BARTScore: Evaluating Generated Text as Text Generation

Weizhe Yuan
Carnegie Mellon University
weizhey@cs.cmu.edu

Graham Neubig
Carnegie Mellon University
gneubig@cs.cmu.edu

Pengfei Liu *
Carnegie Mellon University
pliu3@cs.cmu.edu

Abstract

A wide variety of NLP applications, such as machine translation, summarization, and dialog, involve text generation. One major challenge for these applications is how to *evaluate* whether such generated texts are actually fluent, accurate, or effective. In this work, we conceptualize the *evaluation of generated text as a text generation problem*, modeled using pre-trained sequence-to-sequence models. The general idea is that models trained to convert the generated text to/from a reference output or the source text will achieve higher scores when the generated text is better. We operationalize this idea using BART [32], an encoder-decoder based pre-trained model, and propose a metric BARTScore with a number of variants that can be flexibly applied in an unsupervised fashion to evaluation of text from different perspectives (e.g. informativeness, fluency, or factuality). BARTScore is conceptually simple and empirically effective. It can outperform existing top-scoring metrics in 16 of 22 test settings, covering evaluation of 16 datasets (e.g., machine translation, text summarization) and 7 different perspectives (e.g., informativeness, factuality). Code to calculate BARTScore is available at https://github.com/neulab/BARTScore, and we have released an interactive leaderboard for meta-evaluation at http://explainaboard.nlpedia.ai/leaderboard/task-meval/ on the ExplainaBoard platform [38], which allows us to interactively understand the strengths, weaknesses, and complementarity of each metric.

1 Introduction

One defining feature of recent NLP models is the use of neural representations trained on raw text, using unsupervised objectives such as language modeling [6,54], or denoising autoencoding [9,32,55]. By learning to predict the words or sentences in natural text, these models simultaneously learn to extract features that not only benefit mainstream NLP tasks such as information extraction [23,37], question answering [1,26], text summarization [40,78] but also have proven effective in development of automatic metrics for evaluation of text generation itself [63,66]. For example, BERTScore [76] and MoverScore [77] take features extracted by BERT [9] and apply unsupervised matching functions to compare system outputs against references. Other works build supervised frameworks that use the extracted features to learn to rank [57] or regress [63] to human evaluation scores.

However, in the context of generation evaluation, one may note that there is a decided *disconnect between how models are pre-trained using text generation objectives and how they are used as down-stream feature extractors*. This leads to potential under-utilization of the pre-trained model parameters. For example, the output prediction layer is not used at all in this case. This disconnect is particularly striking because of the close connection between the pre-training objectives and the generation tasks we want to evaluate.

[*] Corresponding author.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

In this paper, we instead argue for a formulation of *evaluation of generated text as a text generation problem*, directly evaluating text through the lens of its probability of being generated from or generating other textual inputs and outputs. This is a better match with the underlying pre-training tasks and allows us to more fully take advantage of the parameters learned during the pre-training phase. We solve the modeling problem with a pre-trained sequence-to-sequence (seq2seq) model, specifically BART [32], and devise a metric named BARTSCORE, which has the following characteristics: (1) BARTSCORE is parameter- and data-efficient. Architecturally there are no extra parameters beyond those used in pre-training itself, and it is an unsupervised metric that doesn't require human judgments to train. (2) BARTSCORE can better support evaluation of generated text from different perspectives (e.g., informativeness, coherence, factuality, §4) by adjusting the inputs and outputs of the conditional text generation problem, as we demonstrate in §3.2. This is in contrast to most previous work, which mostly examines correlation of the devised metrics with output quality from a limited number of perspectives. (3) BARTSCORE can be further enhanced by (i) providing textual prompts that bring the evaluation task closer to the pre-training task, or (ii) updating the underlying model by fine-tuning BART based on downstream generation tasks (e.g., text summarization).

Experimentally, we evaluate different variants of BARTSCORE from 7 perspectives on 16 datasets. BARTSCORE achieves the best performance in 16 of 22 test settings against existing top-scoring metrics. Empirical results also show the effectiveness of the *prompting* strategy supported by BARTSCORE. For example, simply adding the phrase "`such as`" to the translated text when using BARTSCORE can lead to a 3% point absolute improvement in correlation on "German-English" machine translation (MT) evaluation. Additional analysis shows that BARTSCORE is more robust when dealing with high-quality texts generated by top-performing systems.

2 Preliminaries

2.1 Problem Formulation

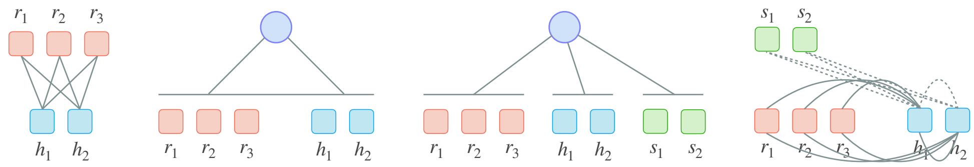
As stated above, our goal is to assess the quality of generated text [3,47]. In this work, we focus on conditional text generation (e.g., machine translation), where the goal is to generate a *hypothesis* ($\boldsymbol{h} = h_1, \cdots, h_m$) based on a given *source* text ($\boldsymbol{s} = s_1, \cdots, s_n$). Commonly, one or multiple human-created *references* ($\boldsymbol{r} = r_1, \cdots, r_l$) are provided to aid this evaluation.

2.2 Gold-standard Human Evaluation

In general, the gold-standard method for evaluating such texts is still human evaluation, where human annotators assess the generated texts' quality. This evaluation can be done from perspectives, and we list a few common varieties below (all are investigated in §4):

1. **Informativeness** (INFO): How well the generated hypothesis captures the key ideas of the source text [18].
2. **Relevance** (REL): How consistent the generated hypothesis is with respect to the source text [19].
3. **Fluency** (FLU): Whether the text has no formatting problems, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read [13].
4. **Coherence** (COH): Whether the text builds from sentence to sentence to a coherent body of information about a topic [7].
5. **Factuality** (FAC): Whether the generated hypothesis contains only statements entailed by the source text [30].
6. **Semantic Coverage** (COV): How many semantic content units from reference texts are covered by the generated hypothesis [50].
7. **Adequacy** (ADE): Whether the output conveys the same meaning as the input sentence, and none of the message is lost, added, or distorted [29].

