Lightweight Reference-Less Summary Quality Evaluation via Key Feature Extraction

Shunan Zang[1,2], Chuang Zhang[1](✉), Jingwen Lin[1,2], Xiaojun Chen[1], and Shuai Zhang[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zangshunan,zhangchuang,linjingwen,chenxiaojun,zhangshuai}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. One of the main problems with automatic text summarization is the lack of a "gold standard" for summary quality evaluation. ROUGE [9] is the most widely used evaluation metric for summary quality. However, its evaluation merely concentrates on reference summary and overlap features of sentences rather than focusing on more critical semantic features. Some other exiting methods have issues with improper noise handling and high cost. To solve these problems, we propose a lightweight reference-less summary quality evaluation method (***SE-tiny***), which evaluates the summary from two aspects: the summary's self-quality and the degree of matching the features of the summary with the key features of the source text. Then, we optimize computational efficiency and space cost. Compared with existing methods, ***SE-tiny*** improves the quality of evaluation and reduces the cost. Besides, our method does not rely on reference summaries and can be generalized to evaluation on summarization datasets. For the goal of reproducibility, we make the ***SE-tiny*** project's code and models available.

Keywords: Automatic Text Summarization · Reference-Less Summarization Quality Evaluation · Summarization Datasets

1 Introduction

In text summarization tasks, evaluating summary quality is a challenging challenge that severely limits model performance development. Three questions need to be clarified before we can evaluate summary quality. (1) What is a summary? The summary is not the same length as the source text and contains a limited amount of information. The summary is a collection of key information in the source text, rather than an abstraction of all the information. (2) How do humans evaluate a summary quality? Humans evaluate the quality of a summary in two steps, first evaluating the summary's self-quality, and then judging how well the summary matches the key information in the source text. (3) What kind of summary evaluation method is good? A good summary evaluation method should

L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14261, pp. 498–510, 2023.
https://doi.org/10.1007/978-3-031-44198-1_41

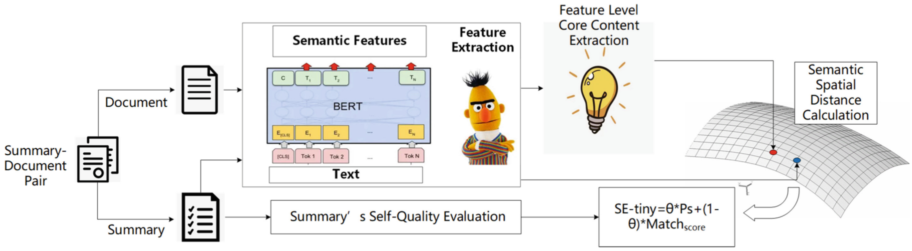
Feature Level Core Content Extraction
Semantic Features
Feature Extraction
C
T1
T2
...
TN
BERT
E[CLS]
E1
E2
EN
[CLS]
Tok 1
Tok 2
Tok N
Text
Document
Summary-Document Pair
Summary
Semantic Spatial Distance Calculation
Summary' s Self-Quality Evaluation
SE-tiny=θ*Ps+(1-θ)*Matchscore

Fig. 1. ***SE-tiny*** architecture

minimize the cost while evaluating the summary quality comprehensively and accurately.

The two primary types of summary evaluation methods are reference-based and reference-less. The former completely depends on the quality of the reference summary, which is costly and labor-intensive. The latter uses the source text information to match the summary, saving the cost of the reference summary. However, current reference-less methods share common drawbacks. Firstly, the noise information in the source text is not handled well, which affects the evaluation results. Second, the amount of information in the summary differs from that in the source text. Using all the semantic information of the source text to match the summary not only violates human logic but also fails to generate correct matching results. Furthermore, most of the well-performing reference-less methods are based on complex models, which incurs a large cost.

