Datasets and Metrics: We use years 2017 to 2019 of the WMT Metrics Shared Task, to-English language pairs. For each year, we used the official WMT test set, which include several thousand pairs of sentences with human ratings from the news domain. The training sets contain 5,360, 9,492, and 147,691 records for each year. The test sets for years 2018 and 2019 are noisier, as reported by the organizers and shown by the overall lower correlations.

We evaluate the agreement between the automatic metrics and the human ratings. For each year, we report two metrics: Kendall’s Tau τ (for consistency across experiments), and the official WMT metric for that year (for completeness). The official WMT metric is either Pearson’s correlation or a robust variant of Kendall’s Tau called DARR, described in the Appendix. All the numbers come from our own implementation of the benchmark.⁴ Our results are globally consistent with the official results but we report small differences in 2018 and 2019, marked in the tables.

⁴The official scripts are public but they suffer from documentation and dependency issues, as shown by a README file in the 2019 edition which explicitly discourages using them.

Models: We experiment with four versions of BLEURT: BLEURT, BLEURTbase, BLEURT -pre and BLEURTbase -pre. The first two models are based on BERT-large and BERT-base. In the latter two versions, we skip the pre-training phase and fine-tune directly on the WMT ratings. For each year of the WMT shared task, we use the test set from the previous years for training and validation. We describe our setup in further detail in the Appendix. We compare BLEURT to participant data from the shared task and automatic metrics that we ran ourselves. In the former case, we use the the best-performing contestants for each year, that is, chrF++, BEER, Meteor++, RUSE, Yisi1, ESIM and Yisi1-SRL (Mathur et al., 2019). All the contestants use the same WMT training data, in addition to existing sentence or token embeddings. In the latter case, we use Moses sentenceBLEU, BERTscore (Zhang et al., 2020), and MoverScore (Zhao et al., 2019). For BERTscore, we use BERT-large uncased for fairness, and roBERTa (the recommended version) for completeness (Liu et al., 2019). We run MoverScore on WMT 2017 using the scripts published by the authors.

Results: Tables 2, 3, 4 show the results. For years 2017 and 2018, a BLEURT-based metric

model	de-en τ / DA	fi-en τ / DA	gu-en τ / DA	kk-en τ / DA	lt-en τ / DA	ru-en τ / DA	zh-en τ / DA	avg τ / DA
sentBLEU	19.4 / 5.4	20.6 / 23.3	17.3 / 18.9	30.0 / 37.6	23.8 / 26.2	19.4 / 12.4	28.7 / 32.2	22.7 / 22.3
BERTscore w/ BERT	26.2 / 17.3	27.6 / 34.7	25.8 / 29.3	36.9 / 44.0	30.8 / 37.4	25.2 / 20.6	37.5 / 41.4	30.0 / 32.1
BERTscore w/ roBERTa	29.1 / 19.3	29.7 / 35.3	27.7 / 32.4	37.1 / 43.1	32.6 / 38.2	26.3 / 22.7	41.4 / 43.8	32.0 / 33.6
ESIM	28.4 / 16.6	28.9 / 33.7	27.1 / 30.4	38.4 / 43.3	33.2 / 35.9	26.6 / 19.9	38.7 / 39.6	31.6 / 31.3
YiSi1 SRL 19	26.3 / 19.8	27.8 / 34.6	26.6 / 30.6	36.9 / 44.1	30.9 / 38.0	25.3 / 22.0	38.9 / 43.1	30.4 / 33.2
BLEURTbase -pre	30.1 / 15.8	30.4 / 35.4	26.8 / 29.7	37.8 / 41.8	34.2 / 39.0	27.0 / 20.7	40.1 / 39.8	32.3 / 31.7
BLEURTbase	31.0 / 16.6	31.3 / 36.2	27.9 / 30.6	39.5 / 44.6	35.2 / 39.4	28.5 / 21.5	41.7 / 41.6	33.6 / 32.9
BLEURT -pre	31.1 / 16.9	31.3 / 36.5	27.6 / 31.3	38.4 / 42.8	35.0 / 40.0	27.5 / 21.4	41.6 / 41.4	33.2 / 32.9
BLEURT	31.2 / 16.9	31.7 / 36.3	28.3 / 31.9	39.5 / 44.6	35.2 / 40.6	28.3 / 22.3	42.7 / 42.4	33.8 / 33.6

Table 4: Agreement with human ratings on the WMT19 Metrics Shared Task. The metrics are Kendall Tau (τ) and WMT’s Direct Assessment metrics divided by 100. All the values reported for Yisi1_SRL and ESIM fall within 0.2 percentage of the official WMT results.