Most existing evaluation metrics were designed to cover a small subset of these perspectives. For example, BLEU [51] aims to capture the adequacy and fluency of translations, while ROUGE [36] was designed to match the semantic coverage metric. Some metrics, particularly trainable ones, can perform evaluation from different perspectives but generally require maximizing correlation with each type of judgment separately [8].


r_1 r_2 r_3
h_1 h_2
r_1 r_2 r_3 h_1 h_2
r_1 r_2 r_3 h_1 h_2 s_1 s_2
s_1 s_2
r_1 r_2 r_3 h_1 h_2

(a) Matching (b) Regression (c) Ranking (d) Generation

Figure 1: Evaluation metrics as different tasks, where s_i, h_i and r_j represent *source*, *hypothesis* and *reference* words respectively.

As we describe more in §4, BARTSCORE can evaluate text from the great majority of these perspectives, significantly expanding its applicability compared to these metrics.

2.3 Evaluation as Different Tasks

There is a recent trend that leverages neural models for automated evaluation in different ways, as shown in Fig. 1. We first elaborate on their characteristics by highlighting differences in task formulation and evaluation perspectives.

T1: Unsupervised Matching. Unsupervised matching metrics aim to measure the semantic equivalence between the reference and hypothesis by using a *token-level matching* functions in distributed representation space, such as BERTScore [76], MoverScore [77] or discrete string space like ROUGE [35], BLEU [51], CHRF [53]. Although similar matching functions can be used to assess the quality beyond semantic equivalence (e.g, factuality, a relationship between source text and hypothesis), to our knowledge prior research has not attested to the capability of unsupervised matching methods in this regard; we explore this further in our experiments (Tab. 5).

T2: Supervised Regression. Regression-based models introduce a parameterized regression layer, which would be learned in a supervised fashion to accurately predict human judgments. Examples include recent metrics BLEURT [63], COMET [57] and traditional metrics like S^3 [52], VRM [21].

T3: Supervised Ranking. Evaluation can also be conceived as a ranking problem, where the main idea is to learn a scoring function that assigns a higher score to better hypotheses than to worse ones. Examples include COMET [57] and BEER [65], where COMET focuses the machine translation task and relies on human judgments to tune parameters in ranking or regression layers, and BEER combines many simple features in a tunable linear model of MT evaluation metrics.

T4: Text Generation. In this work, we formulate evaluating generated text as a text generation task from pre-trained language models. The basic idea is that a high-quality hypothesis will be easily generated based on source or reference text or vice-versa. This has not been covered as extensively in previous work, with one notable exception being PRISM [66]. Our work differs from PRISM in several ways: (i) PRISM formulates evaluation as a paraphrasing task, whose definition that two texts are with the same meaning limits its applicable scenarios, like factuality evaluation in text summarization that takes source documents and generated summaries as input whose semantic space are different. (ii) PRISM trained a model from scratch on parallel data while BARTSCORE is based on open-sourced pre-trained seq2seq models. (iii) BARTSCORE supports prompt-based learning [60, 64] which hasn't been examined in PRISM.

3 BARTScore

3.1 Sequence-to-Sequence Pre-trained Models

Although pre-trained models differ along different axes, one of the main axes of variation is the training objective, with two main variants: language modeling objectives (e.g., masked language modeling [9]) and seq2seq objectives [55]. In particular, seq2seq pre-trained models are particularly well-suited to conditioned generation tasks since they consist of both an encoder and a decoder, and predictions are made auto-regressively [32]. In this work, we operationalize our idea by using

BART [32] as our backbone due to its superior performance in text generation [12,42,72]. We also report preliminary experiments comparing BART with T5 [55] and PEGASUS [75] in the Appendix.

Given a seq2seq model parameterized by θ, a source sequence containing n tokens $\mathbf{x} = \{x_1, \cdots, x_n\}$ and a target sequence containing m tokens $\mathbf{y} = \{y_1, \cdots, y_m\}$. We can factorize the generation probability of $\mathbf{y}$ conditioned on $\mathbf{x}$ as follows:

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_{t=1}^{m} p(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{x}, \theta) \tag{1}$$

By exploring these probabilities, we design metrics that can gauge the quality of the generated text.

3.2 BARTScore

The most general form of our proposed BARTScore is shown in Eq. 2, where we use the weighted log probability of one text $\mathbf{y}$ given another text $\mathbf{x}$. The weights are used to put different emphasis on different tokens, which can be instantiated using different methods like Inverse Document Frequency (IDF) [25] etc. In our work, we weigh each token equally.[2]

$$\text{BARTScore} = \sum_{t=1}^{m} \omega_t \log p(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{x}, \theta) \tag{2}$$

Due to its generation task-based formulation and ability to utilize the entirety of BART's pre-trained parameters, BARTScore can be flexibly used in different evaluation scenarios. We specifically present four methods for using BARTScore based on different generation directions, which are,

- *Faithfulness* ($\boldsymbol{s} \rightarrow \boldsymbol{h}$): from source document to hypothesis $p(\boldsymbol{h}|\boldsymbol{s}, \theta)$. This direction measures how likely it is that the hypothesis could be generated based on the source text. Potential application scenarios are factuality and relevance introduced in §2.2. This measure can also be used for estimating measures of the quality of only the target text, such as coherence and fluency (§2.2).
- *Precision* ($\boldsymbol{r} \rightarrow \boldsymbol{h}$): from reference text to system-generated text $p(\boldsymbol{h}|\boldsymbol{r}, \theta)$. This direction assesses how likely the hypothesis could be constructed based on the gold reference and is suitable for the precision-focused scenario.
- *Recall* ($\boldsymbol{h} \rightarrow \boldsymbol{r}$): from system-generated text to reference text $p(\boldsymbol{r}|\boldsymbol{h}, \theta)$. This version quantifies how easily a gold reference could be generated by the hypothesis and is suitable for pyramid-based evaluation (i.e., semantic coverage introduced in §2.2) in summarization task since pyramid score measures fine-grained Semantic Content Units (SCUs) [50] covered by system-generated texts.
- $\mathcal{F}$ score ($\boldsymbol{r} \leftrightarrow \boldsymbol{h}$): Consider both directions and use the arithmetic average of *Precision* and *Recall* ones. This version can be broadly used to evaluate the semantic overlap (informativeness, adequacy detailed in §2.2) between reference texts and generated texts.

