RISE: Leveraging Retrieval Techniques for Summarization Evaluation

David Uthus
Google Research
duthus@google.com

Jianmo Ni
Google Deepmind
jianmon@google.com

Abstract

Evaluating automatically-generated text summaries is a challenging task. While there have been many interesting approaches, they still fall short of human evaluations. We present RISE, a new approach for evaluating summaries by leveraging techniques from information retrieval. RISE is first trained as a retrieval task using a dual-encoder retrieval setup, and can then be subsequently utilized for evaluating a generated summary given an input document, without gold reference summaries. RISE is especially well suited when working on new datasets where one may not have reference summaries available for evaluation. We conduct comprehensive experiments on the SummEval benchmark (Fabbri et al., 2021) and a long document summarization benchmark. The results show that RISE consistently achieves higher correlation with human evaluations compared to many past approaches to summarization evaluation. Furthermore, RISE also demonstrates data-efficiency and generalizability across languages.

1 Introduction

Summarization evaluation has became a topic of interest in recent years. In the past, many summarization approaches have relied on ROUGE (Lin, 2004) for evaluating generated summaries. Yet, as reported by Fabbri et al. (2021), ROUGE and other automated metrics tend to fall short when compared to human evaluations. To overcome this, many new approaches have been developed leveraging pre-trained language models, showing various degrees of success.

We present our new approach to summarization evaluation called Retrieval-Inspired Summarization Evaluation (RISE). As with recent approaches, RISE leverages pre-trained language models. But unlike past approaches, we treat evaluation as a retrieval task, leveraging techniques from information retrieval. This is done by using a dual-encoder approach (Gillick et al., 2019; Ni et al., 2022), feeding in the source document and the summary to be evaluated, in order to get a final score for evaluation.

The benefits of RISE are as follows:

- Our experiments show that RISE strongly correlates with human metrics, outperforming many recent approaches. It works well when fine-tuned with in-domain data, when transferred to new domains, transferred to new languages, or on a small amount of data.
- It has the benefit of not being reliant on reference summaries during evaluation or output calibration. This allows it to work well in new domains or online use cases, where it may be expensive or impractical to obtain reference summaries.
- RISE can be further improved as better pre-trained language models are released in the future. We have released checkpoints and code to evaluate with our models and train users' own evaluation models.[1]

2 Related Work

There has been a lot of work in recent years on evaluation of summarization. These approaches have been diverse, and there are different ways of categorizing them. For example, Yuan et al. (2021) grouped evaluation metrics into four categories: matching, regressions, ranking and generation. In this case, RISE falls under ranking. One of the idealistic benefits of creating such a ranking approach is that while we need references during training of such a model, for evaluation we can focus on just the source document and the generated summary to be evaluated.

[1] https://github.com/google-research/google-research/tree/master/rise

Findings of the Association for Computational Linguistics: ACL 2023, pages 13697–13709
July 9-14, 2023 ©2023 Association for Computational Linguistics

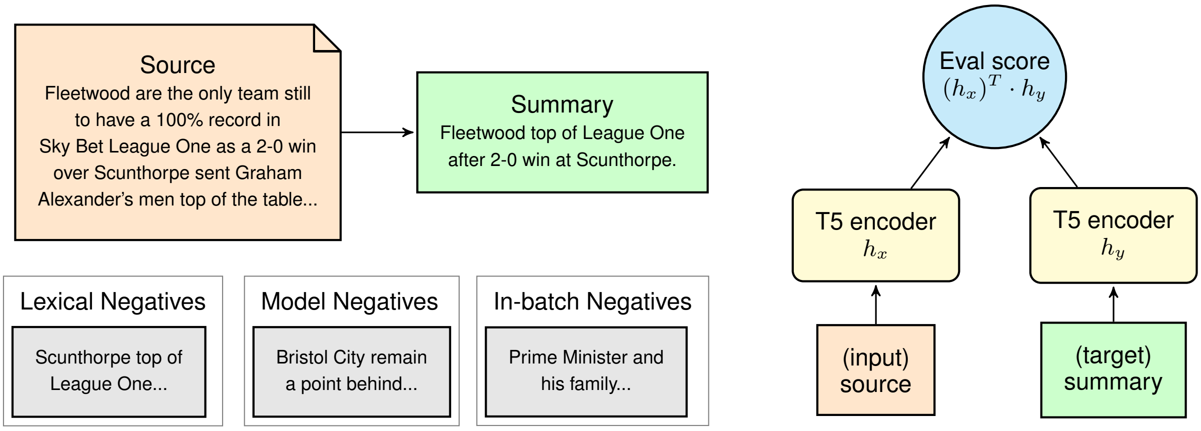
Source
Fleetwood are the only team still to have a 100% record in Sky Bet League One as a 2-0 win over Scunthorpe sent Graham Alexander's men top of the table...
Summary
Fleetwood top of League One after 2-0 win at Scunthorpe.
Lexical Negatives
Scunthorpe top of League One...
Model Negatives
Bristol City remain a point behind...
In-batch Negatives
Prime Minister and his family...
Eval score
$(h_x)^T \cdot h_y$
T5 encoder
h_x
T5 encoder
h_y
(input) source
(target) summary

