PrefScore: Pairwise Preference Learning for Reference-free Summarization Quality Assessment

Ge Luo and **Hebi Li** and **Youbiao He** and **Forrest Sheng Bao**
Iowa State University
Ames, IA, USA
{gluo,hebi,yh54}@iastate.edu, forrest.bao@gmail.com

Abstract

Evaluating machine-generated summaries without a human-written reference summary has been a need for a long time. Inspired by preference labeling in existing work of summarization evaluation, we propose to judge summary quality by learning the preference rank of summaries using the Bradley-Terry power ranking model from inferior summaries generated by corrupting base summaries. Extensive experiments on several datasets show that our weakly supervised scheme can produce scores highly correlated with human ratings.

1 Introduction

Summarization is a task in natural language processing in which automatic systems generate summaries from documents. To judge the quality of system-generated summaries, human evaluation is the best option, but it is non-trivial and laborious. Hence, many automatic metrics have been developed. They can be categorized as reference-based ones and reference-free ones, depending on whether reference summaries are needed in the evaluation stage.

Reference-based metrics include ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), METEOR (Banerjee and Lavie, 2005), S^3 (Peyrard et al., 2017), MoverScore (Zhao et al., 2019), BertScore (Zhang et al., 2020), etc. Calculating the lexical overlap or the embedding similarity between a system-generated summary and its corresponding human-written reference summary, they reportedly have high correlations with human assessments.

Because creating human-written reference summaries is laborious and expensive, recent works are shifting to reference-free metrics. SummaQA (Scialom et al., 2019) and BLANC (Vasilyev et al., 2020) leverage pretrained language models to carry out text understanding tasks to evaluate the helpfulness of a summary for understanding its source document. SUPERT (Gao et al., 2020b) measures the semantic similarity against a pseudo reference summary in a multi-document summarization setting. However, reference-free metrics may show a lower correlation (Fabbri et al., 2021) with human evaluation scores than some of the reference-based metrics.

To trade off between the human effort needed and the quality of the evaluation, some work pursues a pairwise preference approach which collects preference labels over sentences in documents or over summaries from a human assessor as it requires less cognitive effort than writing a reference summary or manually scoring a machine-generated summary. Zopf (2018) proposes a reference-free evaluation approach by estimating sentence-level preferences on source documents rather than directly on the generated summaries. Gao et al. (2020a) train a linear model to estimate a summary preference utility function via active preference learning to guide a reinforcement learning based summarization system. But they do not examine the learned preference model as a metric for summarization evaluation.

Inspired by human-involved pairwise preference in summarization evaluation (Zopf, 2018; Gao et al., 2020b) and simple NLP data augmentation methods like EDA (Wei and Zou, 2019), in this work, we explore reference-free summary quality assessment via pairwise preference learning using negative sampling. A pre-trained text embedding model is used in a siamese network to learn the preference utility in an end-to-end, weakly supervised fashion. The closest work to ours is LS_Score (Wu et al., 2020). We achieve improved performance by using a better-attended model, a loss function based on preference learning, and introducing a mixed transitive negative sampling strategy. In addition, we promote our work to cross-domain and multi-document settings.

We show that the learned models are competitive

Proceedings of the 29th International Conference on Computational Linguistics, pages 5896–5903
October 12–17, 2022.

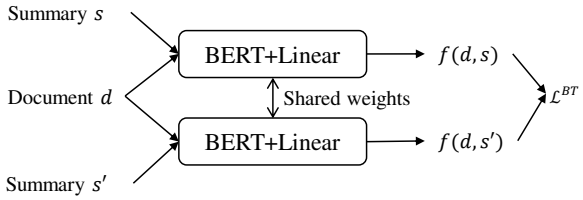

Figure 1: Model architecture.

compared to the state-of-the-art reference-free metrics. Our code is at https://github.com/NKWBTB/PrefScore.

2 Method

2.1 Model Architecture

The goal of a reference-free evaluation system is to learn a regressor f which takes a document d and its summary s as the input to produce a score $f(d, s)$ which represents the quality of the summary s. Learning such a regressor via supervised learning is very difficult because existing human-rated summary evaluation datasets (NIST, 2010; Grusky et al., 2018; Bhandari et al., 2020) contain too few samples, around 100 samples each, to train a generalizable model.