3.3 BARTScore Variants

We also investigate two extensions to BARTScore: (i) changing $\mathbf{x}$ and $\mathbf{y}$ through prompting, which can *bring the evaluation task closer to the pre-training task.* (ii) changing θ by considering different fine-tuning tasks, which can *bring the pre-training domain closer to the evaluation task.*

3.3.1 Prompt

Prompting is a practice of adding short phrases to the input or output to encourage pre-trained models to perform specific tasks, which has been proven effective in several other NLP scenarios [24,58,59,61,64]. The generative formulation of BARTScore makes it relatively easy to incorporate these insights here as well; we name this variant BARTScore-Prompt.

Given a prompt of l tokens $\mathbf{z} = \{z_1, \cdots, z_l\}$, we can either (i) append it to the source text, in which case we get $\mathbf{x}' = \{x_1, \cdots, x_n, z_1, \cdots, z_l\}$, and calculate the score based on this new source text using Eq.2. or (ii) prepend it to the target text, getting $\mathbf{y}' = \{z_1, \cdots, z_l, y_1, \cdots, y_m\}$. Then we can also use Eq.2 given the new target text.

[2] We have tried several other weighting schemes, including: (i) uniform weighting while ignoring stop words. (ii) IDF weighting. (iii) using the prior probability of each target token (calculated within the target sequence) as the weighting factor. However, none of those outperformed the uniform weighting scheme.

3.3.2 Fine-tuning Task

Different from BERT-based metrics, which typically use classification-based tasks (e.g., natural language inference) [68] to fine-tune, BARTScore can be fine-tuned using generation-based tasks, which will make the pre-training domain closer to the evaluation task. In this paper, we explore two downstream tasks. (1) Summarization. We use BART fine-tuned on `CNNDM` dataset [20], which is available off-the-shelf in Huggingface Transformers [71]. (2) Paraphrasing. We continue fine-tuning BART from (1) on `ParaBank2` dataset [22], which contains a large paraphrase collection. We used a random subset of 30,000 data and fine-tuned for one epoch with a batch size of 20 and a learning rate of $5e^{-5}$. We used two 2080Ti GPUs, and the training time is less than one hour.

4 Experiment

This section aims to evaluate the reliability of different automated metrics, which is commonly achieved by quantifying how well different metrics correlate with human judgments using measures (e.g., Spearman Correlation [73]) defined below (§4.1.2).

4.1 Baselines and Datasets

4.1.1 Evaluation Metrics

We comprehensively examine metrics outlined in §2.3, which either require human judgments to train (i.e., *supervised metrics*): **COMET** [57], **BLEURT** [63], or are human judgment-free (i.e., *unsupervised*): **BLEU** [51] **ROUGE-1 and ROUGE-2, ROUGE-L**, **CHRF** [53], **PRISM** [66], **MoverScore** [77], **BERTScore** [76]. The detailed comparisons of those metrics can be found in Appendix. We use the official code for each metric.

4.1.2 Measures for Meta Evaluation

Pearson Correlation [15] measures the linear correlation between two sets of data. **Spearman Correlation** [73] assesses the monotonic relationships between two variables. **Kendall's Tau** [27] measures the ordinal association between two measured quantities. **Accuracy**, in our experiments, measures the percentage of correct ranking between factual texts and non-factual texts. We follow previous works in the choices of measures for different datasets to make a fair comparison.

4.1.3 Datasets

The datasets we use are summarized in Tab. 1. We consider three different tasks: summarization (SUM), machine translation (MT), and data-to-text (D2T).

Machine Translation We obtain the source language sentences, machine-translated texts and reference texts from the `WMT19` metrics shared task [44]. We use the DARR corpus and consider 7 language pairs, which are `de-en`, `fi-en`, `gu-en`, `kk-en`, `lt-en`, `ru-en`, `zh-en`.

Text Summarization (1) `REALSumm` [4] is a meta-evaluation dataset for text summarization which measures *pyramid recall* of each system-generated summary. (2) `SummEval` [13] is a collection of human judgments of model-generated summaries on the `CNNDM` dataset annotated by both expert judges and crowd-source workers. Each system generated summary is gauged through the lens of *coherence*, *consistency*, *fluency* and *relevance*.[3] (3) `NeR18` The

Table 1: A summary of tasks, datasets, and evaluation perspectives that we have covered in our experiments. Explanation of evaluation perspectives can be found in §2.2.

Tasks	Datasets	Eval. Perspectives
SUM	REALSUM	Cov
	SummEval	Coh Fac Flu Info
	NeR18	Coh Flu Rel Info
	Rank19 QAGS-C QAGS-X	Fac
MT	DE FI GU KK IT RU ZH	Ade Flu
D2T	BAGEL SFHOT SFRES	Info

[3] We rephrase the original "relevance" into "informativeness" and "consistency" into "factuality" based on the descriptions in their paper and our definitions in §2.2.

NEWSROOM dataset [18] contains 60 articles with summaries generated by 7 different methods are annotated with human scores in terms of *coherence*, *fluency*, *informativeness*, *relevance*.

Factuality (1) Rank19 [14] is used to meta-evaluate factuality metrics. It is a collection of 373 triples of a source sentence with two summary sentences, one correct and one incorrect. (2) QAGS20 [67] collected 235 test outputs on CNNDM dataset from [16] and 239 test outputs on XSUM dataset [48] from BART fine-tuned on XSUM. Sentences in each summary are annotated with correctness scores w.r.t. factuality.

