Evaluating the Factual Correctness for Abstractive Summarization

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

1 Introduction

Summarization aims to distill essential information from the source text and has been widely applied to headline generation, lawsuit abstraction, biomedical and clinical text summarization. There are two main approaches for summarization: extractive summarization and abstractive summarization. While extractive summarization directly copies words and sentences from the source text, abstractive summarization can paraphrase the source text, leading to more flexible and compressed summaries.

Most works about abstractive summarization aim to improve the ROUGE score (Lin, 2004) — a commonly-used metric for measuring the n-gram overlap between generated summaries and reference summaries. However, evaluating summarization by only measuring n-gram similarity is not perfect and convincing. An important but missing aspect for evaluating abstractive summarization is factual correctness. According to Kryściński et al. (2019a), around **30%** of summaries generated by abstractive models contain factual inconsistencies. This is a critical issue for further applications of abstractive summarization. Table 1 shows an example of factual incorrectness.

In this work, we propose **factual score** — a new evaluation metric to evaluate the factual correctness for abstractive summarization. We first generate summaries using four state-of-the-art summarization models (Seq2seq (Bahdanau et al., 2015), Pointer-Generator (See et al., 2017), ML (Paulus et al., 2018), ML+RL (Paulus et al., 2018)) on widely-used CNN/DM dataset (Hermann et al., 2015). Then, we adopt open information extraction (OpenIE) methods to extract facts from generated summaries and reference summaries. Finally, we use sentence encoder to generate fact embeddings and compute factual score by averaging cosine-similarity of each fact pair.

We further explore the sensitivity of the factual score to factual inconsistencies by manually generating false examples with five semantically variant transformations. Our results demonstrate that the factual score has the highest sensitivity to factual inconsistencies compared with other evaluation metrics like ROUGE score and BERT score (Zhang et al., 2019). Experiments also show that the factual score is highly correlated with ROUGE score and BERT score. Even though our experiments are far from exhaustive, we hope our work could shed light on the evaluations of factual correctness for abstractive summarization.

Source Text	... jacob mincer , a pioneer in labor economics who was the first to quantify the payoff from education and training , died sunday at his home in manhattan . he was 84 . the cause was complications of parkinson 's disease ...
Reference Summary	jacob mincer , pioneer in labor economics , dies at age 84 .
Reference Facts	(jacob mincer; dies; at age 84)
Generated Summary	jacob mincer , pioneer in labor economics who was first to quantify payoff from education and training , died june .
Generated Facts	(labor economics; [to] quantify; payoff from education and training) (jacob mincer; died; june)

Table 1: Example of factual incorrectness from generated summary. Word colored in red is false.

CS230: Deep Learning, Fall 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Method

2.1 Summary Generation

Given a long sequence of tokens $\mathbf{x} = (x_1, x_2, ..., x_m)$, we want to generate a short sequence of tokens $\mathbf{y} = (y_1, y_2, ..., y_n)$, which condenses the information from source text (usually $n \ll m$). In this work, we use 4 strongly-performed abstractive summarization models to generate summaries.

Seq2seq Sequence to sequence (seq2seq) model with attention (Bahdanau et al., 2015) consists of an encoder and a decoder, which are usually implemented using variants of RNNs (LSTM, GRU). The encoder generates a contextualized representation h_m for the source sequence, which is used to initialize the decoder state s_0. At decoding step t, we compute the context vector c_t based on attention a_t. This context vector is then combined with decoder state s_t to generate the probability distribution of next token over vocabulary $\hat{y}_t$. We optimize negative log-likelihood loss $\mathcal{L}_{\text{nll}}$ during training. [1]

$$
\begin{aligned}
h_t &= \text{RNN}(h_{t-1}, x_t) \\
s_t &= \text{RNN}(s_{t-1}, y_t) \\
a_t^i &= \frac{\exp(s_t \cdot h_i)}{\sum_j \exp(s_t \cdot h_j)} \\
c_t &= \sum_i a_t^i h_i \\
\hat{y}_t &= \text{MLP}([s_t; c_t]) \\
\mathcal{L}_{\text{nll}} &= -\sum_{t=1}^{n} \log \hat{y}_t
\end{aligned}
$$