In response to the above problems, we propose ***SE-tiny***. Taking inspiration from human evaluation, the method is divided into three steps. (1) Evaluate the summary's self-quality in terms of compression ratio, fluency, and readability. (2) To acquire the representations of the summaries and source texts, we first map them to the feature space. Then, we extract key features from the features of the source text (simulate the extraction of key information from the source text during the human evaluation) and use these key features to perform similarity calculations with the features of the summary. This not only increases evaluation accuracy, but also aligns with the logic when humans evaluate summaries. Moreover, compared to existing methods that directly match summary and entire source text information, this method greatly simplifies computation. (3) We adopt a linear method to fuse the results of step1 and step2. Besides, we use two lightweight language models, n-gram and BERT-base [3] to reduce the cost of model loading. Experimental results show that ***SE-tiny*** not only guarantees the quality of the evaluation but also reduces the cost.

Contributions. (1) We propose a lightweight reference-less summary quality evaluation method (***SE-tiny***), which fully utilizes the source text information, removes noise, and conforms to the logic of human evaluation. (2) We optimize ***SE-tiny*** from two aspects: the computational complexity and the cost of the model. (3) We construct a high-quality dataset (***SE-tiny-db***) for evaluation of reference-less summarization and a reference-less summary evaluation system

(***SE-tiny-S***) that can be easily invoked. (4) Since ***SE-tiny*** does not rely on reference summaries, we generalize it to evaluation on summarization datasets.

2 Background and Motivation

2.1 Reference-Based Evaluation Metric

Model-Free Metrics: ROUGE [9] is a widely used metric, which evaluates summary using the co-occurrence of n-grams between machine summary and reference summary.

Model-Based Metrics: ROUGE-WE [13] uses the Word2Vec, and cosine distance to calculate the similarity of the two words, and then obtains the similarity between the machine summary and the reference summary. ROUGE-G [17] is a summary evaluation metric based on graph semantic matching. BERTScore [22] uses the contextual embeddings in the BERT [3] to represent the words and then calculates the similarity between the machine summary and the reference summary by the cosine similarity. MoverScore [23] calculates the similarity between the machine summary and the reference summary by the Word Mover distance. Based on MoverScore [23], Clark et al [2] use SMS and S+WMS to divide the text into multiple sentence vectors or a mixture of sentence vectors and word vectors to perform similarity detection on longer continuous text content. Clark et al [21] have proposed a content-based weighted generative summary evaluation metric.

These metrics can partly solve the quality evaluation problem of summary, but they have the following drawbacks. (1) These methods rely on reference summaries which are costly and labor-intensive, and the evaluation results completely depend on the quality of the reference summary. (2) The information in the source text is greatly wasted. (3) Model-based evaluation methods have a large cost, which is not conducive to practical use. Therefore, the researchers proposed the reference-less evaluation metrics.

2.2 Reference-Less Evaluation Metrics

Model-Free Metrics: Louis et al. [12] first introduce a reference-less summary evaluation method that uses JS divergence to determine whether the word distribution in the machine summary is similar to the word distribution in the source text.

Model-Based Metrics: Chen et al. [1] have proposed a summary evaluation method based on a question answering system. SummaQA [16] uses generated summaries to answer cloze-style questions, and evaluates summary quality by reporting F_1 overlap scores and QA model confidence. SUM-QE [20] uses the BERT [3] to evaluate the quality of summary in five dimensions: grammar, redundancy, clarity of reference, content relevance, and article organization. Kryscinski et al. [7] present a method based on the factual content of the source text, using machine summary to match multiple text fragments extracted from the source text. SUPERT [4] is an evaluation method for reference-less multi-document summaries. LS_Score [19] is an unsupervised evaluation method based on contrastive learning, which combines semantic and linguistic dimensions to evaluate the summary, and uses contrastive learning to optimize the model. QuestEval [15] uses a question answering(QA) system to assess the semantic match between the source text and the summary. Finally, the F1-score between the predicted answer and the real answer is used as the semantic matching degree between the summary and the source text.