Data to Text We consider the following datasets which target utterance generation for spoken dialogue systems. (1) BAGEL [45] provides information about restaurants. (2) SFHOT [70] provides information about hotels in San Francisco. (3) SFRES [70] provides information about restaurants in San Francisco. They contain 202, 398, and 581 samples respectively, each sample consists of one meaning representation, multiple references, and utterances generated by different systems.

4.2 Setup

4.2.1 Prompt Design

To perform prompting, we first need to find proper prompts within a search space. Instead of considering a large discrete search space [64][4] or continuous search space [34], we use simple heuristics to narrow our search space. In particular, we use manually devised seed prompts and gather paraphrases to construct our prompt set.[5] The seed prompts and some examples of paraphrased prompts are shown in Tab. 2. Details are listed in the Appendix.

Table 2: Seed prompts and examples of final prompts. "Number" denotes the size of our final prompt set that was acquired from the seed prompts.

Usage	Number	Seed	Example
$\boldsymbol{s} \rightarrow \boldsymbol{h}$	70	in summary	in short, in a word, to sum up
$\boldsymbol{h} \leftrightarrow \boldsymbol{r}$	34	in other words	to rephrase it, that is to say, i.e.

4.2.2 Settings

Variants. We consider four variants of BARTScore, which are (1) BARTScore, which uses the vanilla BART; (2) BARTScore-cnn, which uses the BART fine-tuned on the summarization dataset CNNDM; (3) BARTScore-cnn-Para, where BART is first fine-tuned on CNNDM, then fine-tuned on ParaBank2. (4) BARTScore-Prompt, which is enhanced by adding prompts.

Selection of Prompts. For the summarization and data-to-text tasks, we use all entries (either all prompts designed for $\boldsymbol{s} \rightarrow \boldsymbol{h}$ or all prompts designed for $\boldsymbol{h} \leftrightarrow \boldsymbol{r}$ depending on the BARTScore usage chosen) in the prompt set by prefixing the decoder input and getting different generation scores (calculated by Eq.2) for each hypothesis based on different prompts. We finally get the score for one hypothesis by taking the average of all its generation scores using different prompts ([24]; details about *prompt ensembling* can be found in the Appendix). For the machine translation task, due to the more expensive computational cost brought by larger text sets, we first use WMT18 [43] as a development set to search for one best prompt and obtain the phrase "Such as", which is then used for the test language pairs.

Selection of BARTScore Usage. Although BARTScore can be used in different ways (shown in §3.2)), in different tasks, they can be chosen based on how targeted evaluation perspectives are defined (described in §2.2) as well as the types of tasks. Specifically, (i) For those datasets whose gold standard human evaluation are obtained based on recall-based pyramid method, we adopt recall-based BARTScore ($\boldsymbol{h} \rightarrow \boldsymbol{r}$). (ii) For those datasets whose human judgments focus on linguistic quality (coherence, fluency) and factual correctness (factuality), or the source and hypothesis texts are in the same modality (i.e., language), we use faithfulness-based BARTScore ($\boldsymbol{s} \rightarrow \boldsymbol{h}$). (iii) For data-to-text and machine translation tasks, to make a fair comparison, we use BARTScore with the F-score version that other existing works [66] have adopted when evaluating generated texts.

[4] We explored this first and found that discovered prompts led to worse performance.

[5] We use the website https://www.wordhippo.com/ to search for synonyms.

Table 3: Kendall's Tau correlation of different metrics on WMT19 dataset. The highest correlation for each language pair achieved by *unsupervised* method is **bold**, and the highest correlation *overall* is underlined. **Avg.** denotes the average correlation achieved by a metric across all language pairs.

	de-en	**fi-en**	**gu-en**	**kk-en**	**lt-en**	**ru-en**	**zh-en**	**Avg.**
SUPERVISED METHODS								
BLEURT	0.174	0.374	0.313	0.372	0.388	0.220	0.436	0.325
COMET	0.219	0.369	0.316	0.378	0.405	0.226	0.462	0.339
UNSUPERVISED METHODS								
BLEU	0.054	0.236	0.194	0.276	0.249	0.115	0.321	0.206
CHRF	0.123	0.292	0.240	0.323	0.304	0.177	0.371	0.261
PRISM	0.199	0.366	**0.320**	0.362	0.382	0.220	0.434	0.326
BERTScore	0.190	0.354	0.292	0.351	0.381	**0.221**	0.430	0.317
BARTSCORE	0.156	0.335	0.273	0.324	0.322	0.167	0.389	0.281
+ CNN	0.190	0.365	0.300	0.348	0.384	0.208	0.425	0.317
+ CNN + Para	0.205†	0.370†	0.316	**0.378**†	**0.386**†	0.219	0.442†	0.331
+ CNN + Para + Prompt	**0.238**‡	**0.374**‡	0.318	0.376†	**0.386**†	0.219	**0.447**‡	**0.337**

Table 4: Spearman correlation of different metrics on three human judgement datasets. For prompt-based learning, we consider adding prompts to the best-performing BARTSCORE (Ω) on each dataset. The highest correlation overall for each aspect on each dataset is **bold**.