Therefore, we use pairwise preference learning as a weakly supervised workaround. By corrupting a summary into an inferior one, existing summarization datasets containing no human ratings as training labels but only gold, reference summaries can be transformed into massive training data for preference learning.

The training label is designed based on the Bradly-Terry (BT) model (Bradley and Terry, 1952). Given a reference summary s and a perturbed summary s' of the document d, the BT model estimates $f(d, s)$ and $f(d, s')$ such that the probability of s being superior than s' is:

$$p(s \succ s'|d) = \frac{\exp(f(d,s))}{\exp(f(d,s)) + \exp(f(d,s'))}. \quad (1)$$

This leads to our model design (Figure 1) using a siamese network. Leveraging the recent work of BERT-like (Devlin et al., 2019) contextualized embedding, a document d and a summary s are viewed as two sequence of tokens T_d and T_s. The input sequence are constructed as $([\text{CLS}], T_d, [\text{SEP}], T_s, [\text{SEP}])$, then the output of the [CLS] token containing both information from document and summary are sent to a linear layer to produce the final score $f(d, s)$. During the training, a pair of summaries will be sent to the siamese network. It can be seen as training a classifier to determine which summary is better. The loss is therefore:

$$\mathcal{L}^{BT} = -\sum_{d}\sum_{s' \in S'} [\log(p(s \succ s'|d))] \quad (2)$$

where S' is a set of inferior summaries deviated from s in methods to be discussed below in § 2.2. The learned ranking utility f is used as our summary evaluator and does not require a reference summary in the test/evaluation stage.

2.2 Mixed Transitive Negative Sampling

Given a reference summary s, we can obtain the set $S' = \{s'_1, s'_2, \ldots, s'_n\}$ of inferior summaries by mutating the reference summary s iteratively: s'_1 is mutated from s, s'_2 from s'_1, and so on. We can obtain a preference sequence of summaries $s \succ s'_1 \succ \cdots \succ s'_n$. The process is illustrated in Figure 2. In each iteration, unmodified tokens in s'_i is randomly selected and mutated to generate summary s'_{i+1}. The process continues until all tokens are mutated.

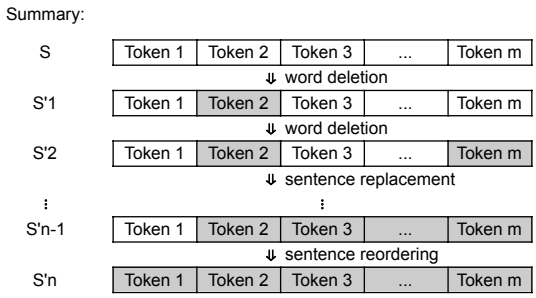

Figure 2: An example of the mixed transitive negative sampling process. The original part is in white, while the modified part is indicated as grey blocks.

Four mutation methods are employed: 1) **deleting a sentence** from the summary, resulting in information loss in the summary. 2) **replacing a sentence** in the summary with a sentence from other summaries, introducing extra information and redundancy in the summary. 3) **deleting a word** from the summary, influencing the sentence structure and readability. 4) **reorder sentences or words**, aggravating the coherence in the summary.

In each iteration, one of the four mutation methods is randomly chosen. Unlike plain negative

sampling that mutates samples in only one way or in only one iteration, our *mixed transitive negative sampling* accumulates the effects of different mutations into samples, enabling a model trained upon to learn different aspects of summaries.

3 Experiments

3.1 Test Sets

There are not many datasets with human evaluations to machine-generated summaries. Unfortunately, they are almost all in the news article domains. We use three established ones:

TAC2010 (NIST, 2010) is a multi-document summarization dataset which reports three scores: content, fluency and overall. It consists of 46 topics, each of which is associated with a set of 10 documents. We evaluate the metrics over summaries generated by 43 systems. For a summary, we calculate the mean score for all documents paired with the summary as an extension for our metric in the multi-document scenario. Only Set A for the regular summarization task is used here.