Figure 1: Diagram of RISE. RISE first depends on an source document and a summary. As part of the training process, it will also require example negatives. These can be either in-batch negatives, lexical negatives (via augmenting the original summary), or model negatives (via mining for similar negatives with a trained model). When evaluating a summary, RISE takes in the source document as input and summary to be evaluated as the target. It will encode both input and summary with a T5 encoder. Finally, it will take the dot product of these encodings, resulting in the score we use for evaluating the summary.

Method	Source-free	Reference-free	Model-based
ROUGE	✓	✗	✗
CHRF	✓	✗	✗
BERTScore	✓	✗	✓
SMART	✗	✗	✓
T5-ANLI	✗	✓	✓
BARTScore	✗	✓	✓
RISE	✗	✓	✓

Table 1: Comparisons of different summarization evaluation methods. Source-free methods often requires golden summaries as reference for evaluation; meanwhile reference-free methods only rely on the source input, which is more practical when gold summaries are hard to obtain.

In this paper, we are more focused on metrics that are reference-free or reference-dependent. Table 1 shows how we can group such metrics. Note that some of the metrics were designed specifically for summarization, while others were designed for generation in general. This is also not an exhaustive list of all metrics.

Some of the metrics rely on comparing a generated summary with a reference summary, such as ROUGE (Lin, 2004), CHRF (Popović, 2015), and BERTScore (Zhang et al., 2020). Others rely on comparing the generated summary with the source document, such as T5-ANLI (Honovich et al., 2022), BARTScore (Yuan et al., 2021), and the work we present here, RISE. SMART (Amplayo et al., 2022), unlike these other metrics, can compare with both the source and reference when computing metrics.

More recently, BARTScore has been proposed as a competitive reference-free evaluation method that leverages the power of pre-trained language models. Given an input document, it computes the likelihood of generating the summary from the BART model and then use the likelihood as the quality score. The benefits of this approach is that it can work with a pre-trained language model without requiring any finetuning, though finetuning does help improve its performance for specific datasets. A drawback of this approach is that the metric scores are challenging to interpret, making them less practical to use for quality filtering or calibration of summarization models.

The benefit of being a reference-free approach is that it can work well when one needs to evaluate a new generated summary where there is no reference summary available. But doing so is a more challenging task, especially when needing to handle longer inputs of source documents (which can be further challenging when the source document is very long).

3 Model

Figure 1 shows how RISE works. As previously mentioned, RISE leverages pre-trained language models via a dual-encoder network. To do so, RISE builds upon T5X Retrieval[2] as the framework for the dual-encoder model. Each encoder in the model is the encoder of a pre-trained T5 model (Raffel et al., 2019). T5 is an encoder-decoder model, thus we are only using the encoder half of the model.

To evaluate a summary, RISE will feed in the source document d_i into one of the encoders (the left encoder in Figure 1), and the summary s_i to be evaluated into the other encoder. Both documents are encoded, and finally we take the dot product of the resultant output encodings in order to get the final score used for evaluation.