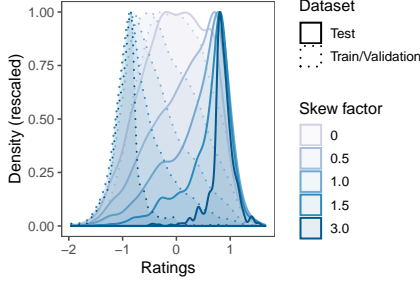


Figure 1: Distribution of the human ratings in the train/validation and test datasets for different skew factors.

dominates the benchmark for each language pair (Tables 2 and 3). BLEURT and BLEURTbase are also competitive for year 2019: they yield the best results for all language pairs on Kendall’s Tau, and they come first for 3 out of 7 pairs on DARR. As expected, BLEURT dominates BLEURTbase in the majority of cases. Pre-training consistently improves the results of BLEURT and BLEURTbase. We observe the largest effect on year 2017, where it adds up to 7.4 Kendall Tau points for BLEURTbase (zh-en). The effect is milder on years 2018 and 2019, up to 2.1 points (tr-en, 2018). We explain the difference by the fact that the training data used for 2017 is smaller than the datasets used for the following years, so pre-training is likelier to help. In general pre-training yields higher returns for BERT-base than for BERT-large—in fact, BLEURTbase with pre-training is often better than BLEURT without.

Takeaways: Pre-training delivers consistent improvements, especially for BLEURT-base. BLEURT yields state-of-the-art performance for all years of the WMT Metrics Shared task.

5.2 Robustness to Quality Drift

We assess our claim that pre-training makes BLEURT robust to quality drifts, by constructing

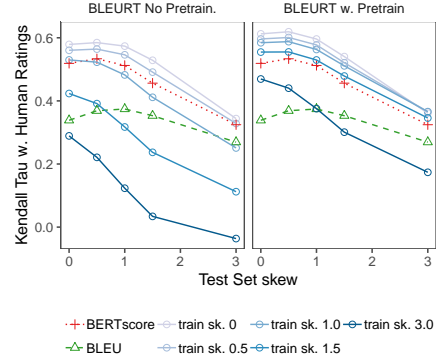


Figure 2: Agreement between BLEURT and human ratings for different skew factors in train and test.

a series of tasks for which it is increasingly pressured to extrapolate. All the experiments that follow are based on the WMT Metrics Shared Task 2017, because the ratings for this edition are particularly reliable.⁵

Methodology: We create increasingly challenging datasets by sub-sampling the records from the WMT Metrics shared task, keeping low-rated translations for training and high-rated translations for test. The key parameter is the *skew factor* α , that measures how much the training data is left-skewed and the test data is right-skewed. Figure 1 demonstrates the ratings distribution that we used in our experiments. The training data shrinks as α increases: in the most extreme case ($\alpha = 3.0$), we use only 11.9% of the original 5,344 training records. We give the full detail of our sampling methodology in the Appendix.

We use BLEURT with and without pre-training and we compare to Moses sentBLEU and BERTscore. We use BERT-large uncased for both BLEURT and BERTscore.

