A Close Examination of Factual Correctness Evaluation in Abstractive Summarization

Stanford CS224N Custom Project

Yuhui Zhang *
Department of Computer Science
Stanford University
yuhuiz@stanford.edu

Abstract

Generating fabricated facts has been a long-standing problem of abstractive summarization models, and has significantly limited their applicability in practice. Previous works about improving factual correctness only rely on human evaluations, which weakens the transparency and reproducibility. In this work, we aim to examine how to evaluate factual correctness. We start with a human study to thoroughly understand what affects factual correctness evaluations, and we further assess whether current automatic factual evaluation metrics are able to capture factual errors. Our experiments demonstrate that the attributes of models and datasets can drastically affect the evaluation of factual correctness, and how to design an accurate, model- and data-agnostic evaluation metrics still remains a challenge to the NLP community.

1 Introduction

Abstractive summarization aims to distill essential information from the source document but not necessarily preserve original expressions [1]. While recent progress on abstractive summarization can successfully produce fluent and informative summaries [2, 3, 4], they are not optimized for a critical aspect — factual correctness. According to Kryscinski et al. [5], around 30% of summaries generated by neural abstractive models contain factual errors, which severely limits further applications of summarization systems. Table 1 shows an example of the generated summary with factual errors.

Doc	Kevin Patrick Dawes , 33 , was abducted in 2012 as he entered Syria . US officials told the Washington Post that the Syrian government never acknowledged detaining Mr Dawes , but they believe the government or an affiliated group was holding him . The State Department said Mr Dawes was turned over to Russian authorities ..... He appeared blindfolded in a video a month after his abduction but has not been heard from since .
Gen	The US president of the United Arab Emirates (State) has been abducted in Syria , US officials say .

Table 1: Example of the generated summary with factual errors. Summaries are generated by PGC [2] on XSum [6] dataset. We highlight false facts in red and true facts in blue. In the generated summary, neither *US president* nor *United Arab Emirates (State)* is mentioned in the document.

While some recent works have explored several different ways to improve factual correctness of generated summaries via fact-aware decoding [7], reranking [8], or reinforcement learning [9], only relying on human evaluations weakens their transparencies and reproducibility. As it is important guidance to design faithful models, how to accurately detect these factual errors and reliably evaluate factual consistencies become increasingly non-negligible. However, existing commonly-used summarization

*Key Information to include: External mentor: Yuhao Zhang, Christopher D. Manning

evaluation metrics such as ROUGE [10] and BERTScore [11] simply measure word overlaps between generated summary and human reference summary, and whether they can reflect factual alignments remains unstudied. Only limited works have focused on factual correctness evaluations [12, 13, 14].

In this work, we aim to thoroughly examine how to evaluate factual correctness, and what properties should satisfy for factual evaluation metrics. We start with a human study to thoroughly understand how humans judge factual correctness, which has never been addressed in previous works to the best of our knowledge. We demonstrate that attributes of summarization models and datasets can drastically affect the factual correctness of generated summaries. We find that, on CNNDM [15] dataset, allowing the model to directly copy words from source document [2] significantly reduces factual error rates, and factual consistencies can be simply evaluated by word overlaps [10]. However, these conclusions no longer hold when moving to XSum [6] dataset, where summaries are more abstractive and highly-paraphrased. We also find that while the reference summary is often used as the gold standard for evaluating summary quality, it can not provide enough information for checking factual consistencies.

We further assess whether current automatic factual evaluation metrics can capture factual errors. We use **FactCCX** [13] as our benchmark, a state-of-the-art BERT-based [16] factual consistency checking model that can identify conflicts between documents and claims. As there is no existing training data for fact checking, FactCCX is trained in a weakly-supervised fashion, where training data are automatically generated through several rule-based semantical transformations (e.g., back-translation, negation, entity swap). Our experiments demonstrate that FactCCX is intrinsically fragile. It only works well on easy examples and examples which mostly require direct copying from the context, and can not generalize well to abstractive summarization datasets. We also observe that FactCCX may overfit to bias in training data (e.g., translationese).