Motivation: Although these metrics no longer rely on reference summaries and utilize source text information, they also suffer from the following drawbacks. (1) The noise information in the source text is not well handled, which affects the evaluation results. (2) Using all the semantic information of the source text to match the summary is not only inaccurate but also fails to generate correct matching results. (3) Model-based evaluation methods have a large cost, which is not conducive to practical use. In order to overcome the above problems, we propose a lightweight reference-less method ***SE-tiny*** for summarization quality evaluation.

3 Method

In order to overcome the shortcomings in the above analysis, we propose ***SE-tiny***. We divide the summary evaluation into the following two dimensions: (1) Summary's self-quality dimension. (2) The matching dimension of the summary and the source text. Based on the evaluation dimensions proposed above, ***SE-tiny*** is divided into the following three steps: (1) Summary's self-quality evaluation. (2) Calculate the similarity between the summary features and the key features extracted from the source text. (3) Integrate the scores of step1 and step2. The details are as follows.

3.1 Summary Self-quality

The summary's self-quality is mainly divided into compression ratio, readability, and fluency. The compression ratio is a static metric and can be directly calculated. In order to reduce the model cost while maintaining the evaluation quality, we use the summary sequence S as input, and obtain the probability value of the sequence S through a trained n-gram language model (CNN/DM) to represent the readability and fluency scores. Furthermore, considering that the n-gram model is unfriendly to long summaries, we add a penalty factor.

step1: For summary sequence S, the compression ratio is denoted as "Compress(S)". If "Compress(S)" within [0.75-0.98], continue to the next step, otherwise the quality score of the summary is 0.

step2: We fuse the probability values of the unigram (S_{g1}) and bigram (S_{g2}) outputs of the sequence S as readability and fluency scores (P_s). Adding a penalty factor for text length reduces the unfriendly of n-grams for long summaries.

$$P_s = Min\_max(\alpha \lg S_{g1} + \beta \lg S_{g2} + \eta |S|) \tag{1}$$

Where $|S|$ is the text length, α, η is tuning parameters, $\alpha + \beta = 1$ here. Min_max(*) is a normalization function.

In the current summarization evaluation, only human scoring can be regarded as the “gold standard". We use the Spearman correlation between our method and human scoring (***SE-tiny-DB***) to analyze the value of α, η. The specific analysis results are shown in Fig. 2. We can find that the Spearman correlation is the highest when $\alpha = 0.401, \eta = 0.429$ here.

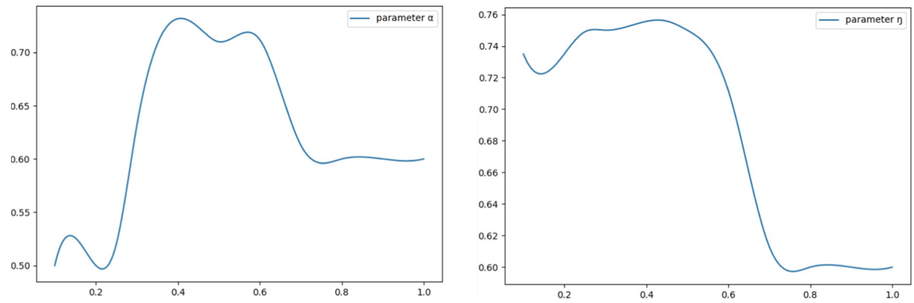

Fig. 2. The effect of parameter α and η on ***SE-tiny***. The abscissa is the α and η value, the ordinate is the Spearman correlation between our method scores and human scores (***SE-tiny-DB***).

3.2 Matching Degree

We use the BERT [3] contextual embedding to represent the summaries and the source texts. In order to conform to the logic when humans evaluate summary, we extract key features from the features of the source text to perform similarity calculations with the features of the summary. This eliminates noise, improves evaluation quality, and reduces computational cost. The specific method is as follows:

Text Preprocessing: The preprocessing part removes stop words, prepositions, and some content-independent words.