	REALSumm	SummEval				NeR18				
	COV	COH	FAC	FLU	INFO	COH	FLU	INFO	REL	Avg.
ROUGE-1	**0.498**	0.167	0.160	0.115	0.326	0.095	0.104	0.130	0.147	0.194
ROUGE-2	0.423	0.184	0.187	0.159	0.290	0.026	0.048	0.079	0.091	0.165
ROUGE-L	0.488	0.128	0.115	0.105	0.311	0.064	0.072	0.089	0.106	0.164
BERTScore	0.440	0.284	0.110	0.193	0.312	0.147	0.170	0.131	0.163	0.217
MoverScore	0.372	0.159	0.157	0.129	0.318	0.161	0.120	0.188	0.195	0.200
PRISM	0.411	0.249	0.345	0.254	0.212	0.573	0.532	0.561	0.553	0.410
BARTSCORE	0.441	0.322†	0.311	0.248	0.264	0.679†	0.670†	0.646†	0.604†	0.465
+ CNN	0.475	**0.448**‡	0.382†	0.356†	0.356†	0.653†	0.640†	0.616†	0.567	0.499
+ CNN + Para	0.471	0.424†	**0.401**‡	**0.378**‡	0.313	0.657†	0.652†	0.614†	0.562	0.497
+ Ω + Prompt	0.488	0.407†	0.378†	0.338†	**0.368**‡	**0.701**‡	**0.679**‡	**0.686**‡	**0.620**‡	**0.518**

Significance Tests. To perform rigorous analysis, we adopt the bootstrapping method (p-value < 0.05) [28] for pair-wise significance tests. In all tables, we use † on BARTSCORE if it significantly ($p < 0.05$) outperforms other unsupervised metrics *excluding* BARTSCORE variants. We use ‡ on BARTSCORE if it significantly outperforms all other unsupervised metrics *including* BARTSCORE variants.

4.3 Experimental Results

4.3.1 Machine Translation

Tab. 3 illustrates Kendall's Tau correlation of diverse metrics on different language pairs. We can observe that: (1) BARTSCORE enhanced by fine-tuning tasks (CNN+Para) can significantly outperform all other unsupervised methods on five language pairs and achieve comparable results on the other two. (2) The performance of BARTSCORE can be further improved by simply adding a prompt (i.e., such as) without any other overhead. Notably, on the language pair de-en, using the prompt results in a 0.033 improvement, which even significantly surpasses existing state-of-the-art supervised metrics BLEURT and COMET. This suggests a promising future direction for metric design: *searching for proper prompts to better leverage knowledge stored in pre-trained language models instead of training on human judgment data* [31].

4.3.2 Text Summarization

Tab. 4 shows the meta-evaluation results of different metrics on the summarization task. We can observe that: (1) Simply vanilla BARTScore can outperform BERTScore and MoverScore by a large margin on 8 settings except the Info perspective on SummEval. Strikingly, it achieves improvements of 0.251 and 0.265 over BERTScore and MoverScore respectively. (2) The improvement on REALSum and SummEval datasets can be further improved when introducing fine-tuning tasks. However, fine-tuning does not improve on the NeR18 dataset, likely because this dataset only contains 7 systems with easily distinguishable quality, and vanilla BARTScore can already achieve a high level of correlation (> 0.6 on average). (3) Our prompt combination strategy can consistently improve the performance on *informativeness*, up to 0.072 Spearman correlation on the NeR18 dataset and 0.055 on SummEval. However, the performance from other perspectives such as *fluency* and *factuality* do not show consistent improvements, which we will elaborate on later (§4.4.2).

Analysis on Factuality Datasets The goal of these datasets is to judge whether a *short generated summary* is faithful to the original *long documents*. As shown in Tab. 5, we observe that (1) BARTScore + CNN can almost match *human baseline* on Rank19 and outperform all other metrics, including the most recent top-performing factuality metrics FactCC and QAGS by a large margin. (2) Using paraphrase as a fine-tuning task will reduce BARTScore's performance, which is reasonable since these two texts (i.e., the summary and document) shouldn't maintain the paraphrased relationship in general. (3) Introducing prompts does not bring an improvement, even resulting in a performance decrease.

Table 5: Results on Rank19 and QAGS datasets. where "Q" represents QAGS. Metrics achieve highest correlation are **bold**.

	Rank19	Q-CNN	Q-XSUM
	Acc.	Pearson	
ROUGE-1	0.568	0.338	-0.008
ROUGE-2	0.630	0.459	0.097
ROUGE-L	0.587	0.357	0.024
BERTScore	0.713	0.576	0.024
MoverScore	0.713	0.414	0.054
PRISM	0.780	0.479	0.025
FactCC [30]	0.700	–	–
QAGS [67]	0.721	0.545	0.175
Human [14]	0.839	–	–
BARTScore	0.684	0.661†	0.009
+ CNN	**0.836‡**	**0.735‡**	**0.184‡**
+ CNN + Para	0.788	0.680†	0.074
+ CNN + Prompt	0.796	0.719†	0.094

4.3.3 Data-to-text

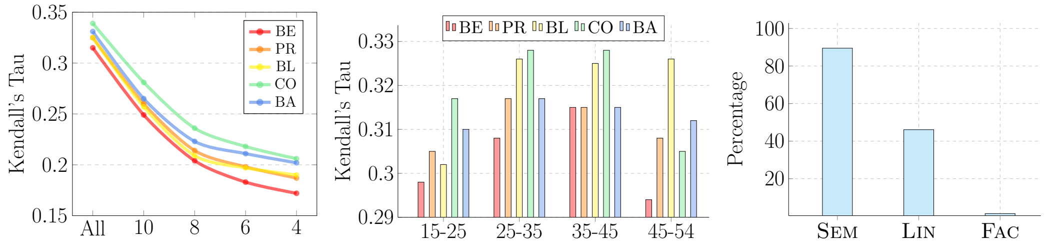
The experiment results on data-to-text datasets are shown in Tab. 6. We observe that (1) fine-tuning on the CNNDM dataset can consistently boost the correlation, for example, up to 0.056 gain on BAGEL. (2) Additionally, further fine-tuning on paraphrase datasets results in even higher performance compared to the version without any fine-tuning, up to 0.083 Spearman correlation on BAGEL dataset. These results surpass all existing top-performing metrics. (3) Our proposed prompt combination strategy can consistently improve correlation, on average 0.028 Spearman correlation. This is consistent with the findings in §4.3.2 that we can improve the aspect of *informativeness* through proper prompting.

Table 6: Results on data-to-text datasets. We report Spearman correlation. Metrics achieve highest correlation are **bold**.