Our work reveals the difficulties of evaluating factual correctness, and suggests that designing an accurate, model- and data-agnostic evaluation metrics remains a challenge for the NLP community. Nonetheless, we hope this work can raise attention and shed light on future research about factual correctness evaluations.

2 Human Evaluation of Factual Correctness

Understanding how humans evaluate factual correctness can provide important guidance to correctly design factual correctness evaluation metrics. In this section, we manually annotate the correctness of summaries generated from different systems and datasets and perform a thorough analysis.

2.1 Data

We focus on two summarization benchmark datasets: **CNNDM** [15] and **XSum** [6]. These two datasets are both curated from online articles from news providers (i.e., CNNDM from CNN and DailyMail, XSum from BBC). All these news providers supplement their documents with a single (for XSum) or several (for CNNDM) introductory sentences summarizing key information contained in the document.

While CNNDM is widely used for abstractive summarization, the high ratio of word overlaps between the source document and reference summary favors extractive strategies too much. For example, even the lead-3 baseline (i.e., simply using the lead three sentences as the generated summary) is on par with state-of-the-art neural abstractive models under ROUGE evaluations [2].

As evidenced by the significantly lower **extractive oracle ROUGE**, the upper-bound performance for summary generated by only extracting words from the source document, reference summaries in the XSum dataset are highly-paraphrased (Table 2). The abstractive nature of reference summaries challenges neural models' capacity to understand and paraphrase languages.

2.2 Model

Summaries are generated by three systems on CNNDM and XSum dataset. PGC [2] enables the model to directly copy words from the source document, addressing the out-of-vocabulary issue. An additional coverage mechanism also reduces repetitive words in generated summaries. FAS [17] first selects salient sentences from the source document and then compresses and paraphrases them to

Dataset	Data Split			Average Length		Extractive Oracle		
	Train	Dev	Test	Source	Reference	R-1	R-2	R-L
CNNDM	287,227	13,368	11,490	664	54	54.67	30.36	50.80
XSum	204,045	11,332	11,334	431	23	29.79	8.81	22.65

Table 2: Statistics of CNNDM and XSum dataset. Reference summaries in the XSum dataset are highly-paraphrased, as evidenced by the significantly lower extractive oracle ROUGE.

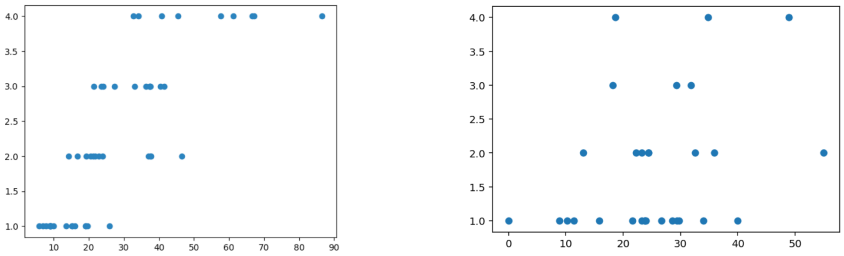

Figure 1: Correlations of ROUGE with human-annotated factual score. Left: CNNDM. Right: XSum. X-axis: ROUGE-L Score. Y-axis: Annotated Factual Score.

generate a concise overall summary. BUS [4] adds a content selector to first determine phrases in a source document that should be part of the summary. Readers can refer to original papers for more details.

2.3 Experiment

While there are many evaluation metrics to measure the overall quality of the generated summary, whether they can reflect factual consistencies between generated summaries and reference summaries still remains a question. We randomly sample 50 summaries generated by PGC on CNNDM and XSum dataset, respectively, and measure factual alignments with 5-scale likert score (e.g., no factual alignment, only minor alignment, somewhat alignment, only minor misalignment, completely aligned) by comparing generated summaries with human reference summaries. Two annotators conduct this experiment.