Feature Extraction: For the bridge between text and features, we choose a lightweight language model BERT-base [3]. S and D represent summary and source text, respectively. $BERT(S)$ and $BERT(D)$ represent the feature representations of S and D. $BERT(*)$ is obtained through the token-level feature representation by using the BERT-base [3].

Extraction of Key Features: The feature vectors of the source text are $T = [t_1, ..., t_n]$, and the key feature vectors of the source text are $K_i = (k_1, ..., k_w)$.

Stack the m key features as a key content matrix $K = [K_1, ..., K_m]$. We define the approximative token vector $\widetilde{t_i}$ for the token t_i as the optimal linear approximation given by key feature matrix: $\widetilde{t_i} = \widetilde{\alpha}_i K$, where $\widetilde{\alpha}_i = argmin||t_i - \widetilde{\alpha}_i K||_2^2$, $|| * ||$ is the Frobenius norm of a matrix. We use the key feature matrix to approximate the features of the source text and minimize the approximate representation error (E).

$$E = \sum_{i=1}^{n} ||t_i - \widetilde{t_i}||_2^2 \tag{2}$$

$$K = argminE(K) \tag{3}$$

Without loss of generality, we make the key feature vectors $[K_i]_{i=1}^m$ to be orthonormal. Then this optimization problem can be solved by Singular Value Decomposition (SVD). $T = UWV^T$, WV^T is an approximate representation of T. We denote key feature vectors ($[K_i]_{i=1}^m$) of the source text by the first m vectors of V^T. The W as the weight matrix of V^T. This ensures approximate representation error is minimized.

Calculate Spatial Distance: We calculate the cosine distance between the key feature vectors of the source text and the feature vectors of the summary to indicate the matching degree. Map the distance into [0, 1] to get the $Match_{score}$.

We use the Spearman correlation between our method and human scoring (***SE-tiny-DB***) to analyze the value of m, and the specific analysis results are shown in Fig. 3. We can find that the spearman correlation is the highest when $m = 4$.

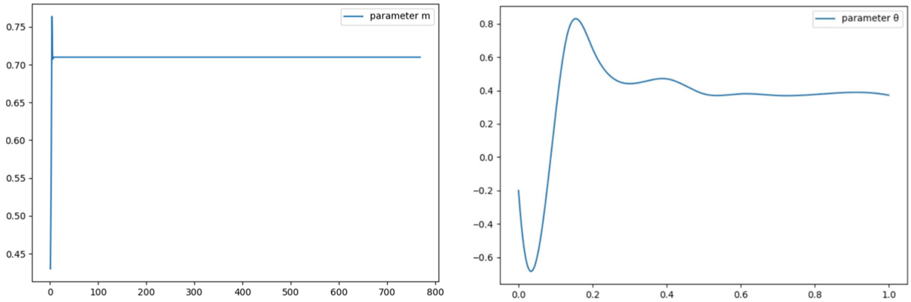

Fig. 3. The effect of parameter m and θ on ***SE-tiny***. The abscissa is the m and θ value, the ordinate is the Spearman correlation between our method scores and human scores (***SE-tiny-DB***).

3.3 SE-Tiny Score

We adopt a linear approach to fuse the scores obtained from the above two dimensions.

$$SE_{tiny} = \theta P_s + (1 - \theta) Match_{score} \tag{4}$$

Where θ is a tuning parameter. We use the Spearman correlation between our method and human scoring (***SE-tiny-DB***) to analyze the value of θ, and the specific analysis results are shown in Fig. 3. We can find that the spearman correlation is the highest when $\theta = 0.13$.

4 Experiment

We verified ***SE-tiny*** from four aspects: correlation, actual effect, cost and cross-datasets transferability. Select the commonly used evaluation metrics ROUGE [9], MoverScore [23], BERTScore [22], LS_Score [19] and the current best performing metric QuestEval [14] as the baselines for evaluating metrics. Select Transformer [18], Presumm [10], Bart [8] and SimCLS [11] as baselines for generating machine summaries. For specific model parameters, please refer to Appendix A.