Newsroom (Grusky et al., 2018) is a single-document summarization dataset reporting four scores: INFormativeness, RELevance, COHerence and FLUence. It contains human-rated summaries generated by 7 systems for 60 documents. Each document-summary pair is rated by three human annotators. We use their mean score as the groundtruth score.

RealSumm (Bhandari et al., 2020), a recent single-document dataset reporting the LitePyramid (Shapira et al., 2019) score which is also content-focused. It sampled 100 documents from the CNN/DailyMail (See et al., 2017) test set, and collected human ratings for summaries generated by 11 extractive systems and 14 abstractive systems.

3.2 Training Sets (documents and reference summaries only, no human evaluations)

Because the test sets are all in the news domain, we select one training set from the news domain for in-domain analysis: **CNN/DailyMail** (CNNDM) (See et al., 2017). For cross-domain analysis, three training sets from different non-news domains are selected: **Billsum** (Kornilova and Eidelman, 2019) from legislative bills, **Scientific papers-ArXiv** (Cohan et al., 2018) from papers on arXiv, and **BigPatent** (Sharma et al., 2019) from patent applications.

The train splits of the four datasets are used separately to train our model. For Billsum, we used all 18,949 samples in the train split. For the other three datasets, the first 40,000 samples in the train split are used for training. For every original reference summary in the training sets, 3 negative samples (inferior summaries) are generated.

3.3 Baselines and Upperbounds

We compare our work with both reference-free and reference-based metrics. The recently developed SummaQA (Scialom et al., 2019), BLANC (Vasilyev et al., 2020), SUPERT (Gao et al., 2020b) and LS_Score (Wu et al., 2020) are our baselines because they are reference-free. [1]

Reference-based metrics serve as soft upper bounds because they are provided with extra human guides which are reference summaries. ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), S^3 (Peyrard et al., 2017), MoverScore (Zhao et al., 2019), BertScore (recall) (Zhang et al., 2020) are included in this study.

Results for LS_Score (Wu et al., 2020) are only reported for Newsroom, which is copied from their paper, as we have not succeeded in reproducing their model using their code to test on other datasets[2]. Despite the difficulty, we implemented our own version of LS_Score.

3.4 Settings

For a fair comparison, we use the same pre-trained language model BERT used by the baselines. Specifically, we use the `bert-base-uncased` variant of the BERT model in HuggingFace Transformer's Pytorch implementation. An input sequence is padded to 512 tokens with [PAD] or truncated to 512 tokens using longer input truncate first strategy and then round robin trimmer. We fine tune the model on NVIDIA RTX 3090 for fixed 16,000 steps using the Adam optimizer with the learning rate of 1e-5 and the batch size of 7.

3.5 Results

We use the summary-level (Peyrard et al., 2017) meta evaluation strategy to report an approach's

[1] By "reference-free", we mean that a reference summary is not needed to judge a machine-generated summary.

[2] Several other researchers reported the same issue `https://github.com/whl97/LS-Score/issues`. We never heard back from the authors in Email and GitHub.

Table 1: Spearman's Correlation on TAC2010.

	Content	Fluency	Overall
	Our approach		
Trained w/CNNDM	**0.5865**	0.4311	**0.5531**
Trained w/Billsum	0.4586	**0.4324**	0.4518
Trained w/ArXiv	0.4727	0.4026	0.4437
Trained w/BigPatent	0.4184	0.3695	0.4007
	Reference-free Baselines		
BLANC-tune	0.4272	0.2943	0.3966
SummaQA-F1	0.3007	0.2431	0.2864
SummaQA-CFD	0.2905	0.1516	0.2620
SUPERT	0.4794	0.3241	0.4266
	Reference-based upper bounds		
R-1	0.5597	0.2570	0.5025
R-2	0.6448	0.3490	0.5894
R-L	0.5032	0.1772	0.4463
MoverScore	0.7213	0.3522	0.6453
BertScore	0.6769	0.3634	0.6162
BLEU	0.6018	0.3462	0.5636
METEOR	0.6682	0.3371	0.6184
S3_pyr	0.7257	0.3628	0.6562
S3_resp	0.7258	0.3578	0.6520

Table 2: Spearman's Correlation on Newsroom.

