Investigating Metric Diversity for Evaluating Long Document Summarisation

Cai Yang[*]
Australian National University
Canberra, Australia
cai.yang@anu.edu.au

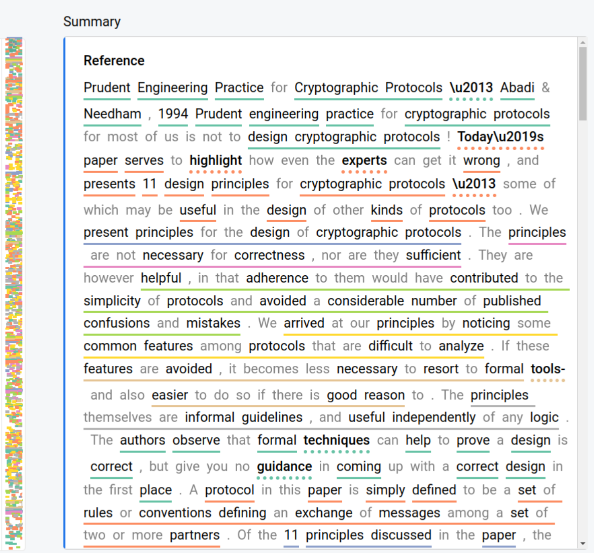
Stephen Wan
CSIRO Data61
Sydney, Australia
stephen.wan@data61.csiro.au

Abstract

Long document summarisation, a challenging summarisation scenario, is the focus of the recently proposed LongSumm shared task. One of the limitations of this shared task has been its use of a single family of metrics for evaluation (the ROUGE metrics). In contrast, other fields, like text generation, employ multiple metrics. We replicated the LongSumm evaluation using multiple test set samples (vs. the single test set of the official shared task) and investigated how different metrics might complement each other in this evaluation framework. We show that under this more rigorous evaluation, (1) some of the key learnings from Longsumm 2020 and 2021 still hold, but the relative ranking of systems changes, and (2) the use of additional metrics reveals additional high-quality summaries missed by ROUGE, and (3) we show that SPICE is a candidate metric for summarisation evaluation for LongSumm[1].

1 Introduction

Text summarisation is an increasingly sought-after capability that is required by corporations and governments for productivity gains. For such use-cases, long documents with complex structures are often used as the input data. However, work on summarizing long documents into detailed summaries has not dominated the summarisation research field. There have been some exceptions to this, for example, work on government reports (Huang et al., 2021; Cao and Wang, 2022) and PubMed literature (Gupta et al., 2021). In contrast, most of the text summarisation work focuses on shorter documents or generating shorter summaries (for example, Wikipedia data (Gholipour Ghalandari et al., 2020), scientific articles (Teufel and Moens, 2002; Cohan et al., 2018), and news summarisation (See et al., 2017))).



Summary

Reference

Prudent Engineering Practice for Cryptographic Protocols \u2013 Abadi & Needham , 1994 Prudent engineering practice for cryptographic protocols for most of us is not to design cryptographic protocols ! Today\u2019s paper serves to highlight how even the experts can get it wrong , and presents 11 design principles for cryptographic protocols \u2013 some of which may be useful in the design of other kinds of protocols too . We present principles for the design of cryptographic protocols . The principles are not necessary for correctness , nor are they sufficient . They are however helpful , in that adherence to them would have contributed to the simplicity of protocols and avoided a considerable number of published confusions and mistakes . We arrived at our principles by noticing some common features among protocols that are difficult to analyze . If these features are avoided , it becomes less necessary to resort to formal tools- and also easier to do so if there is good reason to . The principles themselves are informal guidelines , and useful independently of any logic . The authors observe that formal techniques can help to prove a design is correct , but give you no guidance in coming up with a correct design in the first place . A protocol in this paper is simply defined to be a set of rules or conventions defining an exchange of messages among a set of two or more partners . Of the 11 principles discussed in the paper , the

Figure 1: An example of a long document abstractive summary from the LongSumm data set, presented using SUMMVis (Vig et al., 2021).

To bridge this gap, the shared task of summarizing long scientific articles (LongSumm) was proposed, where the system should produce a detailed and informative technical summary of a source article. This shared task was introduced in the 2020 Scholarly Document Processing workshop (Chandrasekaran et al., 2020). The shared task includes an extractive and abstractive version of the problem. The former is based on the TalkSumm dataset (Lev et al., 2020), an alignment of presentation transcripts to the publication. The latter is captured using a data set of technical blogs and publications (Chandrasekaran et al., 2020).

The abstractive data set is interesting in that summaries must provide both high-level and low-level details. An example is provided in Figure 1, where the summary is a blog "walkthrough" of the main points of a paper (presented using the SUMMVis tool (Vig et al., 2021), showing colored alignments of content to the source material).

The 2020/2021 LongSumm shared tasks resulted

[*] Work done during the internship at CSIRO Data61.

[1] Our code is available at https://github.com/caiyangcy/SDP-LongSumm-Metric-Diversity

Proceedings of the Third Workshop on Scholarly Document Processing, pages 115–125
October 12–17, 2022.

in a couple of key learnings for abstractive summarisation: (1) that there was no clear difference in performance between extractive and abstractive methods; and (2) approaches that focus on the representation of long documents, such as the Bigbird (Zaheer et al., 2020) and Pegasus (Zhang et al., 2020a) combination outperformed simpler abstractive methods like BART (Lewis et al., 2020).

One potential weakness of the LongSumm shared tasks is that they were limited to the ROUGE family of metrics (Lin, 2004), including recall of unigrams (ROUGE-1), bigrams (ROUGE-2), and longest common subsequences (ROUGE-LCS). In contrast, current trends in Natural Language Generation (NLG), for example, the E2E evaluation (Dušek et al., 2020), and Image Caption Generation (ICG), for example, the MS Coco evaluation (Chen et al., 2015), employ multiple metrics.