⁵The organizers managed to collect 15 adequacy scores for each translation, and thus the ratings are almost perfectly repeatable (Bojar et al., 2017)

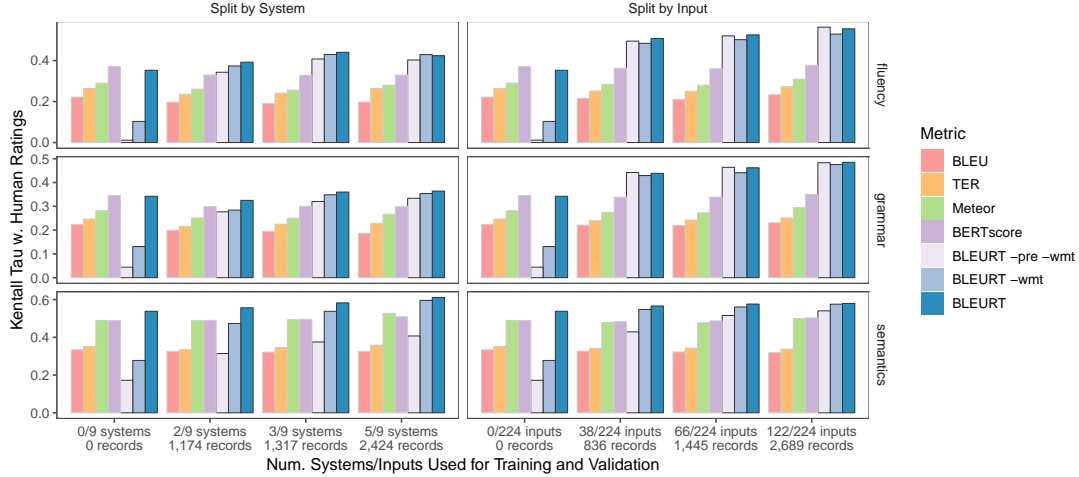


Figure 3: Absolute Kendall Tau of BLEU, Meteor, and BLEURT with human judgements on the WebNLG dataset, varying the size of the data used for training and validation.

Results: Figure 2 presents BLEURT’s performance as we vary the train and test skew independently. Our first observation is that the agreements fall for all metrics as we increase the test skew. This effect was already described in the 2019 WMT Metrics report (Ma et al., 2019). A common explanation is that the task gets more difficult as the ratings get closer—it is easier to discriminate between “good” and “bad” systems than to rank “good” systems.

Training skew has a disastrous effect on BLEURT without pre-training: it is below BERTscore for $\alpha = 1.0$, and it falls under sentBLEU for $\alpha \geq 1.5$. Pre-trained BLEURT is much more robust: the only case in which it falls under the baselines is $\alpha = 3.0$, the most extreme drift, for which incorrect translations are used for train while excellent ones for test.

Takeaways: Pre-training makes BLEURT significantly more robust to quality drifts.

5.3 WebNLG Experiments

In this section, we evaluate BLEURT’s performance on three tasks from a data-to-text dataset, the WebNLG Challenge 2017 (Shimorina et al., 2019). The aim is to assess BLEURT’s capacity to adapt to new tasks with limited training data.

Dataset and Evaluation Tasks: The WebNLG challenge benchmarks systems that produce natural language description of entities (e.g., buildings, cities, artists) from sets of 1 to 5 RDF triples. The organizers released the human assessments for 9 systems over 223 inputs, that is, 4,677 sentence

pairs in total (we removed null values). Each input comes with 1 to 3 reference descriptions. The submissions are evaluated on 3 aspects: semantics, grammar, and fluency. We treat each type of rating as a separate modeling task. The data has no natural split between train and test, therefore we experiment with several schemes. We allocate 0% to about 50% of the data to training, and we split on both the evaluated systems or the RDF inputs in order to test different generalization regimes.

Systems and Baselines: BLEURT-pre-wmt, is a public BERT-large uncased checkpoint directly trained on the WebNLG ratings. BLEURT-wmt was first pre-trained on synthetic data, then fine-tuned on WebNLG data. BLEURT was trained in three steps: first on synthetic data, then on WMT data (16-18), and finally on WebNLG data. When a record comes with several references, we run BLEURT on each reference and report the highest value (Zhang et al., 2020).

We report four baselines: BLEU, TER, Meteor, and BERTscore. The first three were computed by the WebNLG competition organizers. We ran the latter one ourselves, using BERT-large uncased for a fair comparison.