To train the dual-encoder models, we apply contrastive learning and use the positive pairs of (document, summary). Assume s_i^+ is considered as a positive summary for document d_i. During training, all other summaries in a batch are considered as negatives. The models are trained using an in-batch sampled softmax (Henderson et al., 2017):

$$\mathcal{L} = -\log \frac{e^{\text{sim}(d_i, s_i^+)/\tau}}{\sum_{j \in \mathcal{B}} e^{\text{sim}(d_i, s_j^+)/\tau}}. \quad (1)$$

The similarity scoring function *sim* is the cosine distance in our experiments[3]. At inference time, the similarity scores are used to estimate the quality of generated summaries. $\mathcal{B}$ is a mini-batch of examples and τ is the softmax temperature. During training, we could prepare additional negatives s_i^- for each document d_i, and the loss can be computed as:

$$\mathcal{L} = -\log \frac{e^{\text{sim}(d_i, s_i^+)/\tau}}{\sum_{j \in \mathcal{B}} e^{\text{sim}(d_i, s_j^+)/\tau} + e^{\text{sim}(d_i, s_i^-)/\tau}}. \quad (2)$$

By building on top of T5 architecture with T5X Retrieval (Ni et al., 2022), this gives us the benefit of being able to leverage different pre-trained T5 models. This includes the original T5 model for tasks with shorter inputs, LongT5 (Guo et al., 2022) for tasks with longer inputs, and mT5 (Xue et al., 2021) for multilingual tasks.

3.1 Training

RISE, being built upon T5X Retrieval, needs to be finetuned so that it can learn to score a summary given an input source document. To do so, we can make use of various summarization datasets for finetuning. There are three strategies for finetuning that we have explored:

- Training only on in-domain data - can see how well the model performs if only trained on the targeted task.
- Training only on out-of-domain data - can see how well the model transfers to new domains.
- Training on both in- and out-domain data - see how well the model performs if trained on a mixture of data outside the domain and within the domain.

3.2 Generating Negatives

Given a suitable dataset, we then can finetune RISE as a retrieval task. As RISE is being trained as a retrieval model, it requires example negatives during training to help differentiate the correct summary to "retrieve" given a set of summaries. By default, when finetuning one can use other summaries within a batch as candidate summaries. Additionally, a model can be trained with additional hard negatives per example. We looked at three possibilities for generating these (illustrated in Figure 1): lexical negatives, model negatives, or a combination of both.

3.2.1 Lexical Negatives

Lexical negatives are example negatives generated by augmenting a reference summary. The goal is to augment the summary in such a way that it exhibits characteristics that make it a poor summary compared to the reference. This would then allow the model to learn to differentiate a good summary from a poor summary.

For data augmentations, we looked at several methods for augmenting the original summary:

- Swapping noun entities - randomly swap noun entities in the summary with one from the original source document.
- Shuffling words - randomly shuffle the words in the summary.
- Dropping words - randomly drop words from the summary.
- Dropping characters - randomly drop characters from the summary.
- Swapping antonyms - swap words with their antonyms.

[2] https://github.com/google-research/t5x_retrieval

[3] Specifically, we apply l2-normalization to the document and summary encodings, then compute their dot-product.

One benefit of working with augmentations is that we only need to train a RISE model once with the augmented negatives. The augmentations can be done in an offline manner, and thus reused for multiple experiments. Once we have generated these augmentations, we then need to train a RISE model once on a given dataset augmented with negatives, and the resulting model will be our final model to be used for evaluating summaries.

3.2.2 Model Negatives

Model negatives are example negatives mined from a dataset. The goal is to find negative examples that are similar to a reference summary, so that the model can learn to better differentiate these similar summaries.

To create this set of model negatives, we first finetune a RISE model on a dataset, and then we can mine within a dataset for similar summaries for each source document. To do so, we encode all the documents and summaries, then for each document, find the top n most similar summaries (excluding the associated summary of a document). After we have finished mining for negatives, we then need to train a second model with the dataset that now contains the model negatives. Once we have trained this second model, we can then use it for evaluating summaries.

The benefits of model negatives is that one is not dependent on needing to create methods for augmenting data. It can also find similar summaries for a given reference that would not be achievable via augmentation, thus providing a model with a broader source of negatives. The drawbacks though of model negatives is that one is required to train a model twice, once to be used for negative mining, and a second time for the final model to be used for evaluation.

3.2.3 Combining Lexical and Model Negatives

It is possible to combine the two above approaches for obtaining negatives. Doing so would allow us to leverage the strengths of both types of negatives.

To do so, we first finetune a RISE model on the lexical negatives. This resulting model is then used for the above mining process to find model negatives. These model negatives are combined with the lexical negatives, creating a larger negative set for each example. Then we finetune a final model on the combined dataset.