There are also issues with the application of ROUGE to new data sets. For example, ROUGE has been shown to be problematic when used on text types other than news, like microblogs (Mackie et al., 2014), meeting summaries, (Liu and Liu, 2008) and online review text (Tay et al., 2019).[2]

Given that it is not clear that ROUGE is necessarily the best metric for this new domain, we take the approach that diversity of metrics is key. We thus employ the metrics from NLG E2E shared task and MS Coco evaluation scripts. We also add some of the new metrics from these fields, such as SPICE (Anderson et al., 2016), a metric considering semantic graphs that has been demonstrated to improve image captioning evaluation, and BERTScore (Zhang et al., 2020b), a metric that utilizes BERT contextual embeddings to better capture lexical and structural semantics and which is increasingly used in evaluating text summarisation.[3] These metrics can be seen as covering a range of linguistic phenomena. We provide more detail on the metrics in Section 4.

To consider the role of the different metrics for the LongSumm evaluation, we use a spectrum of different system approaches, including oracle methods, baselines, and state-of-the-art approaches. In addition, where the original LongSumm evaluation uses a single test set, we repeat our experiments multiple times with different training-testing data set splits to account for variance.

Our contributions are as follows. (I) We retest key outcomes from the earlier shared tasks, e.g, (i) abstractive and extractive methods perform similarly on the LongSumm abstractive data set, and (ii) the relative performance of tested algorithms. (II) We show that the informativeness of ROUGE might be affected by stopword matching. (III) We show that SPICE agrees somewhat with ROUGE and BERTScore, offering a complementary view on summarisation quality.

The remainder of the paper is structured as follows. In Section 2, we outline related work. Section 3 describes the different summarisation methods and baseline approaches. We outline our experimental procedure in Section 4. In Section 5, we describe our experimental results that address the research questions above. Section 6 presents qualitative analysis and future work. We present concluding remarks in Section 7.

2 Related Work

In this section, we outline some of the highlights in which the NLP community has critically examined evaluation methodology. We provide more details on shared task data, leading approaches, and metrics examined in subsequent sections.

We note that the field of machine translation has been a source of inspiration for other NLP fields. Indeed, the ROUGE metric is itself inspired by the BLEU metric from translation research. This field has shown that reliance on intrinsic metrics and reference summaries is problematic. For example, the BLEU metric may not correlate with human judgments (Callison-Burch et al., 2006). Indeed, in recent years, machine translation has turned to the research topic of Quality Estimation (QE) (Specia and Astudillo, 2018), the task of estimating run-time translation quality without ground truth data. Our work has some superficial similarities to QE methodology, in examining summary rankings and high and low-quality quartiles. However, our analysis differs from the core focus of QE, as we investigate the utility of multiple metrics.

Within the NLG community, BLEU has been used as an evaluation metric even though it is problematic. For example, it has been shown not to correlate with human judgments (for example, (Belz and Reiter, 2006) and (Cahill, 2009)). The use of these metrics is further called into doubt when we

[2] Note: the ROUGE metric was originally designed for the DUC 2001 data set of news articles at a time when extractive summarisation methods were the dominant method. For more information about DUC 2001, visit https://duc.nist.gov/pubs.html#2001

[3] Indeed, BERTScore is an official metric of the LongSumm 2022 shared task.

see that n-gram matching metrics like BLEU are also not suitable for evaluating text simplification (Sulem et al., 2016), a closely related task to text summarisation. This has led to the research in new metrics (for example, GLEU (Mutton et al., 2007) and BLEURT (Das and Parikh, 2019)). In this work, we follow the NLG and ICG best practice, which is to use a combination of metrics, knowing that each individual metric may have its failings.

There have been some recent works on evaluating summarisation metrics (Bhandari et al., 2020; Fabbri et al., 2021), which highlights the limitation of current metrics and the need for upgrading evaluation protocols. We note that other metrics exist to overcome some of the limitations of ROUGE (Schluter, 2017), such as needing to account for multiple judgments of content saliency as in the Pyramid method (Nenkova and Passonneau). A linear ensemble of diverse metrics has also been shown to be able to outperform metrics in isolation (Kasai et al., 2022). The NLG community has tended to report human quality assessments, for example, collecting judgments for *quality* and *naturalness* (Novikova and Rieser, 2018). In this respect, our work is again complementary in that we use SUMMVis (Vig et al., 2021) to inspect the quality of the system summaries.

3 Baselines and Approaches

3.1 Oracles

To estimate an upper bound on performance for the metrics, we employ a series of "oracle" methods, so-called because they use the reference summaries to approximate a perfect content selection mechanism. The oracle methods are:

(Or-TopK) Oracle-Top K Sentences Matching For each sentence from the reference summary, we extract the k most similar sentence from the document. Similarity is measured through the longest contiguous matching subsequence by using `SequenceMatcher` from `difflib`.

(Or-TopK-SS) Oracle-Surrounding Sentences The process is similar to Oracle-Single Sentence Matching, except the preceding and subsequent sentence of the most similar sentence will also be selected.

(Or-TopK-PM) Oracle-Paragraph Matching Instead of finding the most similar sentence, paragraphs are chosen and included in the summary. We do this by selecting the paragraph to which the most similar sentence belongs.

(Or-SW) Oracle-only Stopwords This entry only includes stopwords in the summaries. We do this by selecting stopwords from the reference summaries and including them in the summary.

3.2 Baseline Text Summarizers

The baseline summarisation methods are:

(RandN) Randomly select n sentences and include them in the summary.