4.1 Evaluation Dataset

Since there is currently no high-quality dataset for reference-less summary evaluation methods, we constructed a dataset (***SE-tiny-DB***) dedicated to evaluating reference-less summary and invited 6 linguists to score the <summary-document> pairs. The specific scoring standard are as follows.

(1) Summary Self-quality Evaluation:

step1: If the compression ratio of the summary is in the range of [0.75-0.98], the next scoring is performed, otherwise it is directly scored as 0.

step2: We count the frequency of occurrence of disfluency in the summary as f. $f \geq 5$: 0 points; $f = 4$: 1 points; $f = 3$: 2 points; $f = 2$: 3 points; $f = 1$: 4 points; $f = 0$: 5 points. We got the fluency score ($Score_{fluency}$) based on the above standard.

step3: We count the frequency of incomprehensible occurrences in the summary as r. $r = 5$: 0 points; $r = 4$: 1 points; $r = 3$: 2 points; $r = 2$: 3 points; $r = 1$: 4 points; $r = 0$: 5 points. We got the readability score ($Score_{readability}$) based on the above standard.

(2) Evaluation of Matching Degree: The percentage of the summary covering the key content of the source text is denoted as k. $k < 1/5$: 0 points; $1/5 \leq k < 2/5$: 1 point; $2/5 \leq k < 3/5$: 2 points; $3/5 \leq k < 4/5$: 3 points; $4/5 \leq k < 1$: 4 points; $k = 1$: 5 points. We got the matching degree score ($Score_{match}$) based on the above standard.

(3) Dataset Construction Process: Our data source comes from the existing datasets CNN/DM [6] and Newsroom [5]. Specific steps are as follows:

step1: We divided linguistic experts into two groups. The first group of experts scored the data in ***SE-tiny-DB*** according to the scoring standard. The second group of experts checked the scoring results of the first group.

step2: The two groups of experts exchanged and repeated **step1**.

According to the established scoring standard, the summary data scores in CNN/DM [6] and Newsroom [5] are concentrated in the range of [2, 5] points, which is not conducive to our use. In response to this situation, we have processed the summarization data in ***SE-tiny-DB***. The processing methods include: deleting some key points in the summary, rewriting the summary to make it difficult to understand, rewriting the summary to make it less fluent, etc. After these processes and repeating the scoring steps, we get the final dataset, and the statistical results of the dataset are shown in Table 1.

Table 1. Statistics of the distribution of data in ***SE-tiny-DB*** on each score.

	0 points	1 point	2 points	3 points	4 points	5 points
Fluency	178	417	472	500	408	43
Readability	127	305	582	532	402	70
Matching degree	122	308	526	476	486	100

4.2 Correlation

Correlation with Human Metrics. Human metrics are the current “gold standard" for summary quality evaluation. We selected fluency, readability, matching degree and composite score for correlation calculation (Spearman) on ***SE-tiny-DB***. The specific results are shown in Table 2. Compared with other methods, ***SE-tiny*** has the highest correlation with human metrics, which fully reflects the superiority of our method.

Table 2. Correlation (Spearman) evaluation of ***SE-tiny*** and some baseline methods with human scoring. Total represents the composite score of human metrics

	Fluency	Readability	Match	Total
ROUGE-L(F)	0.37	0.34	-0.10	0.28
MoverScore	0.35	0.31	0.28	0.27
BERTScore(F)	0.18	0.23	0.31	0.25
LS_Score	0.42	0.46	0.64	0.53
QuestEval	0.49	0.51	0.70	0.68
SE-tiny	**0.66**	**0.55**	**0.78**	**0.71**

4.3 Scoring Effect

In order to verify the actual scoring effect of ***SE-tiny***, we use our method and some baseline methods to score the machine summaries generated by the text summarization models. The specific scoring results are shown in Table 3. We can find that the evaluation results of ***SE-tiny*** are consistent with the quality of the models.