	BAGEL	SFRES	SFHOT	Avg.
ROUGE-1	0.234	0.115	0.118	0.156
ROUGE-2	0.199	0.116	0.088	0.134
ROUGE-L	0.189	0.103	0.110	0.134
BERTScore	0.289	0.156	0.135	0.193
MoverScore	0.284	0.153	0.172	0.203
PRISM	0.305	0.155	0.196	0.219
BARTScore	0.247	0.164†	0.158	0.190
+ CNN	0.303	0.191†	0.190	0.228
+ CNN + Para	0.330†	0.185†	0.211†	0.242
+ Ω + Prompt	**0.336‡**	**0.238‡**	**0.235‡**	**0.270**

4.4 Analysis

We design experiments to better understand the mechanism by which BARTScore obtains these promising results, specifically asking three questions: Q1: Compared to other unsupervised metrics, where does BARTScore outperform them? Q2: How does adding prompts benefit evaluation? Q3: Will BARTScore introduce biases in unpredictable ways?



(a) Top-k systems (b) Reference length (c) Evaluation perspective

Figure 2: Fine-grained analysis (a,b) and prompt analysis (c). In (a, b), BE, PR, BL, CO, BA represent BERTScore, PRISM, BLEURT, COMET and BARTSCORE respectively. In (c), SEM, LIN, FAC denote *semantic overlap*, *linguistic quality* and *factual correctness* respectively.

4.4.1 Fine-grained Analysis

To answer Q1, we choose the MT task and break down the performance of each metric into different buckets based on different axes.

Top-k Systems We report the average correlation across all language pairs achieved by each metric given only translations from top-k systems. We vary the number of k, and the results are shown in Fig. 2-(a). We can see that BARTSCORE can outperform all other metrics (including one supervised metric BLEURT) except the existing state-of-the-art supervised metric COMET for different k, and the decrease in correlation becomes smoother than others when considering top-scoring systems. This indicates that BARTSCORE is robust to high-quality generated texts.

Reference Length We break down each test set into four buckets based on the reference length, which are $[15, 25)$, $[25, 35)$, $[35, 45)$, $[45, 54]$ and compute the Kendall's Tau average correlation of different metrics across all language pairs within each bucket.[6] The results are shown in Fig. 2-(b). We observe that BARTSCORE can outperform or tie with other unsupervised metrics over different reference lengths. Also, its correlation with human judgments is more stable compared to all other metrics. This indicates its robustness to different input lengths. More other analyses can be found in Appendix.

4.4.2 Prompt Analysis

For Q2, we choose the summarization and data-to-text tasks for analysis where we used all prompts from our prompt set. We first group all the evaluation perspectives into three categories: (1) *semantic overlap* (informativeness, pyramid score, and relevance) (2) *linguistic quality* (fluency, coherence) (3) *factual correctness* (factuality). We then calculate the percentage of prompts that result in performance improvements for each perspective within a dataset. Finally, we compute the average percentage of prompts that can lead to performance gains for each category. The results are shown in Tab. 2-(c). We can see that for *semantic overlap*, almost all prompts can lead to the performance increase, while for *factuality* only a few prompts can improve the performance. This also explains the results in §4.3.2 where we found that *combining the results of different prompts can lead to consistent increases in semantic overlap but worse performance in factuality*. Regarding *linguistic quality*, the effect of adding a prompt is not that predictive, which is also consistent with our findings in §4.3.2.

4.4.3 Bias Analysis

To answer Q3, we conduct bias analysis. Bias would indicate that the scores are too high or too low compared to the scores they are given by human annotators. Therefore, to see whether such biases exist, we inspected the rank differences given by human annotators and BARTScore (fine-tuned on CNNDM dataset) on the REALSumm dataset where 24 systems are considered, including both abstractive models and extractive models as well as models based on pre-trained models and models

[6] In each bucket, we remove the language pairs that do not contain over 500 samples. This results in the removal of kk-en in $[35, 45)$ and the removal of gu-en, kk-en, lt-en in $[45, 54]$.

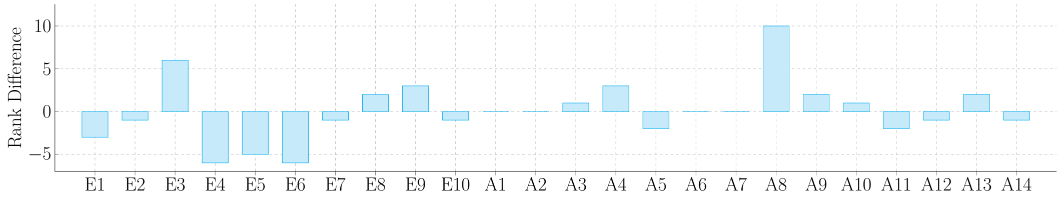


Figure 3: Bias analysis of BARTScore. The “Rank Difference" is the rank obtained using human judgements minus the rank got from BARTScore. Systems beginning with letter “E" are extractive systems while systems beginning with letter “A" are abstractive systems.

that are trained from scratch. We list all the systems below. And the resulting rank difference is shown in Fig. 3.

Extractive Systems E1: BanditSum [11]; E2: Refresh [49]; E3: NeuSum [81]; E4: LSTM-PN-RL [80]; E5: BERT-TF-SL [80]; E6: BERT-TF-PN [80]; E7: BERT-LSTM-PN-RL [80]; E8: BERT-LSTM-PN [80]; E9: HeterGraph [69]; E10: MatchSum [79].

Abstractive Systems A1: Ptr-Gen [62]; A2: Bottom-up [17]; A3: Fast-Abs-RL [5]; A4: Two-stage-RL [74]; A5: BERT-Ext-Abs [41]; A6: BERT-Abs [41]; A7: Trans-Abs [41]; A8: UniLM-1 [10]; A9: UniLM-2 [2]; A10: T5-base [56]; A11: T5-large [56]; A12: T5-11B [56]; A13: BART [33]; A14: SemSim [42].

As shown in Fig. 3, BARTScore is less effective at distinguishing the quality of extractive summarization systems while much better at distinguishing the quality of abstractive summarization systems. However, given that there is a trend for using abstractive systems as more and more pre-trained sequence-to-sequence models being proposed, BARTScore's weaknesses on extractive systems will be mitigated.