(LeadN) Select the first n sentences. This is known to be a strong baseline for other data sets.

3.3 2020/2021 Best Published Methods

For this study, we take the extractive and abstractive entries from the 2020 (Chandrasekaran et al., 2020) and 2021 (Ying et al., 2021a) LongSumm shared tasks. For each method tested, we use the authors' public code repository and use system parameters as described in the original published works.

The published performance of these methods is presented in Table 1. The extractive methods ranking using ROUGE-LCS is: DGCNN $>$ SummaRuNNer $>$ BERTSum-Multi. The abstractive methods ranking is: Bigbird-Pegasus $>$ BART.

3.3.1 DGCNN

Dilated Gated Convolutional Neural Networks (DGCNN) have been used for extractive summarisation (Ying et al., 2021b). It is based on Conv1D layers with residual connections and different dilation rates. The sentences from each document are passed through RoBERTa and the output from the last hidden layers with average pooling is used as the feature representations. These are passed into the DGCNN layers to output a binary label for sentence selection.[4]

3.3.2 SummaRuNNer

SummaRuNNer (Nallapati et al., 2017) is an extractive model consisting of a two-layer bi-directional GRU. The first layer operates on the word level to produce hidden state representations of words while the second layer operates on the sentence level to encode sentence representations. A document representation is obtained through a nonlinear transformation of the sentence representations. Selection (binary) classification is made on

[4] https://aclanthology.org/2021.sdp-1.12

	Recall			F-measure		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
BERTSum-Multi (Sotudeh Gharebagh et al., 2020)	0.5460	0.1728	0.2090	0.5311	0.1677	0.2034
SummaRuNNer (Ghosh Roy et al., 2020)	0.4390	0.1498	0.1898	0.4938	0.1686	0.2138
DGCNN (Ying et al., 2021a)	0.5275	0.1711	0.2209	0.2262	0.1747	0.5415
Bigbird-Pegasus (Ying et al., 2021a)	0.5080	0.1740	0.2156	0.1634	0.4755	0.2016
BART (Ying et al., 2021a)	0.1921	0.0533	0.1062	0.1122	0.0310	0.0620

Table 1: Top-performing entries reported by SDP-2020 and SDP-2021 and their reported performance.

sentences, which considers the content, document context, salience and novelty. Ghosh Roy et al. (2020) apply this method in LongSumm.[5] [6]

3.3.3 BERTSum-Multi

BERTSum-Multi (Sotudeh Gharebagh et al., 2020; Sotudeh et al., 2021) is a variant of extractive summarisation approach BERTSum (Liu and Lapata, 2019). The variant, proposed for the LongSumm shared task, uses joint task training to select sentences and predict section labels for each sentence. It outperforms the standard BERTSum algorithm for LongSumm data (Sotudeh Gharebagh et al., 2020; Sotudeh et al., 2021).[7]

3.3.4 Bigbird-Pegasus

The Bigbird-Pegasus approach (Ying et al., 2021a) is an abstractive model proposed for the LongSumm shared task. It incorporates Bigbird (Zaheer et al., 2020), a sparse attention mechanism that overcomes the quadratic complexity in the encoder, which is designed to capture more context at the document level. This document representation is then used with Pegasus, an abstractive summarisation approach that is pretrained through gap sentences generation and masked language modeling (Zhang et al., 2020a).[8][9]

3.3.5 BART

BART (Lewis et al., 2020) is an abstractive model whose pretrained objective is to denoise the input text, which is corrupted by token deletion, token masking, sentence permutation, text infilling and document rotation. It was proposed for use in LongSumm by (Ying et al., 2021a).[10][11]

4 Experimental Procedure

4.1 Data

In this work, we use the abstractive subset of the LongSumm data set for evaluation purposes. As the public release of this data set does not have a specified test set, we are required to create our own training, development, and testing partitions.

4.2 Evaluation conditions

We randomly sample 22 test cases from the public data set as held out data, repeating this procedure 10 times, ensuring disjoint training and testing sets. Summaries are limited to 600 words for evaluation, following the LongSumm shared task.

4.3 Evaluation Metrics

In this work, we use a diverse set of evaluation metrics, following best practices from the NLG and ICG communities. Unless otherwise specified, we use the implementation from the E2E shared task.[12]

Our categories of metrics are (with the dominant metrics used in that community in bold):

- Translation: **BLEU**, NIST, METEOR
- Summarisation: **ROUGE** family of metrics
- Image Captioning: CIDEr, **SPICE**
- Semantic: **BERTScore**, METEOR, SPICE

4.3.1 BLEU

BLEU (Papineni et al., 2002) was originally proposed for machine translation. It is based on the product of modified n-gram precision and brevity penalty that penalizes short sentences. BLEU weights each n-gram equally.

4.3.2 NIST

Adapted from BLEU, NIST (Doddington, 2002) pays more attention to less frequent n-grams. It uses the arithmetic mean as opposed to the geometric mean in BLEU for the modified n-gram

[5] https://github.com/sayarghoshroy/Summaformers

[6] model: https://github.com/hpzhao/SummaRuNNer

[7] github.com/Georgetown-IR-Lab/ExtendedSumm

[8] aclanthology.org/2021.sdp-1.12

[9] Pretrained model: summarisation/arxiv. See console.cloud.google.com/storage/browser/bigbird-transformer/summarisation/arxiv/pegasus

[10] Pretrained model: *"facebook/bart-large"*.

[11] huggingface.co/docs/transformers/model_doc/bart

[12] github.com/tuetschek/e2e-metrics

precision and weights each n-gram by its frequency in the references.