4 Results

We use SummEval (Fabbri et al., 2021) to evaluate how well our approach correlates with human evaluations. SummEval is a collection of human annotations on the quality of 16 models and their outputs for 100 examples from the CNN / Daily Mail task (Nallapati et al., 2016). As with past approaches, we focus on the annotations made by expert annotators, and use Kendall's tau for system-level correlation.

As described in their work, there are four criteria used for human judgements:

- Coherence – the collective quality of the sentences in the summary.
- Consistency – the consistency of the facts between the source and summary.
- Fluency – the quality of the individual sentences in the summary.
- Relevance – the selection of important content in the summary from the source.

4.1 Methodology

For our various experiments, we make use of T5 (specically version T5.1.1) when working with shorter contexts, LongT5 for longer contexts, and mT5 for multilingual tasks. For T5 and mT5, we use input lengths of 4,096 for the input document, and for LongT5 we use input lengths of 16,384. For all variants, we use a input length of 512 for the summary.

All models were trained for 30,000 steps; a batch size of 64; and the same T5 default learning rate with warmup steps set to 1500, base learning rate set to 0.001, and a decay factor of 7e−5. All models were also trained on the full training set for each respective dataset, with exception to Section 4.3.4, in which we trained on partial datasets.

For exploration, we make use of a variety of abstractive summarization datasets. This allows us to see how well RISE works whether the task is included or not within the training. The training sets included are CNN / Daily Mail (Nallapati et al., 2016), Multi-News (Fabbri et al., 2019), arXiv (Cohan et al., 2018), PubMed (Cohan et al., 2018), BigPatent (Sharma et al., 2019), SAMSum (Gliwa et al., 2019), Reddit TIFU (Kim et al., 2019), and MLSUM (Scialom et al., 2020).

These datasets also allow us to explore situations of both short and long contexts. CNN / Daily Mail, SAMSum, Reddit TIFU, and Multi-News can be used for training models of shorter context, while arXiv, PubMed, BigPatent and Multi-News[4] can be used for longer context. MLSUM is used for testing on multilingual tasks. Unless otherwise noted, all results are shown for Large-sized models.

For generating augmentations, specifically noun entities, we make use of spaCy v3.0[5], and for swapping antonyms, we make use of NLTK v3.7(Loper and Bird, 2002). For swapping noun entities, each entity noun seen in a summary example will be swapped with 50% chance. For dropping words or dropping characters, we drop at 20% chance. These random values were chosen arbitrarily and not further optimized.

All results shown are for the four human metrics of coherence, consistency, fluency, and relevance, along with the average of these metrics, making it easier to compare approaches.

4.2 SummEval Comparisons

Table 2 show the results of our work with past approaches: ROUGE (Lin, 2004), chrF (Popović, 2015), SMS (Clark et al., 2019), BARTScore (Yuan et al., 2021), SMART (Amplayo et al., 2022), BLEURT (Sellam et al., 2020), BERTScore (Zhang et al., 2020), Q^2 (Honovich et al., 2021), T5-ANLI (Honovich et al., 2022), and PRISM (Thompson and Post, 2020). The scores shown for these past models are those reported by the recently-published SMART paper, in which we use the same methodology for evaluation. We have grouped the approaches depending on whether they are reference-dependent or reference-free metrics, i.e., if they need a reference in order to be able to evaluate a score for a generated summary.

We show results from three of our RISE variants – those trained on CNN/DM , Multi-News, and SamSUM. We show those trained with a mixture of lexical negatives composed of 5 negatives with swapped entities and 5 negatives with randomly dropped words. This configuration was shown to show strong performance and able to transfer well across datasets, as explained later in Section 4.3. These models are all Large-size.

As can be seen, all three variants show high correlation scores, particularly for consistency, fluency and relevance. As expected, finetuning on in-domain data, in this case on CNN/DM, showed the strongest results. Notably, comparing with other models that fine-tuned on CNN/DM, e.g., BARTScore-CNN, RISE achieves an absolute improvement of +16.7 points on the average metrics.