Table 3. The actual scoring effect of ROUGE, BERTScore (F), and ***SE-tiny*** on the machine summaries generated by the text summarization models.

	ROUGE-1	ROUGE-2	ROUGE-L	BERT(F)	***SE-tiny***
Lead3	40.42	17.62	36.67	75.16	61.42
TextRank	35.23	13.90	31.48	65.71	56.89
Pointer	39.34	17.21	35.23	76.54	63.76
Transformer	39.66	17.19	36.66	87.68	68.47
BertSumAbs	41.72	19.39	38.76	88.26	69.18
BertSumExtAbs	43.23	20.24	39.63	88.22	70.91
Bart	44.39	21.21	41.28	88.33	69.16
SimCLS	46.67	22.15	43.54	89.24	71.85

4.4 Model Cost

A good evaluation model needs to have less cost, we calculate the cost of various methods to complete the evaluation of 10K <summary-document> pairs. The specific results are shown in Table 4. Compared with model-based evaluation methods, ***SE-tiny*** saves time and has less space cost.

Table 4. Model cost comparison

	ROUGE	BERTScore	LS_Score	***SE-tiny***
Time (s)	<300	8,418	9,761	**4,181**
Space (MB)	<10	4,280	3,525	**716**

4.5 Cross-Datasets Transferability

A good evaluation method should also have good transferability. In order to verify this feature, we train SE-tinyon two datasets CNN/DM [6] and NEWSROOM [5], and then evaluate the data quality of SE-tiny-DB and calculate the correlation (Spearman) with human scoring. SE-tiny-DB in 4.1 above contains part of the data in CNN/DM [6] and NEWSROOM [5] after scoring. The specific experimental results are shown in Table 4. Both SE-tiny-CNN/DM and SE-tiny-CNN/DM* are the correlation score between CNN/DM [6] scoring results and human scoring, SE-tiny-CNN/DM is trained using CNN/DM [6] data, and SE-tiny-CNN/DM* is trained using NEWSROOM [5]. Both SE-tiny-NEWSROOM and SE-tiny-NEWSROOM* are the correlation score between NEWSROOM [5] scoring results and human scoring, SE-tiny-NEWSROOM is trained using NEWSROOM [5] data, and SE-tiny-NEWSROOM* is trained using CNN/DM

[6]. We can find that the scoring results in other datasets are similar to the scoring results trained in this dataset. This fully demonstrates the transferability of our method.

Table 5. Cross-datasets transferability

	Fluency	Readability	Match	Total
SE-tiny-CNN/DM*	0.63	0.64	0.75	0.74
SE-tiny-CNN/DM	0.67	0.58	0.78	0.76
SE-tiny-NEWSROOM*	0.53	0.54	0.61	0.56
SE-tiny-NEWSROOM	0.57	0.60	0.68	0.62

5 Conclusion

SE-tiny is a lightweight reference-less summary quality evaluation method based on key feature extraction. By matching the summary features with the key features of the source text, this evaluation method conforms to human evaluation logic, eliminates noise, improves evaluation quality, and reduces the cost. Experimental results show that our method outperforms existing model-based summarization evaluation methods in both performance and cost. At the same time, our method does not rely on reference summaries and can be generalized to evaluation on summary datasets.

6 Limitation and Future Work

Our method is a general approach, effective for commonly used summarization data, but lacks specificity for domain-specific summarization data evaluation. In order to overcome this shortcoming, more algorithms for specific domains need to be proposed in the future.

With the improvement of computing power, more and more large models have been proposed (Chat-gpt, GPT-3 etc.), but when the model develops to a certain scale, how to reduce the cost of the model while ensuring the quality of summarization evaluation is also an area for future development. In addition to the pruning algorithm, reducing overhead directly from the model architecture is also a future trend. In the future, we hope to propose more lightweight and efficient summarization evaluation algorithms.