4.3.3 ROUGE*

ROUGE family of metrics (Lin, 2004) is based on n-gram overlap between system-generated summaries and reference summaries. Following the SDP workshops, we use ROUGE-1 , ROUGE-2 and ROUGE-LCS as our evaluation metrics.

4.3.4 CIDEr

CIDEr (Vedantam et al., 2015) was first proposed for image captioning tasks to capture consensus. CIDEr computes the cosine similarity using Term Frequency Inverse Document Frequency (TF-IDF) vectors for each n-gram. We use a variant of CIDEr with Gaussian penalty (named CIDEr-D) introduced to reduce the effects of word repetition.

4.3.5 METEOR

METEOR (Banerjee and Lavie, 2005) aligns the system output and references based on exact word matching and morphological variations such as stems, synonyms, and paraphrases of words. METEOR is calculated as the harmonic mean of precision and recall, along with a penalty factor to favour longer matching sequences.

4.3.6 SPICE

Metrics mentioned above are sensitive to n-gram overlap. However, n-gram overlap is neither necessary nor sufficient for two sentences to convey the same meaning (Giménez and i Villodre, 2007). SPICE is based on the hypothesis that semantic propositional content is an important component of image caption human evaluation (Anderson et al., 2016). SPICE constructs scene graphs based on input text processed via semantic parsing. It computes precision, recall and F1 score based on the binary matching of logical tuples, which contains objects, attributes and relations from the scene graphs.

Although SPICE is designed to operate on a system generated and reference caption, we adapt it to the summarisation scenario, and use a full system generated and reference summaries as input.[13] While the captioning scenario corresponds to a comparison of two sentences, our usage is a comparison of sets of sentences. We show that even this simple adaptation shows agreement with ROUGE and BERTScore metrics.

4.3.7 BERTScore

N-gram models can under-estimate performance on semantically-correct matched phrases (Zhang et al., 2020b) and fail to penalize semantically-critical ordering changes (Isozaki et al., 2010). To overcome such issues, BERTScore (Zhang et al., 2020b) maps tokens to BERT contextual embeddings (Devlin et al., 2019) and computes precision, recall and F-measure through cosine similarity of word tokens, optionally weighted by the inverse document frequency to emphasize rare tokens.[14]

5 Results

5.1 Agreement of Metrics on Baselines

We begin by examining how the metrics score the oracle and baseline methods. These will provide some insights on upper bounds in performance (oracle methods), performance due to chance (random methods), and performance due to trivial generation (stopword baseline).

We present the baseline and oracle methods in Table 2. We see that the best oracle method is one that takes the best matching source document sentence (that is aligned with a reference sentence), and that adding additional context, whether by paragraph or surrounding sentences, does not improve performance (e.g., Or-TopK=1-PM does not improve on Or-TopK=1). Similarly, returning the top 3-5 aligned sentences does not help. This may be due to lexical divergences between the reference and system summaries, so matches are predominantly in the first sentence.

Interestingly, there is not a large difference in scores between random and lead methods; both increase as more sentences are selected. Note, BERTScore measures for baselines and oracle methods have a narrow range of 2-3 points.

We note that stopwords account for a large proportion of lexical correspondences in ROUGE, as evidenced by the high ROUGE-1 and ROUGE-2 scores for Or-SW, which are in the same range as the SOTA scores in Table 3. This suggests yet another weakness; namely, word recall may be overly dominated by non-content words like stopwords.

5.2 Agreement of Metrics on Systems

We present the results of system comparisons in Table 3. It is clear that the best systems outperform the baseline methods in every case. However, there

[13] github.com/tylin/coco-caption

[14] pypi.org/project/bert-score/

system	BLEU	NIST	ROUGE-1-F1	ROUGE-2-F1	ROUGE-LCS-F1	CIDEr	METEOR	SPICE	BERTScore
Or-TopK=1	**0.2739**(0.0432)	**6.2682**(0.5135)	**0.5439**(0.0233)	**0.2453**(0.0336)	0.3430(0.0299)	**0.2207**(0.1656)	**0.2456**(0.0141)	**0.3061**(0.0387)	**0.8521**(0.0057)
Or-TopK=3	0.1403(0.0216)	4.5371(0.1952)	0.4959(0.0193)	0.1828(0.0196)	0.2281(0.0132)	0.0324(0.0342)	0.1973(0.0090)	0.2225(0.0184)	0.8380(0.0044)
Or-TopK=1-PM	0.0937(0.0115)	3.6976(0.1293)	0.4256(0.0165)	0.1233(0.0110)	0.1802(0.0118)	0.0446(0.0398)	0.1732(0.0049)	0.1719(0.0117)	0.8218(0.0035)
Or-TopK=1-SS	0.1241(0.0150)	4.1964(0.1516)	0.4668(0.0176)	0.1553(0.0150)	0.1979(0.0126)	0.0346(0.0358)	0.1848(0.0071)	0.1969(0.0160)	0.8285(0.0036)
Or-SW	0.0134(0.0020)	0.0314(0.0084)	0.4959(0.0090)	0.1484(0.0058)	-	0.0001(0.0002)	0.0885(0.0032)	0.0063(0.0025)	0.7378(0.0039)
RandN=3	0.0001(0.0001)	0.0000(0.0000)	0.1360(0.0174)	0.0265(0.0069)	0.0809(0.0093)	0.0003(0.0008)	0.0252(0.0029)	0.0577(0.0077)	0.8067(0.0028)
RandN=5	0.0016(0.0009)	0.0002(0.0004)	0.1993(0.0182)	0.0405(0.0070)	0.1049(0.0080)	0.0008(0.0011)	0.0414(0.0040)	0.0822(0.0090)	0.8103(0.0032)
RandN=10	0.0158(0.0032)	0.1156(0.0876)	0.3054(0.0118)	0.0630(0.0067)	0.1348(0.0050)	0.0016(0.0037)	0.0776(0.0057)	0.1166(0.0057)	0.8144(0.0030)
LeadN=3	0.0001(0.0001)	0.0000(0.0000)	0.1695(0.0125)	0.0470(0.0046)	0.1019(0.0060)	0.0004(0.0010)	0.0303(0.0021)	0.0850(0.0059)	0.8236(0.0042)
LeadN=5	0.0018(0.0010)	0.0001(0.0002)	0.2424(0.0111)	0.0673(0.0075)	0.1315(0.0071)	**0.0081**(0.0140)	0.0495(0.0037)	0.1145(0.0087)	**0.8262**(0.0038)
LeadN=10	**0.0202**(0.0040)	**0.1399**(0.0928)	**0.3279**(0.0142)	**0.0837**(0.0088)	**0.1539**(0.0080)	0.0032(0.0054)	**0.0864**(0.0044)	**0.1321**(0.0080)	0.8204(0.0041)
Best Oracle	0.2739	6.2682	0.5439	0.2453	0.4857	0.2207	0.2456	0.3061	0.8521
Best Baseline	0.0202	0.1399	0.3279	0.0837	0.1539	0.0032	0.0864	0.1321	0.8204
δ(Oracle-Baseline)	0.2537	6.1283	0.2160	0.1616	0.3318	0.2175	0.1592	0.1740	0.0317