Metric	Coh	Con	Flu	Rel	Avg
Reference-dependent metrics					
ROUGE-1	.350	.550	.527	.583	.503
ROUGE-2	.233	.600	.494	.433	.440
ROUGE-L	.117	.117	.259	.350	.211
BLEU	.217	.050	.326	.383	.244
chrF	.350	.617	.561	.550	.519
BERTScore	.333	-.030	.142	.200	.161
MoverScore	.217	-.050	.259	.350	.194
BLEURT	.533	.200	.410	.467	.403
SMS	.267	.600	.360	.400	.407
SMART-1	.433	.667	.644	.667	.603
SMART-2	.417	.750	.628	.583	.594
SMART-L	.567	.567	.611	.733	**.619**
Reference-free metrics					
PRISM	.233	.600	.360	.367	.390
T5-ANLI	.250	.583	.544	.517	.473
BARTScore	.350	.617	.494	.450	.478
BARTScore+CNN	.550	.317	.594	.583	.511
Q^2	.250	.750	.577	.450	.507
RISE$_{Multi-News}$	.533	.733	.711	.700	.669
RISE$_{SamSUM}$	.533	.700	.678	.700	.653
RISE$_{CNN}$	.533	.733	.745	.700	**.678**

Table 2: Results comparing past approaches with some of the RISE variants. For the RISE variants, these are Large-sized models finetuned on a given dataset using lexical negatives, composed of a mixture of 5 summaries with swapped entities and 5 summaries with randomly dropped words for each dataset example. SMART metrics are those reported when using BLEURT as the string matcher.

Additionally, we can see that RISE also performs well when finetuned on other, out-of-domain data, showing how well the model transfers to new summarization datasets. While Multi-news is a bit similar to CNN/DM, in that both are in the domain of news articles, SamSUM is a dialogue summarization corpus and RISE stills transfer well when tested again the CNN/DM-based SummEval dataset.

Comparing with other metrics, first examining the similar reference-free metrics, we can see RISE performs more strongly than any past approach. This is important when working in domains where one may want to evaluate new inputs that do not have a reference summary. RISE also compares well with reference-dependent metrics, performing slightly better than the best metric SMART.

[4] As reported by Guo et al. (2022), Multi-News when tokenized has on average 1,902 tokens and 4,853 at 90^{th} percentile, thus can be used with T5 when input limit is set to 4096 and also for LongT5 with its longer limits of 16k.

[5] https://spacy.io/

Augmentations	Coh	Con	Flu	Rel	Avg
		Multi-News			
SE	.467	.733	.678	.667	.636
SW	.350	.617	.561	.550	.519
DW	.467	.733	.678	.667	.636
DC	.367	.633	.577	.567	.536
SA	.400	.633	.577	.600	.553
SE+DW	.533	.733	.711	.700	.669
		SamSUM			
SE	.600	.633	.644	.633	.628
SW	.333	.633	.544	.567	.519
DW	.417	.750	.661	.650	.619
DC	.317	.617	.527	.550	.503
SA	.383	.683	.594	.617	.569
SE+DW	.533	.700	.678	.700	.653
		CNN/DM			
SE	.733	.467	.644	.633	.619
SW	.400	.667	.577	.567	.553
DW	.467	.733	.711	.667	.644
DC	.417	.650	.561	.550	.544
SA	.483	.550	.494	.483	.503
SE+DW	.533	.733	.745	.700	.678

Table 3: Results of different augmentations. SE is for swapping entities with source, SW is for randomly swapping words, DW is for randomly dropping words, DC is for randomly dropping characters, and SA is for randomly swapping words with their antonyms.

Negatives	Coh	Con	Flu	Rel	Avg
		Multi-News			
None	.250	.550	.494	.450	.436
SE+DW	.533	.733	.711	.700	.669
Mined 5	.367	.700	.577	.533	.544
Mined 10	.367	.667	.577	.533	.536
Mined 15	.417	.683	.594	.583	.569
Mined 20	.383	.650	.594	.583	.533
SE+DW+Mined 5	.517	.717	.695	.650	.644
		SamSUM			
None	.283	.550	.494	.483	.453
SE+DW	.533	.700	.678	.700	.653
Mined 5	.333	.600	.544	.533	.503
Mined 10	.283	.583	.494	.517	.469
Mined 15	.300	.600	.510	.533	.486
Mined 20	.317	.583	.527	.517	.486
SE+DW+Mined 5	.483	.750	.661	.683	.644
		CNN/DM			
None	.417	.683	.561	.583	.561
SE+DW	.533	.733	.745	.700	.678
Mined 5	.433	.700	.611	.633	.594
Mined 10	.450	.683	.628	.650	.603
Mined 15	.367	.667	.544	.567	.536
Mined 20	.450	.717	.628	.617	.603
SE+DW+Mined 5	.550	.717	.762	.750	.695

Table 4: Results of using no negatives (None), using augmentation, mining for negatives, and combining augmentations with mined negatives.