Results: Figure 3 presents the correlation of the metrics with human assessments as we vary the share of data allocated to training. The more pre-trained BLEURT is, the quicker it adapts. The vanilla BERT approach BLEURT-pre-wmt requires about a third of the WebNLG data to dominate the baselines on the majority of tasks, and it still lags behind on *semantics* (split by system). In

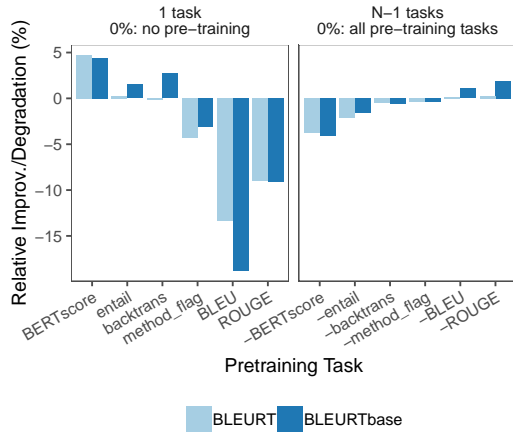


Figure 4: Improvement in Kendall Tau on WMT 17 varying the pre-training tasks.

contrast, BLEURT `-wmt` is competitive with as little as 836 records, and BLEURT is comparable with BERTscore with zero fine-tuning.

Takeaways: Thanks to pre-training, BLEURT can quickly adapt to the new tasks. BLEURT fine-tuned twice (first on synthetic data, then on WMT data) provides acceptable results on all tasks without training data.

5.4 Ablation Experiments

Figure 4 presents our ablation experiments on WMT 2017, which highlight the relative importance of each pre-training task. On the left side, we compare BLEURT pre-trained on a single task to BLEURT without pre-training. On the right side, we compare full BLEURT to BLEURT pre-trained on all tasks except one. Pre-training on BERTscore, entailment, and the backtranslation scores yield improvements (symmetrically, ablating them degrades BLEURT). Oppositely, BLEU and ROUGE have a negative impact. We conclude that pre-training on high quality signals helps BLEURT, but that metrics that correlate less well with human judgment may in fact harm the model.⁶

6 Related Work

The WMT shared metrics competition (Bojar et al., 2016; Ma et al., 2018, 2019) has inspired

⁶Do those results imply that BLEU and ROUGE should be removed from future versions of BLEURT? Doing so may indeed yield slight improvements on the WMT Metrics 2017 shared task. On the other hand the removal may hurt future tasks in which BLEU or ROUGE actually correlate with human assessments. We therefore leave the question open.

the creation of many learned metrics, some of which use regression or deep learning (Stanojevic and Simaan, 2014; Ma et al., 2017; Shimanaka et al., 2018; Chen et al., 2017; Mathur et al., 2019). Other metrics have been introduced, such as the recent MoverScore (Zhao et al., 2019) which combines contextual embeddings and Earth Mover’s Distance. We provide a head-to-head comparison with the best performing of those in our experiments. Other approaches do not attempt to estimate quality directly, but use information extraction or question answering as a proxy (Wiseman et al., 2017; Goodrich et al., 2019; Eyal et al., 2019). Those are complementary to our work.

There has been recent work that uses BERT for evaluation. BERTScore (Zhang et al., 2020) proposes replacing the hard n-gram overlap of BLEU with a soft-overlap using BERT embeddings. We use it in all our experiments. Bertr (Mathur et al., 2019) and YiSi (Mathur et al., 2019) also make use of BERT embeddings to capture similarity. SumQE (Xenouelas et al., 2019) fine-tunes BERT for quality estimation as we describe in Section 3. Our focus is different—we train metrics that are not only state-of-the-art in conventional IID experimental setups, but also robust in the presence of scarce and out-of-distribution training data. To our knowledge no existing work has explored pre-training and extrapolation in the context of NLG.

Previous studies have used noising for referenceless evaluation (Dušek et al., 2019). Noisy pre-training has also been proposed before for other tasks such as paraphrasing (Wieting et al., 2016; Tomar et al., 2017) but generally not with synthetic data. Generating synthetic data via paraphrases and perturbations has been commonly used for generating adversarial examples (Jia and Liang, 2017; Iyyer et al., 2018; Belinkov and Bisk, 2018; Ribeiro et al., 2018), an orthogonal line of research.