Table 2: Baselines and Non-trivial Measurement, where N=number of sentences in the ground truth summary. Each cell contains the average score across the 10 test sets (with standard deviation in brackets). Best values are in bold.

system	BLEU	NIST	ROUGE-1-F1	ROUGE-2-F1	ROUGE-LCS-F1	CIDEr	METEOR	SPICE	BERTScore
SummaRuNNer	**0.0840**(0.0130)	3.2979(0.2287)	*0.4205*(0.0236)	*0.1204*(0.0175)	*0.1772*(0.0161)	0.0119(0.0140)	0.1508(0.0066)	**0.1619**(0.0148)	*0.8230*(0.0051)
DGCNN	0.0783(0.0164)	3.3395(0.2509)	0.3975(0.0240)	0.1075(0.0151)	0.1613(0.0109)	0.0135(0.0180)	0.1606(0.0078)	0.1522(0.0139)	0.8145(0.0036)
BERTSum-Multi	0.0757(0.0089)	**3.4014**(0.2060)	0.4204(0.0200)	0.1050(0.0078)	0.1644(0.0089)	*0.0140*(0.0219)	**0.1819**(0.0067)	0.1570(0.0104)	0.8207(0.0031)
BART	*0.0642*(0.0078)	*2.3875*(0.5556)	**0.4248**(0.0249)	**0.1256**(0.0119)	**0.1845**(0.0109)	**0.0173**(0.0185)	*0.1406*(0.0064)	*0.1559*(0.0118)	**0.8304**(0.0046)
Bigbird-Pegasus	0.0285(0.0041)	2.0301(0.4101)	0.3438(0.0162)	0.0662(0.0055)	0.1551(0.0063)	0.0064(0.0095)	0.1161(0.0070)	0.1113(0.0092)	0.8023(0.0030)
Best Extractive	0.0840	3.4014	0.4205	0.1204	0.1772	0.0140	0.1819	0.1619	0.8230
Best Abstractive	0.0642	2.3875	0.4248	0.1256	0.1845	0.0173	0.1406	0.1559	0.8304
Ex vs Ab Winner	(ex)	(ex)	(ab)	(ab)	(ab)	(ab)	(ex)	(ex)	(ab)

Table 3: Extractive or Abstractive models. Each cell contains the average score across the 10 test sets (with standard deviation in brackets). Best values in bold, second best in italics.

is still a considerable margin between the oracle methods (an estimate of an upper bound) and the best system, suggesting that there is still plenty of room for improvement for the task of selecting the content for the generated summary.

As we use multiple test set samples, our results are not exactly the same as the published results displayed in Table 1, however the scores are roughly in the same neighbourhood as the published results. Using ROUGE-LCS F1, our ranking of extractive systems in this replication of LongSumm results is SummerRuNNer > BERTSum-Multi > DGCNN. Curiously, the best-placed extractive method is now ranked last based on ROUGE-LCS alone. For the abstractive systems, we note that Bigbird-Pegasus performed worse than BART, and that the BART ROUGE performance was very different from published results. We suspect the difference is in part due to our use of multiple test sets, which will account for variance in the test data.

Rankings by other metrics are different again. However, the three methods which were repeatedly ranked first were SummaRuNNer, BERTSum-Multi, and BART. The translation metrics ranked extractive approaches best. ROUGE metrics ranked the BART system first. CIDER and SPICE, favour different systems, BART and SummaRuNNer, respectively. For the semantic metrics, the METEOR and SPICE systems ranked extractive methods highest, and BERTScore ranked BART best. Note that only differences measured by BERTScore and METEOR were statistically significant.

We also find that there is no clear winner between the extractive and abstractive methods on this data set, when evaluating with the multiple metrics. If we group together all ROUGE metrics, extractive and abstractive methods are tied on 4 metrics apiece (last row, Table 3).

We thus conclude that our replication weakly agrees with prior published results. We observe, as in prior work, that extractive and abstractive methods perform similarly on the abstractive data set. However, the ranking of methods differs slightly.

5.3 Inspecting top and bottom ranks per metric

We explore the notion of the complementarity of the metrics by examining the top and bottom n ranked generated summaries, as ranked by each of the different metrics. Due to space constraints, we present and discuss a subset of the results here, limiting the discussion to the dominant community metrics (BLEU, ROUGE-LCS (hereafter ROUGE), SPICE, and BERTScore), and considering only output from the three systems that had some agreement across the metrics as performing well (SummaRuNNer, BERTSumm-Multi, and BART).