4.3 Ablations

We explore four ablations to see their impact on RISE – impact of lexical and model negatives, size of the pre-trained model, datasets used for finetuning, and the size of the datasets used. All ablation experiments are done using the Large-size model, with exception to Section 4.3.2 in which we explore the impact of model sizes.

4.3.1 Lexical and Model Negatives

We first look at how lexical and model negatives impact the performance of RISE.

Table 3 shows results of looking at different augmentations, focusing on CNN/DM, SamSUM, and Multi-News for the task. As can be seen, swapping entity nouns and randomly dropping words perform the best as stand-alone augmentations for both tasks. Combining the two (i.e., having 5 of each type of augmentation) results in even stronger performance.

Table 4 shows comparisons with the same datasets when looking at either no negatives (i.e., only relying on in-batch negatives), using lexical negatives, using model negatives, or combining lexical and model negatives. As can be see, lexical negatives have the largest impact for all task domains. More surprisingly is that with working with the combined negatives, CNN/DM performs better while Multi-News and SamSUM performs worse. We believe this is due to the model negatives helps the model when focused on the same task, which in turn makes the model less transferable to other domains. While model negatives by themselves do not show as strong of a performance as lexical negatives, they can still be valuable if it is too expensive to create lexical negatives, or if one wants to train a model that is focused on a given domain and does not need the model to transfer to other domains.

4.3.2 Model Sizes

Table 5 shows results when looking at Base and Large model sizes. The advantage of using Base is that, being smaller, it requires less computation for evaluating summaries. And as can be seen, Base models tend to perform adequate enough if focused on in-domain data, but do not transfer as well to other domains when compared to Large. Even though Large models do come with overhead of requiring more computation, we believe it is worth the trade off to get the much stronger performance, while at the same time not being too prohibitive in

Size	Coh	Con	Flu	Rel	Avg
Multi-News					
Base	.367	.333	.276	.267	.311
Large	.533	.733	.711	.700	.669
SamSUM					
Base	.483	.417	.393	.383	.419
Large	.533	.700	.678	.700	.653
CNN/DM					
Base	.600	.433	.477	.467	.494
Large	.533	.733	.745	.700	.678

Table 5: Results of comparing Base and Large model sizes.

Dataset	Coh	Con	Flu	Rel	Avg
Multi-News	.533	.733	.711	.700	.669
SamSUM	.533	.700	.678	.700	.653
Reddit - TLDR	.500	.733	.711	.667	.653
Reddit - Title	.467	.767	.695	.633	.640
CNN/DM	.533	.733	.745	.700	.678
Mixed - CNN	.567	.633	.611	.600	.603
Mixed + CNN	.533	.733	.711	.667	.661

Table 6: Results of various datasets when tested on SummEval. Mixed datasets are mixing the datasets within the table, either without or with CNN/DM. These are all reporting results when using lexical negatives of 5 with swapped entities and 5 with randomly-dropped words.

computation costs.

4.3.3 Datasets

We next look at the impact of training RISE on different datasets.

Table 6 shows results when running on different datasets that are applicable with a T5 model. This also includes mixing datasets, to see whether we can benefit from mixtures. As the results first show, RISE shows it can transfer well across different datasets. While Multi-News is similar to the CNN/DM dataset used in SummEval, the other datasets SamSUM and the two Reddit variants are rather different. Suprisingly though, the model does not perform as well when mixing datasets – RISE works better if trained on just a single dataset, either for transfering to a new domain or within a domain.

Table 7 shows results when training on datasets with LongT5. As shown, RISE with LongT5 does perform worse than that of T5 – this can be expected as T5's full attention is better suited (this is also supported in the original LongT5 paper, where LongT5 on CNN/DM did not perform as well against other datasets when compared to training on long-context datasets). Despite the weaker performance, the model still performs comparable to other reference-free metrics, many of which we do not know how well they would scale up to long-context datasets. In terms of individual models, the model trained on Multi-News did slightly worse despite being news-related. This may be that LongT5 is better able to capture the full input, thus handling multiple documents, which differs from the other datasets presented and CNN/DM, in that they are all of a single document input.