We compute the correlations of three evaluation metrics with humans annotations: ROUGE, BERTScore, and FACTScore. ROUGE [10] is the standard evaluation metric for summarization, which measures n-gram overlap between generated summaries and reference summaries. Instead of only considering n-gram hard-match, BERTScore [11] measures word soft-match using contextualized word embedding generated by BERT [16]. While ROUGE and BERTScore provide word-level evaluations for summarization, FACTScore [14] evaluates summaries at fact-level. Facts are first extracted from generated summaries and reference summaries using open information extraction systems and then encoded to vectors by sentence encoder. The final score is computed by averaging cosine similarities between each fact pair. Note that *FACTScore* is an evaluation metric, while human-annotated *facutal score* is the 5-scale likert score. Table 3 summarizes the correlations of different evaluation metrics with human-annotated factual score and inter-annotator agreements. Figure 1 shows the correlations of ROUGE with human-annotated factual scores.

Dataset	Metric Correlation			Inter-Annotator Agreement
	ROUGE	BERTScore	FACTScore	
CNNDM	0.81	0.84	0.69	0.91
XSum	0.36	0.32	0.30	0.60

Table 3: Correlations of different evaluation metrics with annotated factual alignment score and inter-annotator agreements. Results on CNNDM and XSum dataset vary significantly.

We further target at finding factual errors by comparing 1) generated summaries with reference summaries; 2) generated summaries with source documents; 3) reference summaries with source

documents, respectively. Note that human-annotated factual score aims to measure that to what extent facts conveyed from the generated summary align with facts conveyed from the reference summary, and whether current evaluation metrics can capture this alignment, while factual errors particularly refer to fabricated facts that are not entailed from the given context. Still, two annotators conduct these experiments. Table 4 shows the proportions of summaries with factual errors of different neural abstractive summarization models on CNNDM and XSum datasets.

Dataset	System	System Performance			Factual Error Rate		
		R-1	R-2	R-L	GEN-SRC	GEN-REF	REF-SRC
CNNDM	PGC	39.49	17.24	36.35	8%*	6%	0%
	FAS	40.88	17.80	38.53	26%*	-	
	BUS	41.52	18.76	38.60	25%*	-	
XSum	PGC	26.87	7.93	21.38	100%	23%	30%

Table 4: System performance and factual error rates of summaries generated by different systems on CNNDM and XSum dataset. * indicates results reported in [8].

2.4 Analysis

Evaluation metrics fail to capture factual alignments on abstractive datasets. On the CNNDM dataset, the ROUGE correlates with the annotated factual score surprisingly well (0.81), and the inter-annotator agreement is pretty high (0.91). We observe different results on the XSum dataset with the same experimental settings (0.36 and 0.60 shown in Table 3). This can be interpreted as the high word overlaps between reference summaries and source documents for the CNNDM dataset, where models learn to copy too much from context, and measuring word overlaps can capture factual alignments.

ROUGE reflects instance-level factual alignments better than system-level. While the correlation of ROUGE and annotated factual score on CNNDM is 0.81, demonstrating that ROUGE reflects instance-level factual alignment well, ROUGE can not distinguish which system generally produces less factual errors. As shown in Table 4, PGC generates the least factual errors (8%) but receives the least ROUGE (36.35), while BUS receives the highest ROUGE (38.60) but generates considerably more errors (25%).

Factual error rates vary significantly among systems. From Table 4, we find that summaries generated by PGC [2] contain remarkably less factual errors (8%) on the CNNDM dataset [15], even compared to models that were later considered significantly better (BUS [4] (25%) and FAS [17]). This may be interpreted as the copy mechanism allows PGC to copy too much from the source document and degrade to extractive models.

Factual error rates vary drastically among datasets. PGC is trained on CNNDM and XSum dataset with almost the same settings. However, nearly all summaries generated by the PGC model contain factual errors when comparing generated summaries with source documents on the XSum dataset, which is drastically different from 8% on the CNNDM dataset. The highly-abstractive nature of the XSum dataset challenges the model's capacity to understand the language and learn to paraphrase.

Factual consistency checking should compare generated summaries with source documents instead of the reference summaries. While the reference summary is often used as the gold standard for evaluating summary quality, it can not provide enough information for checking factual consistencies. The gap of factual error rates become much larger when shifting from CNNDM to XSum (100% vs. 30% shown in Table 4), which can be interpreted as much less information can be inferred from references as evidenced by the significantly lower ROUGE that the model can achieve.