In Table 4, we present a summary of the sim-

Comparisons	BART	SummaRunner	BERTSum	Avg.
RL. vs BS.	0.62(0.19)/0.64(0.12)	0.70(0.13)/0.56(0.15)	0.72(0.13)/0.62(0.11)	0.67(0.15)/0.61(0.13)
RL. vs BL.	0.34(0.13)/0.46(0.22)	0.30(0.18)/0.44(0.15)	0.28(0.16)/0.40(0.15)	0.31(0.16)/0.43(0.17)
RL. vs SP.	0.60(0.15)/0.70(0.10)	0.64(0.15)/0.74(0.04)	0.56(0.15)/0.74(0.20)	0.60(0.15)/0.73(0.11)
BS. vs BL.	0.26(0.18)/0.36(0.22)	0.22(0.17)/0.38(0.14)	0.26(0.13)/0.46(0.20)	0.25(0.16)/0.40(0.18)
BS. vs SP.	0.56(0.15)/0.68(0.13)	0.68(0.13)/0.60(0.18)	0.52(0.10)/0.62(0.17)	0.59(0.13)/0.63(0.16)
BL. vs SP.	0.44(0.20)/0.44(0.20)	0.30(0.18)/0.44(0.15)	0.38(0.11)/0.60(0.18)	0.37(0.16)/0.49(0.18)
Avg.	0.47(0.17)/0.55(0.17)	0.47(0.16)/0.53(0.14)	0.45(0.13)/0.57(0.17)	0.46(0.15)/0.55(0.16)

Table 4: Agreement in the top and bottom quartiles of test cases, as ranked by the BLEU (BL), ROUGE-LCS (RL), SPICE (SP), and BERTScore (BS) metrics.

ilarities in rankings in a pairwise comparison of metrics, across different systems. Specifically, we examine the top and bottom quartiles of a test set of 22 documents (where we take the top and bottom 5 ranked documents).[15] Each cell in the table shows two numbers, one for the agreement of test case ids in the top quartile and the corresponding agreement of the bottom quartile.[16]

We note that the agreement of the bottom quartile is usually higher than the top quartile. This is because this quartile contains the difficult test cases to score automatically, which will tend to be the same for all metrics. The difficulty lies, for example, in the fact that the reference summaries are very short (leaving less opportunity to match the content that might well be reasonable).

Curiously, there are some summaries that are in the top quartile for some metrics which are in the bottom quartile for others. Occasionally, BLEU will place summaries judged to be in the top quartile by another metric into its bottom quartile. We assume this relates to critiques of using BLEU for NLG, where novel text differing from the reference will be penalized.

Most interesting is the diversity of summaries selected in the top quartile. When looking at the average agreement for each metric pair (last column of Table 4), we note that ROUGE and BERTScore have the best agreement of all pairs of metrics, which is constant across different summarisation systems. SPICE metric has the second-best agreement when paired with either ROUGE-LCS or BERTScore. The BLEU metric has the lowest agreement with the others. These results indicate that one should consider the use of SPICE as a summarisation metric.

6 Discussion

6.1 Qualitative Analysis of Metric Complementarity

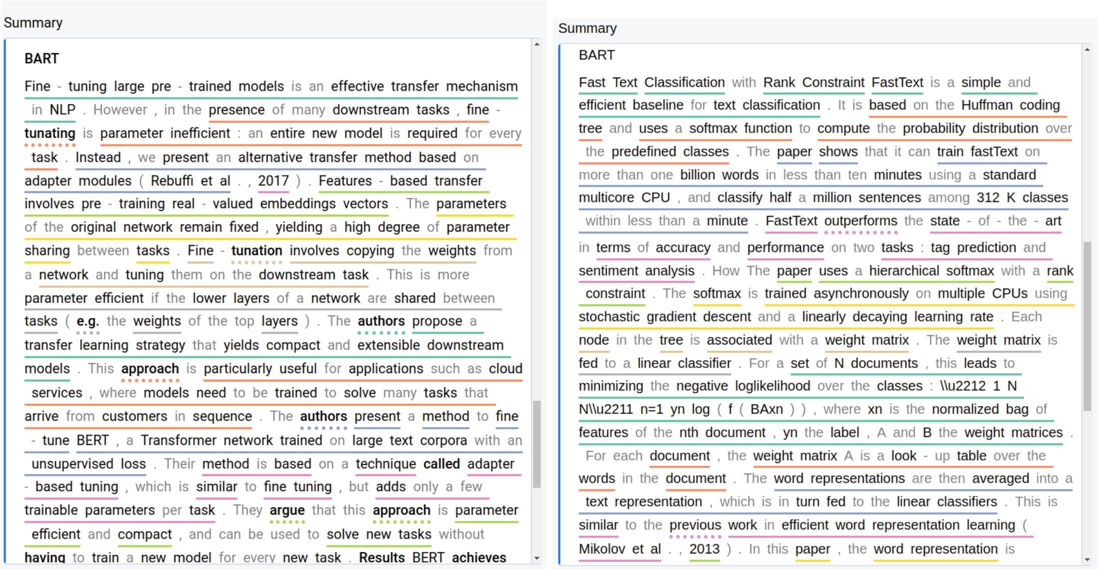
The results in Table 4 raise an interesting question. When utilizing a diverse set of metrics, what are the complementary qualities of a system summary that might be captured by the metrics? That is, do the summaries ranked highly by SPICE and BLEU represent quality summaries that are neglected by ROUGE and BERTScore? For this manual analysis, there are 3 test cases agreed upon by ROUGE and BERTscore, and 4 complementary test cases ranked highly by SPICE and BLEU. Upon inspection of these summaries manually, we find that all seven summaries are generally reasonable.