Dataset	Coh	Con	Flu	Rel	Avg
Multi-News	.483	.350	.427	.383	.411
arXiv	.400	.633	.544	.533	528
PubMed	.517	.617	.594	.583	.578
BigPatent	.517	.450	.460	.417	.461

Table 7: Results of various datasets using LongT5 when tested on SummEval. These are all reporting results when using negatives of 5 with swapped entities and 5 with randomly-dropped words.

Dataset	Coh	Con	Flu	Inf	Avg
MLSUM-ES	.483	.683	.661	.650	.619
MLSUM-DE	.450	.717	.628	.583	.594
MLSUM-FR	.550	.683	.695	.650	.644

Table 8: Results of multilingual datasets when tested on SummEval. These are all reporting results when using negatives of 5 with swapped entities and 5 with randomly-dropped words.

Table 8 shows the results of finetuning on the multilingual summarization task MLSUM. We finetuned on 3 of the languages within this task, Spanish (ES), German (DE), and French (FR). As can be seen, despite having been trained on other languages, RISE still shows strong correlation with human metrics when applied to an English dataset.

4.3.4 Dataset Sizes

As a final ablation, we examined how well the model performs even with a reduced amount of data. Table 2 shows the results of these experiments. As can be seen, the model still does well with reduced-data for the 3 datasets we trained upon. Only when trained on Multi-News with 512 examples do we see a bit of drop off in performance. This indicates that the model can learn well in domains where one might not have much data as with the full datasets used in this paper.

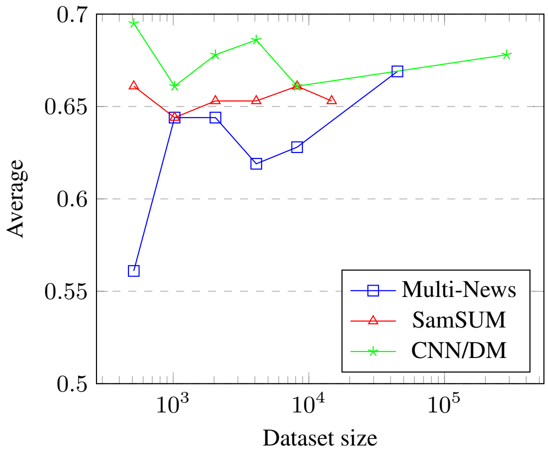

Figure 2: Results of looking at reduced amount of data for training. All three datasets were trained with reduced sizes of 512, 1024, 2048, 4096, and 8192, compared to the full set size of 44,972 for Multi-News, 14,732 for SamSUM, and 287,113 for CNN/DM.

Size	Rel	Fac	Avg
	arXiv		
BARTScore	.12	.24	.17
SMART	.45	.38	.41
RISE$_{arXiv}$	.75	.33	.54
RISE$_{PubMed}$	.74	.66	.70
RISE$_{BigPatent}$	.67	.50	.59
RISE$_{Multi-News}$	.57	.71	.64
	GovReport		
BARTScore	.25	.06	.12
SMART	.31	.26	.28
RISE$_{GovReport}$ - lexical negs	.39	.15	.27
RISE$_{GovReport}$ - combined negs	.61	.22	.42

Table 9: Results comparing past approaches with RISE on longer documents of arXiv and GovReport. Note that the metrics here are using Spearmann rank correlation.

4.4 arXiv and GovReport Comparisons

As an additional comparison, we look at longer documents annotated by Koh et al. (2022). In this work, the authors had human raters annotating various summaries of models on arXiv and GovReport (Huang et al., 2021). They were looking at two metrics, relevance and factual consistency. For these evaluations, we use Spearmann rank correlation.

Table 9 shows the results of comparing RISE with these human evaluations. We compared RISE with the results of BARTScore and SMART. For the arXiv dataset, we can see that applying our various models trained on LongT5 show higher average correlations.

The GovReport dataset is a bit different than past datasets, in that its summaries are much longer. When tokenized, the median summary for arXiv is 249, while GovReport is 657. To allow for a model that can handle these longer summaries better, we finetuned RISE on GovReport. As shown in Table 9 in the bottom half, RISE is then able to perform well in correlation with the human evaluations. More importantly, using only lexical data puts it on par with SMART, but when we add the model negatives (creating a combined lexical and model negatives data), this results in much stronger correlations.