	COH	INF	FLU	REL
	Our approach			
Trained w/ CNNDM	0.6507	**0.7509**	0.6079	0.6645
Trained w/ Billsum	0.6665	0.7169	**0.6557**	0.6469
Trained w/ ArXiv	**0.6758**	0.7345	0.6408	**0.6657**
Trained w/ BigPatent	0.6729	0.7309	0.6498	0.6356
	Reference-free Baselines			
BLANC-tune	0.5862	0.6881	0.5310	0.6078
SummaQA-F1	0.4895	0.5690	0.4664	0.5163
SummaQA-CFD	0.4195	0.5449	0.3719	0.4405
SUPERT	0.6171	0.6929	0.5391	0.6046
LS_Score	0.6271	0.7008	0.5852	0.6381
	Reference-based Upper bounds			
R-1	0.2310	0.3231	0.2150	0.2775
R-2	0.0861	0.1534	0.1015	0.1336
R-L	0.2055	0.3005	0.2006	0.2629
MoverScore	0.1743	0.2186	0.1431	0.2163
BertScore	0.2705	0.3156	0.2390	0.2815
BLEU	-0.0556	-0.0782	-0.0422	-0.0071
METEOR	0.1740	0.2364	0.1690	0.2437
S3_pyr	0.1929	0.2680	0.1782	0.2450
S3_resp	0.1716	0.2519	0.1717	0.2226

average correlation with human ratings over summaries. Since our method is based on preference ranking, we report the Spearman's correlation (Tables 1, 2 and 3). The best scores in the reference-free class are **bold** while top 2 and 3 are underlined. Due to the page limit, we put the extra results of significance tests in the Appendix.

On TAC2010 (Table 1), our models beat all baselines on all aspects with only one exception. In particular, our model trained with CNNDM beats all baselines on all aspects. It even further outperforms ROUGE-1 and ROUGE-L.

On Newsroom (Table 2), our models beat all baselines on all aspects with only one exception. All reference-free approaches, including ours and baselines, outperform reference-based upper bounds. This counter-intuitive result is probably due to that a reference summary mostly has only one sentence in Newsroom.

Table 3: Spearman's Correlation on RealSumm†.

	On abstractive systems	On extractive systems
	Our approach	
Trained w/ CNNDM	**0.3842**	0.1143
Trained w/ Billsum	0.3083	0.0857
Trained w/ ArXiv	0.3204	0.0929
Trained w/ BigPatent	0.3163	**0.1152**
	Reference-free Baselines	
BLANC-tune	0.3067	0.1139
SummaQA-F1	0.2173	0.0837
SummaQA-CFD	0.2433	0.0494
SUPERT	0.2532	0.0748
	Reference-based Upper bounds	
R-1	0.6266	0.2182
R-2	0.5623	0.2206
R-L	0.6035	0.2140
MoverScore	0.4951	0.1899
BertScore	0.5682	0.1920
BLEU	0.3023	0.1639
METEOR	0.6270	0.2502
S3_pyr	0.6426	0.2369
S3_resp	0.6264	0.2369

† RealSumm has only one aspect which is content-focused.

On RealSumm (Table 3), results are reported separately for abstractive and extractive systems. Our models beat all baselines on abstractive systems. All approaches perform better for abstractive summarizers than for extractive ones. Bhandari et al. (2020) ascribe this to the low inter agreement among human annotators for the extractive group.

3.6 Discussion: Domain Impact

Because our approach is training based, in-domain models which are trained with CNNDM have advantages over cross-domain models. But the advantages are only for fact-based aspects (Content for TAC2010, INF and REL for Newsroom, the whole RealSumm), not for linguistic aspects.

Among cross-domain models, which are trained with Billsum, ArXiv, and BigPatent, no one is always the best on all test sets and on all aspects. Despite the domain difference, these models still beat the baselines in nearly all cases. Such cross-domain performances suggest that our approach is domain robust.

One potential use of our approach is to train a summary quality evaluation model for a domain with no or limited summarization data.

Table 4: Experiments on Model Architectures. Spearman's correlation.