7 Conclusion

We presented BLEURT, a reference-based text generation metric for English. Because the metric is trained end-to-end, BLEURT can model human assessment with superior accuracy. Furthermore, pre-training makes the metrics robust particularly robust to both domain and quality drifts. Future research directions include multilingual NLG evaluation, and hybrid methods involving both humans and classifiers.

Acknowledgments

Thanks to Eunsol Choi, Nicholas FitzGerald, Jacob Devlin, and to the members of the Google AI Language team for the proof-reading, feedback, and suggestions. We also thank Madhavan Kidambi and Ming-Wei Chang, who implemented blank-filling with BERT.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.
- Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*.
- Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of ICLR*.
- Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the wmt17 metrics shared task. In *Proceedings of WMT*.
- Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the wmt16 metrics shared task. In *Proceedings of WMT*.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *Proceedings of EMNLP*.
- Arun Tejasvi Chaganty, Stephen Mussman, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evaluation. *Proceedings of ACL*.
- Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. *Proceedings of ACL*.
- Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of NAACL HLT*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL HLT*.
- Ondřej Dušek, Karin Sevegnani, Ioannis Konstas, and Verena Rieser. 2019. Automatic quality estimation for natural language generation: Ranting (jointly rating and ranking). *Proceedings of INLG*.
- Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of NAACL HLT*.
- Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2015. From captions to visual concepts and back. In *Proceedings of CVPR*.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings NAACL HLT*.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of INLG*.
- Ben Goodrich, Mohammad Ahmad Saleh, Peter Liu, and Vinay Rao. 2019. Assessing the factual accuracy of text generation. In *Proceedings of ACM SIGKDD*.
- Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *Proceedings of NAACL HLT*.
- Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *Proceedings of EMNLP*.
- Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.
- Karen Kukich. 1983. Design of a knowledge-based report generator. In *Proceedings of ACL*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out*.
- Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *Proceedings of EMNLP*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.
- Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the wmt18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of WMT*.
- Qingsong Ma, Yvette Graham, Shugen Wang, and Qun Liu. 2017. Blend: a novel combined mt metric based on direct assessment-casict-dcu submission to wmt17 metrics task. In *Proceedings of WMT*.
- Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of WMT*.

- Inderjeet Mani. 1999. *Advances in automatic text summarization*. MIT press.
- Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of ACL*.
- Kathleen McKeown. 1992. *Text generation*. Cambridge University Press.
- Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. *Proceedings of EMNLP*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of ACL*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. *Proceedings of ACL*.
- Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. Ruse: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of WMT*.
- Anastasia Shimorina, Claire Gardent, Shashi Narayan, and Laura Perez-Beltrachini. 2019. Webnlg challenge: Human evaluation results. Technical report.
- Ronnie W Smith and D Richard Hipp. 1994. *Spoken natural language dialog systems: A practical approach*. Oxford University Press.
- Milos Stanojevic and Khalil Simaan. 2014. Beer: Better evaluation as ranking. In *Proceedings of WMT*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*.
- Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv:1910.08684*.
- Gaurav Singh Tomar, Thyago Duque, Oscar Täckström, Jakob Uszkoreit, and Dipanjan Das. 2017. Neural paraphrase identification of questions with noisy pretraining. *Proceedings of the First Workshop on Subword and Character Level Models in NLP*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*.
- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *Proceedings of ICML*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Proceedings of ICLR*.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. *Proceedings of ICLR*.
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *Proceedings of NAACL HLT*.
- Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. *Proceedings of EMNLP*.
- Stratos Xenouelas, Prodromos Malakasiotis, Marianna Apidianaki, and Ion Androutsopoulos. 2019. Sumqe: a bert-based summary quality estimation model supplementary material. In *Proceedings of EMNLP*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Proceedings of ICLR*.
- Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *Proceedings of EMNLP*.

A Implementation Details of the Pre-Training Phase

This section provides implementation details for some of the pre-training techniques described in the main paper.