Even reference summaries may contain factual misalignments. When comparing reference summaries with source documents, we find around 30% reference summaries in the XSum dataset

contain facts that are not entailed in source documents. This raises concerns about the quality of the XSum dataset.

3 Automatic Fact Checking

In this section, we explore the limitations of FactCCX [13], the only available automatic fact checking tool.

3.1 Method

We utilize pretrained FactCCX [13] for fact checking. The overview of FactCCX is shown in Figure 2[2]. Based on BERT [16], FactCCX consumes a source document and a claim, and concatenates them as one sentence for factual correctness classification (i.e., <*CLS*> <*SOURCE*> <*SEP*> <*CLAIM*> <*SEP*>, where <*CLS*> and <*SEP*> are special tokens for BERT). Besides the binary label output from the model (i.e., *CORRECT* and *INCORRECT*), FactCCX also finds supporting / conflicted spans from source document and claim. As BERT only accepts sentences within 512 tokens, the source document will be truncated if the concatenated sentence exceeds the limit. This innate deficiency significantly weakens the reliability of FactCCX, as a fair amount of sentences in both CNNDM and XSum dataset exceeds this limit as shown in Table 2.

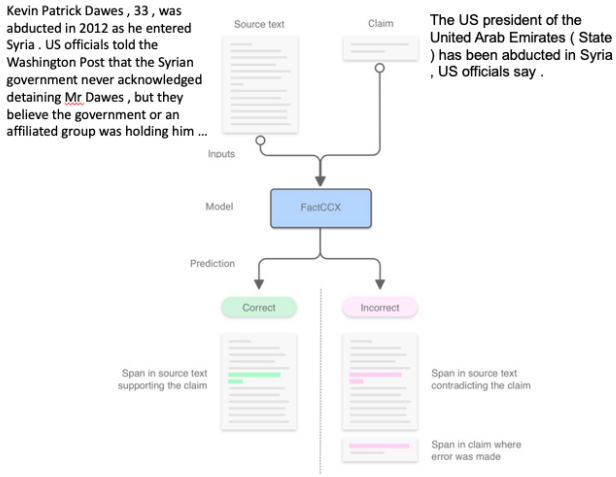

Figure 2: Overview of FactCCX. FactCCX predicts CORRECT for this example (should be INCORRECT).

As there is no supervised data to directly train a fact checking model[3], Kryscinski et al. proposed to train FactCCX using the weakly supervised learning strategy, where the training data are automatically generated from several rule-based semantically-variant or semantically-invariant transformations.

Specifically, one claim is randomly sampled from sentences in the source document, and its back-translated form (i.e., translate the claim to an intermediate language and then translate back to its original language) and itself are used as positive examples, as we suppose back-translation preserve semantics but paraphrase the sentence. These two positive examples are further processed by four types of rule-based transformations (i.e., negation, entity swap, number swap, pronoun swap) to generate negative examples, as sentence semantics are no longer preserved after these transformations. For each claim randomly sampled from the source document, at most 10 document-claim pairs will be generated (i.e., 2 positive examples and 8 negative examples if verb / entity / number / pronoun all can be found in the claim). These pairs generated from the CNNDM dataset are further split to train

[2]Credit: https://github.com/salesforce/factCC

[3]While there are a large amount of training data for natural language inference (NLI) [18, 19], for fact checking, the source is a whole document and is much longer than the claim, which makes model trained on NLI data does not generalize well [8].