4.5 Overall Recommendations

There are many ways to train RISE, and different model architectures one can use.

Given the results, we first recommend using architecture that matches the length and types of inputs. For short inputs, it is best to use a model trained on T5; for multilingual inputs, it is best to use a model trained on mT5; and for long inputs, it is best to use a model trained on LongT5.

What type of data to use also depends how one is expecting to use the model for evaluation. If one needs a model only focused on a given domain, then training with both lexical and model negatives gives best results. If one needs a model that can transfer to other domains, then it is best to use just lexical negatives.

We have released checkpoints from many of the models presented in this paper, allowing for one to reuse this work for their own evaluations on datasets commonly used in summarization.

5 Conclusion

We have presented our new model RISE for evaluating text summaries. As the results show, RISE has strong correlation with human evaluations. Being a reference-free metric, it can be used in new domains where generating golden summaries may be prohibitive. And while RISE shows strong correlation with human evaluations, we do not view RISE as a replacement of other metrics. Instead, we view it as complementary, especially to reference-dependent metrics such as ROUGE, CHRF, and SMART.

Summarization evaluation continues to be a challenging problem. Leveraging data from within the domain can help though with calibration of evaluation metrics. BARTScore had earlier touched upon this, and RISE further helps show the importance of this. We hope this will help spur future research in how we can use domain data to improve such metrics.

One of the benefits of RISE not explored but left for future work is how RISE can be tailored to address specific needs. Since RISE depends on contrastive learning, one can create negatives that reflect characteristics they want to specifically evaluate. Another area of future work is looking at the possibility of using RISE in other generative domains outside of summarization, such as dialogue systems. Many of the techniques here for training can easily be applied to other domains, including creation of lexical and model negatives.

Limitations

As with many other model-based metrics, RISE is best suited for evaluating offline due to the expensive nature of inferring with a large model. It is not as well suited as other metrics like ROUGE or BLEU for evaluating during training or fine-tuning. We leave the exploration of using RISE for evaluation-in-the-loop kind of training for summarization models future work.

Additionally, as with other model-based metrics, it is possible that the models may have seen some of the data during pretraining as is in the eval datasets. We do not think it would be too significant, as the pretraining task (for T5/mT5 for example) is rather different than a summarization task and, more importantly, it does not include the gold reference. Thus the model would not be able to make such a connection easily despite having seen the data.

We chose to work with the T5-family of models due to the ease-of-use for others to implement and improve upon our ideas. We would expect our ideas to work just as well with other models, such as BART, mBART, Longformer, etc.

Following recent works, we have studied the evaluation based on the SummEval benchmark (Fabbri et al., 2021). In the future, we may want to build other benchmarks that covers more domains and languages to compare different methods.

Ethics Statement

RISE is built upon pre-trained language models. Any biases within these models may possibly influence the scoring of summarization models, in that it is possible biases may cause the models to rate one summary better than another.

References

Reinald Kim Amplayo, Peter J. Liu, Yao Zhao, and Shashi Narayan. 2022. SMART: Sentences as basic units for text evaluation.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy. Association for Computational Linguistics.

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. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontañón, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yunhsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. 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 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Huan Yee Koh, Jiaxin Ju, He Zhang, Ming Liu, and Shirui Pan. 2022. How far are we from robust long abstractive summarization?

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.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Maja Popović. 2015. 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. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

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

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. 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. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

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

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations, page 8

☑ A2. Did you discuss any potential risks of your work?
Ethics statement section at end

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

All of Section 3 describes the model and training of such model

☑ B1. Did you cite the creators of artifacts you used?
Throughout Sections 3 and 4

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
License/terms will be provided when code/checkpoints released upon publication

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4 discusses the datasets and benchmarks we used, which fall within their intended usage.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Did not collect any data, and the data used are commonly used public summarization datasets.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3, 4

☒ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
For the most part, we used multiple datasets as created by their original users, and thus would expect number of examples to be reported in the original papers. Only in Section 4.3.4, in which we used a reduced number of examples (as part of an abalation), did we report the numbers used.

C ☑ Did you run computational experiments?

4

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We did not keep track of computational budget for these experiments, thus not reported

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.