A.1 Data Generation

Random Masking: We use two masking strategies. The first strategy samples random words in the sentence and it replaces them with masks (one for each token). Thus, the masks are scattered across the sentence. The second strategy creates contiguous sequences: it samples a start position s , a length l (uniformly distributed), and it masks all the tokens spanned by words between positions s and $s + l$. In both cases, we use up to 15 masks per sentence. Instead of running the language model once and picking the most likely token at each position, we use beam search (the beam size 8 by default). This enforces consistency and avoids repeated sequences, e.g., “ , , , ”.

Backtranslation: Consider English and French. Given a forward translation model $P_{\text{en} \rightarrow \text{fr}}(z_{\text{fr}}|z_{\text{en}})$ and backward translation model $P_{\text{fr} \rightarrow \text{en}}(z_{\text{en}}|z_{\text{fr}})$, we generate \tilde{z} as follows:

$$\tilde{z} = \arg \max_{z_{\text{en}}} (P_{\text{fr} \rightarrow \text{en}}(z_{\text{en}}|z_{\text{fr}}^*))$$

where $z_{\text{fr}}^* = \arg \max_{z_{\text{fr}}} (P_{\text{fr} \rightarrow \text{en}}(z_{\text{fr}}|z))$. For the translations, we use a Transformer model (Vaswani et al., 2017), trained on English-German with the `tensor2tensor` framework.⁷

Word dropping: Given a synthetic example (z, \tilde{z}) we generate a pair (z, \tilde{z}') , by randomly dropping words from \tilde{z} . We draw the number of words to drop uniformly, up to the length of the sentence. We apply this transformation on about 30% of the data generated with the previous method.

A.2 Pre-Training Tasks

We now provide additional details on the signals we used for pre-training.

Automatic Metrics: As shown in the table, we use three types of signals: BLEU, ROUGE, and BERTscore. For BLEU, we used the original Moses SENTENCEBLEU⁸ implementation, using the Moses tokenizer and the default parameters. For ROUGE, we used the `seq2seq` implementation of ROUGE-N.⁹ We used a custom implementation of BERTSCORE, based on BERT-large uncased. ROUGE and BERTscore return three scores: precision, recall, and F-score. We use all three quantities.

Backtranslation Likelihood: We compute all the losses using custom Transformer model (Vaswani et al., 2017), trained on two language pairs (English-French and English-German) with the `tensor2tensor` framework.

Normalization: All the regression labels are normalized before training.

A.3 Modeling

Setting the weights of the pre-training tasks:

We set the weights γ_k with grid search, optimizing BLEURT’s performance on WMT 17’s

⁷<https://github.com/tensorflow/tensor2tensor>

⁸<https://github.com/moses-smt/mosesdecoder/blob/master/mert/sentence-bleu.cpp>

⁹<https://github.com/google/seq2seq/blob/master/seq2seq/metrics/rouge.py>

validation set. To reduce the size of the grid, we make groups of pre-training tasks that share the same weights: $(\tau_{\text{BLEU}}, \tau_{\text{ROUGE}}, \tau_{\text{BERTscore}})$, $(\tau_{\text{en-fr}, z|\tilde{z}}, \tau_{\text{en-fr}, \tilde{z}|z}, \tau_{\text{en-de}, z|\tilde{z}}, \tau_{\text{en-de}, \tilde{z}|z})$, and $(\tau_{\text{entail}}, \tau_{\text{backtran_flag}})$.

B Experiments–Supplementary Material

B.1 Training Setup for All Experiments

We use BERT’s public checkpoints¹⁰ with Adam (the default optimizer), learning rate 1e-5, and batch size 32. Unless specified otherwise, we use 800,00 training steps for pre-training and 40,000 steps for fine-tuning. We run training and evaluation in parallel: we run the evaluation every 1,500 steps and store the checkpoint that performs best on a held-out validation set (more details on the data splits and our choice of metrics in the following sections). We use Google Cloud TPUs v2 for learning, and Nvidia Tesla V100 accelerators for evaluation and test. Our code uses Tensorflow 1.15 and Python 2.7.