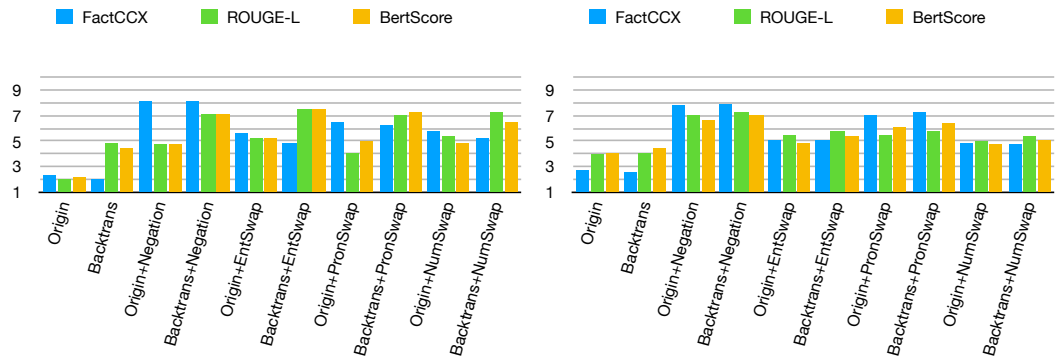

Figure 3: Average ranks of different claims based on FactCCX, ROUGE and BERTScore. As original claim (*Origin*) and back-translated claim (*Backtrans*) are semantically-invariant, they are expected to rank at the top 1. Left: CNNDM. Right: XSum. X-axis: Category. Y-axis: Average rank (1-10).

and evaluate FactCCX. As claims are sampled from source documents and the data are generated by rules, we name this dataset as *Generated Doc-Doc CNNDM*.

3.2 Data

As we focus on exploring the strengths and limitations of FactCCX, we directly use the model pretrained on *Generated Doc-Doc CNNDM* for further evaluations. Besides *Generated Doc-Doc CNNDM*, Kryscinski et al. [13] also provided a small human-annotated dataset, where each claim is the real output from a list of state-of-the-art summarization models and label is annotated by humans. We name it as *Annotated Doc-Gen CNNDM*.

Admittedly, *Annotated Doc-Gen CNNDM* is the best testbed for FactCCX, as claims are the real outputs from summarization models and labels are annotated by humans. However, we observe that the extractive oracle ROUGE on *Annotated Doc-Gen CNNDM* is as high as ROUGE on *Generated Doc-Doc CNNDM* (Table 5). While it is easy to understand the extremely high extractive oracle ROUGE on *Generated Doc-Doc CNNDM* as claims are directly sampled from sentences in the source documents, the comparably high ROUGE on *Annotated Doc-Gen CNNDM* highlights that models trained on CNNDM dataset learn to copy too much from the source. We further prove our hypothesis by manually inspecting examples in *Annotated Doc-Gen CNNDM*. This is consistent with the conclusion that CNNDM favors extractive strategies too much.

From the human evaluation of factual correctness, we find no factual errors can be found from reference summaries in the CNNDM dataset (Table 4), so claims sampled from sentences in the reference summaries can also be treated as positive examples. We make a minor modification for data generation strategy by sampling claims from reference summaries instead of source documents. In this way, we generated another dataset and name it as *Generated Doc-Ref CNNDM*. With the same settings on the XSum dataset, *Generated Doc-Ref XSum* is also generated for comparison, though factual errors can be found from reference summaries in XSum dataset.

3.3 Experiment

We use the FactCCX pretrained on *Generated Doc-Doc CNNDM* training set and evaluate it on these 4 test sets: *Generated Doc-Doc CNNDM*, *Annotated Doc-Gen CNNDM*, *Generated Doc-Ref CNNDM*, and *Generated Doc-Ref XSum*. We report accuracy (ACC), balanced accuracy (BACC), precision (P), recall (R), and F-measure (F1) in Table 5. We use BACC as the performance indicator to address data imbalances, which simply averages the accuracy on positive examples and the accuracy on negative examples. Note that for a random guess baseline (i.e., randomly label CORRECT / INCORRECT), BACC should be equal to 50.

We further collect all the examples with 10 document-claim pairs from *Generated Doc-Ref CNNDM* and *Generated Doc-Ref XSum*, and use the probability of correctness predicted by FactCCX to rank these 10 claims. We also rank them based on extractive oracle ROUGE and BERTScores for comparison. We report the average ranking for each category in Figure 3.