5 Implications and Future Directions

In this paper, we proposed a metric BARTScore that formulates evaluation of generated text as a text generation task, and empirically demonstrated its efficacy. Without the supervision of human judgments, BARTScore can effectively evaluate texts from 7 perspectives and achieve the best performance on 16 of 22 settings against existing top-scoring metrics. We highlight potential future directions based on what we have learned.

Prompt-augmented metrics As an easy-to-use but powerful method, *prompting* [39] has achieved impressive performance particularly on semantic overlap-based evaluation perspectives. However, its effectiveness in factuality and linguistic quality-based perspectives has not been fully demonstrated in this paper. In the future, more works can explore how to make better use of prompts for these and other evaluation scenarios.

Co-evolving evaluation metrics and systems BARTScore builds the connection between metric design and system design, which allows them to share their technological advances, thereby progressing together. For example, a better BART-based summarization *system* may be directly used as a more reliable automated *metric* for evaluating summaries, and this work makes them connected.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. The authors also thank Wei Zhao for assisting with reproducing baseline results. This work was supported by the Air Force Research Laboratory under agreement number FA8750-19-2-0200. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

References

[1] Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. Learning to retrieve reasoning paths over wikipedia graph for question answering. *arXiv preprint arXiv:1911.10470*, 2019.

[2] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652. PMLR, 2020.

[3] Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, and Pengfei Liu. Metrics also disagree in the low scoring range: Revisiting summarization evaluation metrics. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5702–5711, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[4] Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online, November 2020. Association for Computational Linguistics.

[5] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[6] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[7] Hoa Trang Dang. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12, 2005.

[8] Michael Denkowski and Alon Lavie. Extending the METEOR machine translation evaluation metric to the phrase level. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 250–253, Los Angeles, California, June 2010. Association for Computational Linguistics.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

[10] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *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 13042–13054, 2019.

[11] Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[12] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. GSum: A general framework for guided neural abstractive summarization. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Mexico City, June 2021.

[13] A. R. Fabbri, Wojciech Kryscinski, Bryan McCann, R. Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Trans. Assoc. Comput. Linguistics*, 9:391–409, 2021.

[14] Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, 2019.

[15] David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[16] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, 2018.

[17] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[18] Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. 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 708–719, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[19] Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 708–719, 2018.

[20] K. Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *NIPS*, 2015.

[21] Tsutomu Hirao, Manabu Okumura, Norihito Yasuda, and Hideki Isozaki. Supervised automatic evaluation for summarization with voted regression model. *Information Processing & Management*, 43(6):1521–1535, 2007.

[22] J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 44–54, Hong Kong, China, November 2019. Association for Computational Linguistics.

[23] Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. Scirex: A challenge dataset for document-level information extraction. *arXiv preprint arXiv:2005.00512*, 2020.

[24] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[25] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[26] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[27] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[28] Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[29] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[30] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online, November 2020. Association for Computational Linguistics.

[31] Teven Le Scao and Alexander Rush. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online, June 2021. Association for Computational Linguistics.

[32] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

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

[34] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *ArXiv*, abs/2101.00190, 2021.

[35] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[36] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157, 2003.

[37] Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, 2020.

[38] Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaicheng Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. Explainaboard: An explainable leaderboard for nlp. *arXiv preprint arXiv:2104.06387*, 2021.

[39] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021.

[40] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.

[41] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China, November 2019. Association for Computational Linguistics.

[42] Yixin Liu and Pengfei Liu. Simcls: A simple framework for contrastive learning of abstractive summarization. *arXiv preprint arXiv:2106.01890*, 2021.

[43] Qingsong Ma, Ondřej Bojar, and Yvette Graham. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels, October 2018. Association for Computational Linguistics.

[44] Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy, August 2019. Association for Computational Linguistics.

[45] François Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1552–1561, 2010.

[46] Chaitanya Malaviya, Graham Neubig, and Patrick Littell. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[47] Nitika Mathur, Timothy Baldwin, and Trevor Cohn. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online, July 2020. Association for Computational Linguistics.

[48] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, 2018.

[49] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. 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 1747–1759, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[50] Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[51] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[52] Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[53] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[54] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[55] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

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

[57] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics.

[58] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2021.

[59] Timo Schick and Hinrich Schütze. Exploiting cloze questions for few-shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020.

[60] Timo Schick and Hinrich Schütze. Few-shot text generation with pattern-exploiting training. *arXiv preprint arXiv:2012.11926*, 2020.

[61] Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*, 2020.

[62] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[63] Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, July 2020. Association for Computational Linguistics.

[64] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

[65] Miloš Stanojević and Khalil Sima'an. BEER: BEtter evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[66] Brian Thompson and Matt Post. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online, November 2020. Association for Computational Linguistics.

[67] Alex Wang, Kyunghyun Cho, and M. Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In *ACL*, 2020.

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

[69] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online, July 2020. Association for Computational Linguistics.

[70] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

[71] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[72] Weizhe Yuan, Pengfei Liu, and Graham Neubig. Can we automate scientific reviewing? *arXiv preprint arXiv:2102.00176*, 2021.

[73] Jerrold H Zar. Spearman rank correlation. *Encyclopedia of Biostatistics*, 7, 2005.

[74] Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 789–797, Hong Kong, China, November 2019. Association for Computational Linguistics.

[75] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[76] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.

[77] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China, November 2019. Association for Computational Linguistics.

[78] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. *arXiv preprint arXiv:2004.08795*, 2020.

[79] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online, July 2020. Association for Computational Linguistics.

[80] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of*

the 57th Annual Meeting of the Association for Computational Linguistics, pages 1049–1058, Florence, Italy, July 2019. Association for Computational Linguistics.

[81] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia, July 2018. Association for Computational Linguistics.

A Appendix

A.1 Summary of Commonly Used Metrics for Text Generation

Table 7: Summary of commonly used metrics for text generation. (S, H) represents whether a metric has a setting that uses source text and hypothesis text. (R, H) denotes whether a metric has a setting that uses reference text and hypothesis text. (S, R, H) indicates whether a metric has a setting that uses source text, hypothesis text and reference text. We use the following abbreviations for different tasks: SUM - Summarization, MT - Machine Translation, MUL - Multiple tasks, FAC - Factuality. For settings and tasks, we only list the ones justified by the original paper for each metric.