Training Set	Model Arch.	TAC 2010			Newsroom				RealSumm	
		Modified	Linguistic	Overall	COH	INF	FLU	REL	Abstractive	Extractive
CNNDM	PrefScore	**0.5865**	**0.4311**	**0.5531**	**0.6507**	**0.7509**	**0.6079**	**0.6645**	**0.3842**	**0.1143**
	S_Score	0.4567	0.3034	0.4159	0.6204	0.7404	0.5809	0.6426	0.2785	0.1104
	L+S_Score	0.4077	0.3436	0.3784	0.6338	0.7234	0.6058	0.6374	0.3085	0.1070
BigPatent	PrefScore	**0.4184**	**0.3695**	**0.4007**	**0.6729**	**0.7309**	**0.6498**	**0.6356**	**0.3163**	**0.1152**
	S_Score	0.3499	0.2160	0.3155	0.5578	0.5992	0.5326	0.5374	0.2042	0.0958
	L+S_Score	0.3663	0.2984	0.3305	0.6605	0.7020	0.6138	0.6081	0.2589	0.1074
Billsum	PrefScore	**0.4586**	**0.4324**	**0.4518**	**0.6665**	**0.7169**	**0.6557**	**0.6469**	**0.3083**	0.0857
	S_Score	0.3689	0.3368	0.3483	0.4652	0.4280	0.4577	0.3996	0.2157	0.0568
	L+S_Score	0.3518	0.3475	0.3256	0.6199	0.6956	0.5844	0.5979	0.2790	**0.1052**
Arxiv	PrefScore	**0.4727**	**0.4026**	**0.4437**	**0.6758**	**0.7345**	**0.6408**	**0.6657**	**0.3204**	0.0929
	S_Score	0.3791	0.2511	0.3511	0.5972	0.5918	0.5804	0.5078	0.2331	0.0890
	L+S_Score	0.3792	0.2591	0.3405	0.6613	0.7330	0.5963	0.6382	0.3050	**0.1109**

3.7 Bi-Encoder vs. Cross-Encoder

We further conduct experiments to analyze the impact of the model architecture on performance. LS_Score (Wu et al., 2020) uses cosine similarity of the embeddings between a document and its summary as the semantic score (S_Score) which forms a Bi-Encoder architecture. And it computes a perplexity-like score based on the summary's embedding as linguistic score (L_Score), resulting in the final score as $0.01 * L\_Score + S\_Score$. In contrast, we jointly attend a document and a summary and produce the score after a linear layer which forms a Cross-Encoder architecture.

We implement the S_Score and L+S_Score[3] of our own version. The reason for our reimplementation is not only the reproducibility issues mentioned earlier but also that we want to do an apple-to-apple comparison by using the same loss function and the negative sampling strategy.

The results of the study are shown in Table 4. PrefScore outperforms both S_Score and L+S_Score on nearly all test sets and all aspects. It is common to use the cosine similarity in the embedding space as an indicator of semantic similarity. However, it fails to fully utilize the self-attention mechanism of the transformers. By jointly attending the document and the summary, our approach (Fig. 1) can better match information in the summary to that in the document. This could be one of the reasons that PrefScore outperforms S_Score and L+S_Score under the same setting.

4 Conclusion and Future Work

In this paper, we propose to evaluate summarization quality via preference learning and transitive negative sampling. The learned models outperform other reference-free based methods in in-domain experiments and are still competitive in cross-domain experiments.

There are some possible future study directions. The negative sampling methods used in this study are rough and simple. More careful inspection can be done to observe what kind of mistakes are likely made by summarizer models and design mutation methods accordingly. Moreover, our framework uses mean scores as a workaround for the multi-document scenario; it remains an open research problem to promote our work to optimize directly for multi-document summarization evaluation. Finally, we would like to extend our method for the evaluation of other NLG tasks.

[3]We denote our version as L+S_Score to discriminate from the original LS_Score.

Acknowledgements

This work is partially supported by National Science Foundation (NSF) grants No. MCB-1821828 and No. CNS-1817089. The authors would also like to thank reviewers who have given precious feedback on improving this work.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method

of paired comparisons. *Biometrika*, 39(3/4):324–345.