B.2 WMT Metric Shared Task

Metrics. The metrics used to compare the evaluation systems vary across the years. The organizers use Pearson’s correlation on standardized human judgments across all segments in 2017, and a custom variant of Kendall’s Tau named “DARR” on raw human judgments in 2018 and 2019. The latter metrics operates as follows. The organizers gather all the translations for the same reference segment, they enumerate all the possible pairs $(\text{translation}_1, \text{translation}_2)$, and they discard all the pairs which have a “similar” score (less than 25 points away on a 100 points scale). For each remaining pair, they then determine which translation is the best according both human judgment and the candidate metric. Let $|\text{Concordant}|$ be the number of pairs on which the NLG metrics agree and $|\text{Discordant}|$ be those on which they disagree, then the score is computed as follows:

$$\frac{|\text{Concordant}| - |\text{Discordant}|}{|\text{Concordant}| + |\text{Discordant}|}$$

The idea behind the 25 points filter is to make the evaluation more robust, since the judgments collected for WMT 2018 and 2019 are noisy. Kendall’s Tau is identical, but it does not use the filter.

¹⁰<https://github.com/google-research/bert>

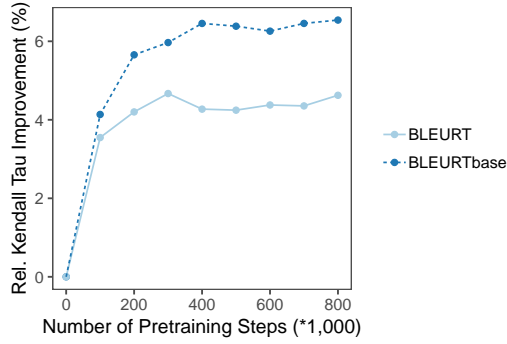


Figure 5: Improvement in Kendall Tau accuracy on all language pairs of the WMT Metrics Shared Task 2017, varying the number of pre-training steps. 0 steps corresponds to 0.555 Kendall Tau for BLEURTbase and 0.580 for BLEURT.

Training setup. To separate training and validation data, we set aside a fixed ratio of records in such a way that there is no “leak” between the datasets (i.e., train and validation records that share the same source). We use 10% of the data for validation for years 2017 and 2018, and 5% for year 2019. We report results for the models that yield the highest Kendall Tau across all records on validation data. The weights associated to each pretraining task (see our Modeling section) are set with grid search, using the train/validation setup of WMT 2017.

Baselines. we use three metrics: the Moses implementation of sentenceBLEU,¹¹ BERTscore,¹² and MoverScore,¹³ which are all available online. We run the Moses tokenizer on the reference and candidate segments before computing sentenceBLEU.

B.3 Robustness to Quality Drift

Data Re-sampling Methodology: We sample the training and test separately, as follows. We split the data in 10 bins of equal size. We then sample each record in the dataset with probabilities $\frac{1}{B^\alpha}$ and $\frac{1}{(11-B)^\alpha}$ for train and test respectively, where B is the bin index of the record between 1 and 10, and α is a predefined *skew factor*. The skew factor α controls the drift: a value of 0 has no effect (the ratings are centered around 0), and value of 3.0 yields extreme differences. Note that

the sizes of the datasets decrease as α increases: we use 50.7%, 30.3%, 20.4%, and 11.9% of the original 5,344 training records for $\alpha = 0.5, 1.0, 1.5$, and 3.0 respectively.

B.4 Ablation Experiment—How Much Pre-Training Time is Necessary?

To understand the relationship between pre-training time and downstream accuracy, we pre-train several versions of BLEURT and we fine-tune them on WMT17 data, varying the number of pre-training steps. Figure 5 presents the results. Most gains are obtained during the first 400,000 steps, that is, after about 2 epochs over our synthetic dataset.

¹¹<https://github.com/moses-smt/mosesdecoder/blob/master/mert/sentence-bleu.cpp>

¹²https://github.com/Tiiiger/bert_score

¹³<https://github.com/AIPHES/emnlp19-moverscore>