Metrics	Supervised	Paradigm	(S, H)	(R, H)	(S, R, H)	Task	Support FAC
ROUGE	✗	Match		✓		SUM	✗
BLEU	✗	Match		✓		MT	✗
CHRF	✗	Match		✓		MT	✗
BERTScore	✗	Match		✓		MUL	✗
MoverScore	✗	Match		✓		MUL	✗
PRISM	✗	Paraphrase	✓	✓		MT	✗
BLEURT	✓	Regress		✓		MT	✗
S3	✓	Regress			✓	SUM	✗
VRM	✓	Regress		✓		SUM	✗
COMET	✓	Regress, Rank			✓	MT	✗
BEER	✓	Rank		✓		MT	✗
BARTScore	✗	Generation	✓	✓		MUL	✓

A.2 Pre-trained Model Selection

Besides BART, we also tried T5 and PEGASUS as our sequence-to-sequence model to get generation scores. We conduct experiments on WMT19, and the results are shown in Tab. 8. We don't observe improvements in using PEGASUS or T5 over BART.

Table 8: Experiment results for PEGASUS and T5 on the WMT19 dataset. The highest correlations are bold.

	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
PEGASUS-large	0.124	0.297	0.237	0.205	0.252	0.148	0.311
PEGASUS-large-cnn	0.174	0.361	0.297	0.337	0.373	0.215	0.415
T5-base	0.170	0.357	**0.300**	0.339	0.348	**0.208**	0.378
T5-large	0.168	0.353	0.287	0.332	0.335	0.193	0.383
T5-base-cnn	0.177	0.364	0.295	0.342	0.347	0.207	0.402
BART	0.156	0.335	0.273	0.324	0.322	0.167	0.389
BART-cnn	**0.190**	**0.365**	**0.300**	**0.348**	**0.384**	**0.208**	**0.425**

A.3 Prompt Set

In Tab. 9, we list the full prompt set for both $\boldsymbol{s} \rightarrow \boldsymbol{h}$ direction and $\boldsymbol{h} \leftrightarrow \boldsymbol{r}$ direction.

A.4 Prompt Combination

Given a source sequence $\mathbf{x}$, a target sequence $\mathbf{y}$ and a set of prompts $\mathbf{z}_1, \mathbf{z}_2, \cdots \mathbf{z}_n$. We denote the prompted target sequence as $[\mathbf{y} : \mathbf{z}_i]$ for any prompt $\mathbf{z}_i$. Under the sequence-to-sequence model

Table 9: Full prompt set for both $\boldsymbol{s} \to \boldsymbol{h}$ and $\boldsymbol{h} \leftrightarrow \boldsymbol{r}$

	Prompt Set				
$\boldsymbol{s} \to \boldsymbol{h}$	Last	Tersely	Succinctly	In summation	To put it succinctly
	After	In brief	All in all	To summarize	Bringing up the rear
	Behind	In short	In outline	In a nutshell	To come to the point
	Lastly	Concisely	In closing	In conclusion	In the final analysis
	In sum	In precis	In passing	In winding up	Without wasting words
	To end	In a word	To conclude	Last in order	At the end of the day
	Curtly	Compactly	Summarising	In a few words	Without waste of words
	Crisply	Summarily	In the rear	As a final point	Finally yet importantly
	At last	To sum up	Summarizing	Not least of all	To put it in a nutshell
	Pithily	Basically	Laconically	To put it briefly	When all is said and done
	Shortly	In the end	At the rear	Not to mince words	To cut a long story short
	In fine	At the end	To be brief	Last but not least	Not to beat about the bush
	Finally	In essence	Last of all	Just as importantly	In drawing things to a close
	Briefly	Ultimately	Elliptically	To put it concisely	Not to put too fine a point on it
$\boldsymbol{h} \leftrightarrow \boldsymbol{r}$	As	To wit	As it were	Case in point	As an illustration
	sc.	That is	Especially	That is to say	To give an example
	i.e.	Such as	For example	To rephrase it	To give an instance
	Like	Scilicet	Particularly	To be specific	To put it another way
	Viz.	Videlicet	Specifically	In plain English	By way of explanation
	Namely	Expressly	For instance	Take for example	By way of illustration
	id est	Specially	To illustrate	Strictly speaking	

parameterized by θ, we combine the generation scores using different prompts as follows:

$$\text{BARTS}\textsc{core}\text{-P}\textsc{rompt} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{t=1}^{m_i}\log p([\mathbf{y}:\mathbf{z}_i]_t|[\mathbf{y}:\mathbf{z}_i]_{<t},\mathbf{x},\theta) \tag{3}$$

Where n is the number of prompts considered, m_i is the target length after adding the i-th prompt.

A.5 Robustness to Language Pair Distance

Translations between different language pairs contain different variances. Here we aim to measure how the performance of a metric will change considering the distance between a language pair. We use language vectors to measure the distance between two languages [46], and consider 6 distances, which are *syntactic*, *geographic*, *phonological*, *genetic*, *inventory* and *featural* distances. We plot the Pearson correlation heatmap in Fig. 4. We observe that the correlation doesn't change much w.r.t. different distances across metrics. And the results show that all metrics have a significant correlation with *genetic* distance. This indicates that metrics are good at measuring translation quality from genetically different languages. This may be because the translation from similar languages is easier than dissimilar languages, making translation systems less distinguishable.

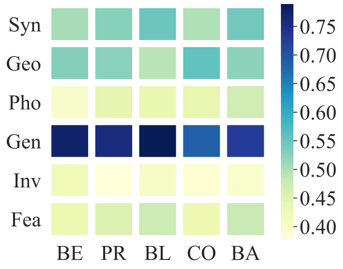


Figure 4: Pearson correlation between language pair distance and correlation with human metrics.