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. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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.

Yang Gao, Christian M Meyer, and Iryna Gurevych. 2020a. Preference-based interactive multi-document summarisation. *Information Retrieval Journal*, 23(6):555–585.

Yang Gao, Wei Zhao, and Steffen Eger. 2020b. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online. Association for Computational Linguistics.

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

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

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.

NIST. 2010. TAC2010 guided summarization competition. `https://tac.nist.gov/2010/Summarization/Guided-Summ.2010.guidelines.html`. Accessed: 2021-08-16.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

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

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. 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 3246–3256, Hong Kong, China. Association for Computational Linguistics.

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

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. 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 682–687, Minneapolis, Minnesota. 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.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online. Association for Computational Linguistics.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Hanlu Wu, Tengfei Ma, Lingfei Wu, Tariro Manyumwa, and Shouling Ji. 2020. Unsupervised reference-free summary quality evaluation via contrastive learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3612–3621, Online. Association for Computational Linguistics.

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*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Markus Zopf. 2018. Estimating summary quality with pairwise preferences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1687–1696, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

A.1 Evaluation Settings

We utilize the SummEval (Fabbri et al., 2021) evaluation toolkit to calculate scores for metrics whose scores are not reported by a test dataset. For all metrics, we use the batch evaluation API with default parameters provided by the package. The results of the SummEval dataset are not included in this study as SummEval and RealSumm are similar datasets whose documents are both sampled from CNN/DailyMail (See et al., 2017).

A.2 Significance Tests

We perform significance tests to see if the improvement of our method over the reference-free baselines is significant. Because applying a direct test on the summary-level evaluation results is difficult, we use a bootstrap-based method to sample the documents in the test sets 1000 times to compute the p-values.

Tables 5, 6 and 7 show the p-values of the hypothesis test that "Is the PrefScore trained using the training sets in the leftmost column significantly better than the baselines at the bottom?" Numbers smaller than the significant level of 0.05 are **bold**.

Our in-domain models trained using CNNDM are significantly better than the baselines. Meanwhile, the three cross-domain models, trained with Billsum, ArXiv, and BigPatent, are significantly better than SummaQA. They are also nearly significantly better than SUPERT. No significant results are observed on extractive systems from RealSumm. We believe this is due to the low inter agreement in the extractive group as described earlier (Bhandari et al., 2020).

Table 5: p-value of Significance Test on TAC2010 Dataset.

Training Set	Content				Fluency				Overall			
CNNDM	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
BillSum	0.17	-	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.05	0.20	**0.00**	**0.00**
BigPatent	-	-	**0.00**	**0.00**	**0.00**	0.07	**0.00**	**0.00**	0.44	-	**0.00**	**0.00**
ArXiv	0.09	-	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.09	0.30	**0.00**	**0.00**
	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD

Table 6: p-value of Significance Test on Newsroom Dataset.

Training Set	COH				INF				FLU				REL			
CNNDM	**0.02**	0.10	**0.00**	**0.00**	**0.01**	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**	**0.01**	**0.00**	**0.00**
BillSum	**0.01**	0.08	**0.00**	**0.00**	0.19	0.23	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.10	0.10	**0.00**	**0.00**
BigPatent	**0.01**	0.06	**0.00**	**0.00**	0.07	0.11	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.17	0.19	**0.00**	**0.00**
ArXiv	**0.00**	0.07	**0.00**	**0.00**	0.09	0.12	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.04**	**0.03**	**0.00**	**0.00**
	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD

Table 7: p-value of Significance Test on RealSumm Dataset.

Training Set	Abstractive				Extractive			
CNNDM	**0.00**	**0.00**	**0.00**	**0.00**	0.49	0.08	0.21	0.07
BigPatent	0.38	**0.01**	**0.01**	0.05	0.51	0.08	0.22	0.08
BillSum	0.47	**0.02**	**0.01**	0.06	-	0.37	0.49	0.20
ArXiv	0.31	**0.01**	**0.01**	**0.03**	-	0.29	0.41	0.17
	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD	BLANC-tune	SUPERT	SummaQA-F1	SummaQA-CFD