Dataset	Metrics					Ext Oracle		
	ACC	BACC	P	R	F1	R-L	R-2	R-1
Generated Doc-Doc CNNDM	97.43	97.42	97.44	97.42	97.42	75.38	63.71	77.45
Annotated Doc-Gen CNNDM	86.48	72.88	69.71	72.88	71.09	72.69	69.00	73.23
Generated Doc-Ref CNNDM	79.28	73.22	81.81	73.22	74.79	39.71	26.36	43.81
Generated Doc-Ref XSum	77.47	66.43	76.86	66.43	68.10	22.65	8.81	29.79

Table 5: FactCCX performance on fact checking datasets as well as extractive oracle ROUGE for the datasets.

3.4 Analysis

FactCCX only works well on easy examples. We categorize document-claim pairs from *Annotated Doc-Gen CNNDM* based on their extractive oracle ROUGE, which measures to what extent n-grams in claims overlap with n-grams in documents. From Figure 4, we observe consistent and significant improvements as extractive oracle ROUGE increases (ROUGE-1, ROUGE-2 and ROUGE-L). While it achieves over 90% accuracy on examples with 0.8-1.0 ROUGE, its performance drops to 70% on examples with 0.0-0.4 ROUGE. Note that a random-guess baseline can achieve 50% accuracy. Therefore, we argue that FactCCX only works well on easy examples, and actually identifying factual errors for these easy examples becomes trivial — the high n-gram overlaps make it extremely easy to directly retrieve the claim from the document and check whether there are factual errors via word-by-word comparisons.

FactCCX does not generalize well to abstractive summaries. While FactCCX achieves 97.42% balanced accuracy (BACC) on *Generated Doc-Doc CNNDM*, its performance significantly drops when evaluating on *Generated Doc-Ref CNNDM* (73.22% BACC) and *Generated Doc-Ref XSum* (66.43% BACC) (Table 5). Again note a random-guess baseline can achieve 50% accuracy. The reason is clear to us as n-gram overlaps become drastically lower (evidenced by extractive oracle ROUGE). It further strengthens our argument — FactCCX only learns to pick the sentence that is most similar to the claim and compare it word by word. When there is no enough signal for FactCCX to retrieve the sentence from the document, it fails to perform fact checking.

FactCCX does not generalize well to real data. While extractive oracle ROUGE on *Annotated Doc-Gen CNNDM* is comparable to ROUGE on *Generated Doc-Doc CNNDM*, enabling FactCCX to easily retrieve the sentence that best resembles the claim, there is a large discrepancy between the balanced accuracy (97.42% vs 72.88% shown in Table 5). This indicates error checking learned from weakly supervised learning can not generalize well to real scenarios, where errors might be much more complex than errors induced through several rule-based transformations.

FactCCX may overfit to bias in training data, such as translationese. We rank all the examples with 10 document-claim pairs based on the probability of correctness predicted by FactCCX , as well as ROUGE and BERTScore. As original claim (*Origin*) and back-translated claim (*Backtrans*) are semantically-invariant, they are expected to rank at the top. As shown in Figure 3, the average ranks for *Origin* and *Backtrans* are around 2 and 3 on CNNDM and XSum dataset, respectively. Though outperforming ROUGE and BERTScore, FactCCX still fails to rank them as the top for many cases. However, we observe that back-translated claims always receive lower ranks than original claims, no matter semantically-invariant or semantically-variant examples (e.g., compare *Origin* and *Backtrans*, *Origin+EntSwap* and *Backtrans+EntSwap*, etc.). This may indicate FactCCX overfit to bias in training data, such as translationese [20].

Other issues of FactCCX. We also find other issues with FactCCX. 1) Probabilities of correctness predicted by FactCCX are strongly centered at 0 and 1, where extractive oracle ROUGE and BERTScore are normally distributed (Figure 5). This raises concerns about using FactCCX to rank examples. 2) BERT only accepts sentences within 512 tokens; many cases are truncated, and key information is lost for fact checking. 3) FactCCX relies on a simple binary classification mechanism to identify supporting / conflicted spans, which does not work for almost all cases.