References

1. Chen, P., Wu, F., Wang, T., Ding, W.: A semantic qa-based approach for text summarization evaluation. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI 2018), the 30th innovative Applications of Artificial Intelligence (IAAI 2018), and the 8thAAAI Symposium on Educational Advances in Artificial Intelligence (EAAI 2018), New Orleans, Louisiana, USA, 2–7 February 2018, pp. 4800–4807. AAAI Press (2018). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16115
2. Clark, E., Celikyilmaz, A., Smith, N.A.: Sentence mover's similarity: automatic evaluation for multi-sentence texts. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 2–7 July - 2 August 2019, Volume 1: Long Papers, pp. 2748–2760. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/p19-1264
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423
4. Gao, Y., Zhao, W., Eger, S.: SUPERT: towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020, pp. 1347–1354. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.124
5. Grusky, M., Naaman, M., Artzi, Y.: Newsroom: a dataset of 1.3 million summaries with diverse extractive strategies. In: Walker, M.A., Ji, H., Stent, A. (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, 1–6 June 2018, Volume 1 (Long Papers), pp. 708–719. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/n18-1065
6. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, pp. 7–12, 2015. Montreal, Quebec, Canada, pp. 1693–1701 (Dec 2015), https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html
7. Kryscinski, W., McCann, B., Xiong, C., Socher, R.: Evaluating the factual consistency of abstractive text summarization. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020, pp. 9332–9346. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.750

8. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020, pp. 7871–7880. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.703, https://doi.org/10.18653/v1/2020.acl-main.703
9. Lin, C., Hovy, E.H.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Hearst, M.A., Ostendorf, M. (eds.) Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, 27 May - 1 June 2003. The Association for Computational Linguistics (2003). https://aclanthology.org/N03-1020/
10. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019, pp. 3728–3738. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1387
11. Liu, Y., Liu, P.: Simcls: A simple framework for contrastive learning of abstractive summarization. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, 1–6 August 2021, pp. 1065–1072. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.acl-short.135
12. Louis, A., Nenkova, A.: Automatically evaluating content selection in summarization without human models. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6–7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 306–314. ACL (2009), https://aclanthology.org/D09-1032/
13. Ng, J., Abrecht, V.: Better summarization evaluation with word embeddings for ROUGE. In: Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., Marton, Y. (eds.) Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015, pp. 1925–1930. The Association for Computational Linguistics (2015). https://doi.org/10.18653/v1/d15-1222
14. Rebuffel, C., et al.: Data-questeval: A referenceless metric for data-to-text semantic evaluation. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7–11 November 2021, pp. 8029–8036. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.emnlp-main.633
15. Scialom, T., et al.: Questeval: summarization asks for fact-based evaluation. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7–11 November 2021, pp. 6594–6604. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.emnlp-main.529

16. Scialom, T., Lamprier, S., Piwowarski, B., Staiano, J.: Answers unite! unsupervised metrics for reinforced summarization models. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019, pp. 3244–3254. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1320
17. ShafieiBavani, E., Ebrahimi, M., Wong, R.K., Chen, F.: A graph-theoretic summary evaluation for rouge. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October - 4 November 2018, pp. 762–767. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/d18-1085
18. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017 (December), pp. 4–9, 2017, Long Beach, CA, USA, pp. 5998–6008 (2017). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
19. Wu, H., Ma, T., Wu, L., Manyumwa, T., Ji, S.: Unsupervised reference-free summary quality evaluation via contrastive learning. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020, pp. 3612–3621. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.294
20. Xenouleas, S., Malakasiotis, P., Apidianaki, M., Androutsopoulos, I.: Sumqe: a bert-based summary quality estimation model. CoRR abs/ arXiv: 1909.00578 (2019)
21. Xu, X., Dusek, O., Li, J., Rieser, V., Konstas, I.: Fact-based content weighting for evaluating abstractive summarisation. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020, pp. 5071–5081. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.455
22. Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K.Q., Artzi, Y.: Bertscore: evaluating text generation with bert. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=SkeHuCVFDr
23. Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C.M., Eger, S.: Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019, pp. 563–578. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1053