For insight, we examine the source-summary alignments generated SUMMVis for the 3 test cases that ROUGE and BERTScore agree upon, and the 4 complementary, presented in Figure 3. We note that the last 4, representing the complementary summaries, seem to share the property that content is selected later in the source document. That is 3 summaries ranked highly by ROUGE/BERTScore summaries seem "top-heavy" and the complementary set seem "bottom-heavy", with respect to where content from the source is drawn from.

We present an example of the ROUGE/BERTScore highly-ranked summary and an example from the complementary set in Figure 2. Upon inspection, the leftmost summary seems to rely heavily on copying and rewriting content from the source document, as indicated by the SUMMVis color-coding of long common sequences. In contrast, the complementary summary (rightmost) seems to exhibit shorter fragments, possibly from novel sentences.

[15] We use a test set with a size of 22 documents as in the official evaluation.

[16] The values are the mean across over 10 test sets, and the standard deviation is in brackets



Summary

BART

Fine - tuning large pre - trained models is an effective transfer mechanism in NLP . However , in the presence of many downstream tasks , fine - tunating is parameter inefficient : an entire new model is required for every task . Instead , we present an alternative transfer method based on adapter modules (Rebuffi et al . , 2017) . Features - based transfer involves pre - training real - valued embeddings vectors . The parameters of the original network remain fixed , yielding a high degree of parameter sharing between tasks . Fine - tunation involves copying the weights from a network and tuning them on the downstream task . This is more parameter efficient if the lower layers of a network are shared between tasks (e.g. the weights of the top layers) . The authors propose a transfer learning strategy that yields compact and extensible downstream models . This approach is particularly useful for applications such as cloud services , where models need to be trained to solve many tasks that arrive from customers in sequence . The authors present a method to fine - tune BERT , a Transformer network trained on large text corpora with an unsupervised loss . Their method is based on a technique called adapter - based tuning , which is similar to fine tuning , but adds only a few trainable parameters per task . They argue that this approach is parameter efficient and compact , and can be used to solve new tasks without having to train a new model for every new task . Results BERT achieves

Summary

BART

Fast Text Classification with Rank Constraint FastText is a simple and efficient baseline for text classification . It is based on the Huffman coding tree and uses a softmax function to compute the probability distribution over the predefined classes . The paper shows that it can train fastText on more than one billion words in less than ten minutes using a standard multicore CPU , and classify half a million sentences among 312 K classes within less than a minute . FastText outperforms the state - of - the - art in terms of accuracy and performance on two tasks : tag prediction and sentiment analysis . How The paper uses a hierarchical softmax with a rank constraint . The softmax is trained asynchronously on multiple CPUs using stochastic gradient descent and a linearly decaying learning rate . Each node in the tree is associated with a weight matrix . The weight matrix is fed to a linear classifier . For a set of N documents , this leads to minimizing the negative loglikelihood over the classes : \\u2212 1 N N\\u2211 n=1 yn log (f (BAxn)) , where xn is the normalized bag of features of the nth document , yn the label , A and B the weight matrices . For each document , the weight matrix A is a look - up table over the words in the document . The word representations are then averaged into a text representation , which is in turn fed to the linear classifiers . This is similar to the previous work in efficient word representation learning (Mikolov et al . , 2013) . In this paper , the word representation is

Figure 2: BART summaries in the ROUGE top quartile (left) and the SPICE top quartile (right).

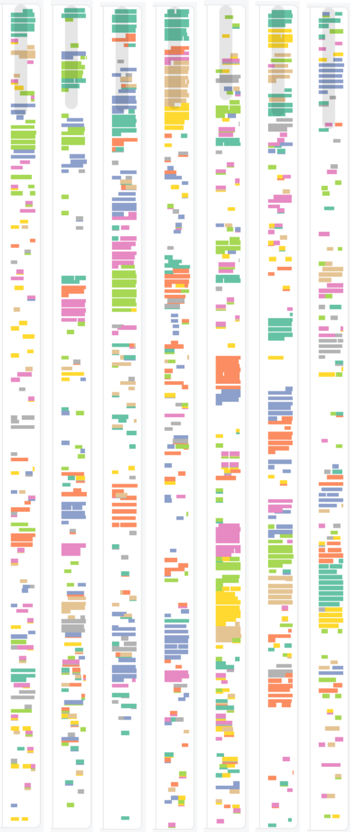


Figure 3: The first three images are ROUGE and BERTSCORE common test cases in the top quartile. The last four images are complementary high-quality summaries in top quartile suggested by SPICE and BLEU. The figures depict portions of the source document that align with the system-generated summary.

6.2 Future Work

Our results show that using multiple metrics may be beneficial in identifying summaries that are of a similar high calibre. In future work, we aim to investigate how the multiple metrics might be used in concert to evaluate systems and provide incremental intrinsic measures of progress.

We also intend to investigate how metrics like SPICE might be used to identify high-quality novel sentences, and to see if the graph comparison underpinnings allows SPICE to make qualitatively different judgments to metrics like BERTScore. Finally, we will explore other adaptations of SPICE accounting for multiple sentences in texts.