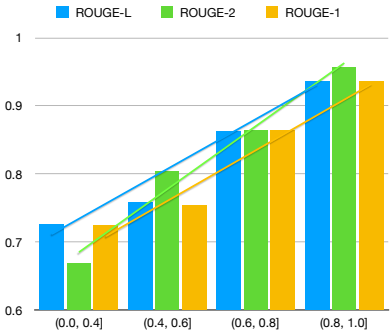

Figure 4: Correlations of accuracy with ROUGE. X-axis: Extractive oracle ROUGE. Y-axis: Accuracy (0-1).

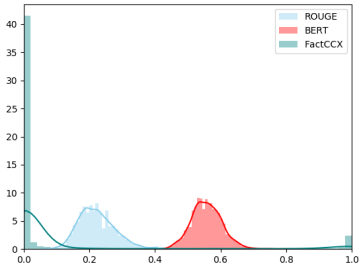

Figure 5: Distributions of probabilities of correctness predicted by FactCCX, extractive oracle ROUGE and BERTScore.

4 Related Work

Neural Summarization Models Two main approaches are used to generate summaries: extractive summarization and abstractive summarization. While extractive summarization simply copies words from the source document, abstractive summarization aims to paraphrase the source document and can generate new phrases. Rush et al. first proposed to use sequence-to-sequence (seq2seq) network for abstractive summarization [1]. Based on seq2seq, many works improve the quality of generated summaries via different mechanisms. See et al. proposed to allow the model to directly copy words from the source document [2]. Chen et al. proposed to first select salient sentences and then compress to a concise summary [17]. Gehrmann et al. add a content selector to determine phrases in a source document that should be part of the summary [4]. Recently, pretraining-based summary generation makes significant progress, but it requires significant computational resources [21].

Evaluating Factual Correctness in Summarization Goodrich et al. first proposed to evaluate the factual accuracy by comparing facts extracted from different information extraction systems [12]. Kryscinski et al. proposed to check factual consistencies in the generated summaries using a BERT-based fact verification model trained via weakly-supervised learning [13]. Zhang et al. proposed to evaluate factual correctness by comparing facts extracted from generated summaries and reference summaries using open information extraction systems [14].

Improving Factual Correctness in Summarization Cao et al. first proposed to improve the faithfulness of abstractive summarization via fact-aware decoding, where the decoder attends to fact triples extracted from the source document using open information extraction systems [7]. Falke et al. proposed to improve the factual correctness by reranking generated summaries based on entailment scores predicted by natural language inference systems [8]. Zhang et al. proposed to use reinforcement learning on fact accuracy to improve the factual correctness of summarizing radiology reports [9]. Zhu et al. proposed to build a knowledge graph from source document and integrate it into the summary generation process via neural graph computation [22].

5 Conclusion

Factual correctness is an important but missing aspect for evaluating abstractive summarization. Previous works mostly only evaluate factual correctness by humans, which weakens the transparency and reproducibility of their works. In this work, we first analyze what affects factual correctness through human studies. We find factual error rates and relevant evaluation metrics are significantly affected by attributes of summarization models and datasets. We further reveal the weaknesses of current automatic factual evaluation metrics, FactCCX, which is far from accurately identifying factual errors. We hope this work can inspire future works about designing factual evaluation metrics.

For future work, we plan to re-train FactCCX on our generated datasets and re-evaluate its performance. Moreover, we will evaluate the state-of-the-art summarization models pretrained on a large corpus once the pretrained models are released.

References

[1] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, 2015.

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

[3] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018.

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

[5] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. 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 540–551, 2019.

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

[7] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

[9] Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D Manning, and Curtis P Langlotz. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. *arXiv preprint arXiv:1911.02541*, 2019.

[10] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

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

[12] Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 166–175, 2019.

[13] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*, 2019.

[14] Yuhui Zhang, Yuhao Zhang, and Christopher D Manning. Evaluating the factual correctness for abstractive summarization. *CS230 Project*, 2019.

[15] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

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

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

[18] Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 632–642. Association for Computational Linguistics (ACL), 2015.

[19] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, 2018.

[20] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. 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 107–112, 2018.

[21] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*, 2019.

[22] Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. Boosting factual correctness of abstractive summarization with knowledge graph. *arXiv preprint arXiv:2003.08612*, 2020.