7 Conclusions

We present a detailed evaluation of multiple text summarisation metrics for long document summarisation. Utilising a oracle, baseline and state-of-the-art systems, we show that a diverse suite of metrics can capture work in a complementary fashion, so that an evaluation framework is not subject to the limitations of a single metric. In a rigorous analysis over 10 repeated trials, we show that performance of the tested approaches is roughly the same as published results. However, while some findings from the LongSumm shared task can be replicated, we find the ranking of methods in our experiments differs from prior results. When we examine the top and bottom quartiles of summarisation performance, we show that ROUGE and BERTScore are often in agreement. Further diversity in evaluation may be obtained using the metrics commonly used natural language generation and image captioning. In particular, we present preliminary results that show that the SPICE metric, which considers graph comparisons of semantic information, also agrees with the ROUGE and BERTScore metrics. We see that SPICE can identify other situations in which summarisation systems are performing well, complementing the insights gained from ROUGE and BERTScore.

8 Acknowledgement

Cai Yang is supported by Undergraduate Vacation Studentship from CSIRO. We sincerely thank both reviewers for their efforts and valuable comments in helping us improve the manuscript.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. *EACL 2006 - 11th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, pages 313–320.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Peng fei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. *ArXiv*, abs/2010.07100.

Aoife Cahill. 2009. Correlating human and automatic evaluation of a German surface realiser. *ACL-IJCNLP 2009 - Joint Conf. of the 47th Annual Meeting of the Association for Computational Linguistics and 4th Int. Joint Conf. on Natural Language Processing of the AFNLP, Proceedings of the Conf.*, (August):97–100.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

Shuyang Cao and Lu Wang. 2022. Hibrids: Attention with hierarchical biases for structure-aware long document summarization. In *ACL*.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview and insights from the shared tasks at scholarly document processing 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 214–224, Online. Association for Computational Linguistics.

X Chen, H Fang, T Y Lin, R Vedantam, S Gupta, P Dollár, and C L Zitnick. 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv:1504.00325*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2:615–621.

Dipanjan Das and Ankur P Parikh. 2019. BLEURT: Learning Robust Metrics for Text Generation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

George R. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics.

OndDušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the {{State}}-of-the-{{Art}} of {{End}}-to-{{End Natural Language Generation}}: {{The E2E NLG Challenge}}. *Computer Speech \& Language*, 59:123–156.

A. R. Fabbri, Wojciech Kryscinski, Bryan McCann, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal. pages 1302–1308.

Sayar Ghosh Roy, Nikhil Pinnaparaju, Risubh Jain, Manish Gupta, and Vasudeva Varma. 2020. Summaformers @ LaySumm 20, LongSumm 20. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 336–343, Online. Association for Computational Linguistics.

Jesús Giménez and Lluís Màrquez i Villodre. 2007. Linguistic features for automatic evaluation of heterogenous mt systems. In *WMT@ACL*.

Vivek Gupta, Prerna Bharti, Pegah Nokhiz, and Harish Karnick. 2021. Sumpubmed: Summarization dataset of pubmed scientific articles. In *ACL*.

Luyang Huang, Shuyang Cao, Nikolaus Nova Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *NAACL*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *EMNLP*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander R. Fabbri, Yejin Choi, and Noah A. Smith. 2022. Bidimensional leaderboards: Generate and evaluate language hand in hand. In *NAACL*.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2020. TalkSumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 2125–2131.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. pages 7871–7880.

Chin-Yew Lin. 2004. {ROUGE}: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Feifan Liu and Yang Liu. 2008. Correlation between rouge and human evaluation of extractive meeting summaries. *ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, (June):201–204.

Yang Liu and Mirella Lapata. 2019. 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. Association for Computational Linguistics.

Stuart Mackie, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2014. On choosing an effective automatic evaluation metric for microblog summarisation. *Proceedings of the 5th Information Interaction in Context Symposium, IIiX 2014*, pages 115–124.

A. Mutton, M. Dras, S. Wan, and R. Dale. 2007. GLEU: Automatic evaluation of sentence-level fluency. In *ACL 2007 - Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Ani Nenkova and Rebecca Passonneau. Evaluating Content Selection in Summarization : The Pyramid Method.

Jekaterina Novikova and Verena Rieser. 2018. Findings of the E2E NLG Challenge. 17:322–328.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, 2(1):41–45.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. *CoRR*, abs/1704.0.

Sajad Sotudeh, Arman Cohan, and Nazli Goharian. 2021. On generating extended summaries of long documents. In *The AAAI-21 Workshop on Scientific Document Understanding (SDU 2021)*.

Sajad Sotudeh Gharebagh, Arman Cohan, and Nazli Goharian. 2020. GUIR @ LongSumm 2020: Learning to generate long summaries from scientific documents. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 356–361, Online. Association for Computational Linguistics.

Lucia Specia and F Astudillo. 2018. Findings of the WMT 2018 Shared Task on Quality Estimation. 2:689–709.

Elior Sulem, Omri Abend, and Ari Rappoport. 2016. BLEU is Not Suitable for the Evaluation of Text Simplification.

Wenyi Tay, Aditya Joshi, Xiuzhen Zhang, Sarvnaz Karimi, and Stephen Wan. 2019. Red-faced ROUGE: Examining the Suitability of ROUGE for Opinion Summary Evaluation. *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 52–60.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Jesse Vig, Wojciech Kryściński, Karan Goel, and Nazneen Fatema Rajani. 2021. SummVis: Interactive Visual Analysis of Models, Data, and Evaluation for Text Summarization.

Senci Ying, Zheng Yan Zhao, and Wuhe Zou. 2021a. LongSumm 2021: Session based automatic summarization model for scientific document. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 97–102, Online. Association for Computational Linguistics.

Senci Ying, Zheng Yan Zhao, and Wuhe Zou. 2021b. LongSumm 2021: Session based automatic summarization model for scientific document. pages 97–102.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and others. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: Pre-Training with extracted gap-sentences for abstractive summarization. *37th International Conference on Machine Learning, ICML 2020*, PartF16814:11265–11276.

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