Pointer-Generator In order to increase the token correctness and solve the out-of-vocabulary problem, See et al. (2017) enables the model to directly copy tokens from the source text. Based on seq2seq with attention, they introduce a gate p_{gen} between 0 and 1 to control the model between generating tokens and copying tokens. The updated probability distribution $\hat{y}_t'$ combines the generating probability and copying probability (estimated by attention). Also, we add an auxiliary coverage loss $\mathcal{L}_{\text{cov}}$ to alleviate the repetitive n-gram generated by neural models.

$$
\begin{aligned}
p_{\text{gen}} &= \sigma(\text{MLP}([s_t; c_t])) \\
\hat{y}_t^{'} &= p_{\text{gen}} \hat{y}_t + (1 - p_{\text{gen}}) a_t \\
\mathcal{L}_{\text{cov}} &= -n + \sum_{i=1}^{m} \max(1, \sum_{j=1}^{n} a_j^i)
\end{aligned}
$$

ML To attend over the input and generated output separately, Paulus et al. (2018) adapt the seq2seq with attention framework and use attention mechanism both in the encoder (c_t) and decoder ($c_t^{(d)}$), recording which words have been attended in the source text and which words have been generated by the decoder.

$$
\begin{aligned}
a_t^{i(d)} &= \frac{\exp(s_t \cdot s_i)}{\sum_j \exp(s_t \cdot s_j)} \\
c_t^{(d)} &= \sum_i a_t^{i(d)} s_i
\end{aligned}
$$

ML+RL Modeling NLL loss $\mathcal{L}_{\text{nll}}$ assumes that reference summary is given during the training phase, which is unknown in the inference stage, leading to the issue of so-called "exposure bias". At the basis of ML, Paulus et al. (2018) takes a reinforcement learning approach which tries to minimize the negative expected reward $\mathcal{L}_{\text{rl}}$.

$$
\mathcal{L}_{\text{rl}} = -(r(y^s) - r(y)) \sum_{t=1}^{n} \log \hat{y}_t^s
$$

[1] All these equations have been simplified for understandings.

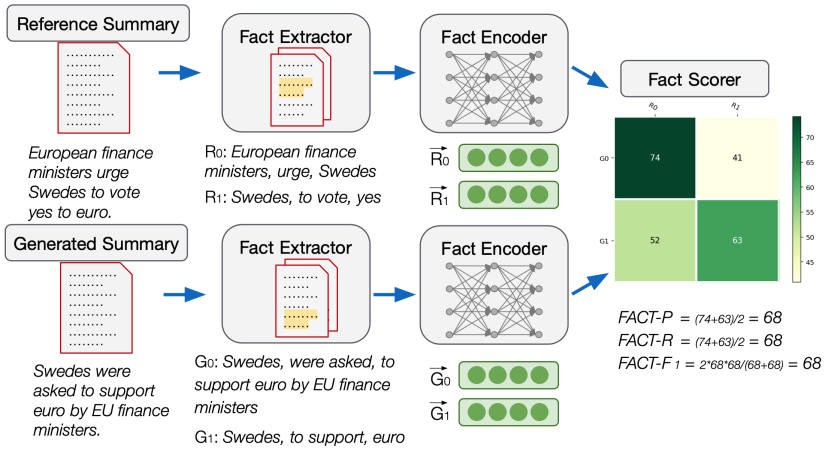

Figure 1: The overview of the factual score computation.

2.2 Fact Extractor

Fact extractor extracts a set of facts implied in the given text. We propose to use open information extraction (OpenIE) methods to extract facts from generated summaries and reference summaries. OpenIE can be formulated as a sequence labeling task. Given an input sequence $\mathbf{w} = (w_1, ..., w_t)$, our goal is to generate a list of tuples. Each tuple is in the form of $(s_1, ..., s_m)$, where each s_i is a contiguous subspan of $\mathbf{w}$. One of the s_i is distinguished as the predicate, while the other spans are considered its arguments (Stanovsky et al., 2018). Among each tuple, we only extract triple $(\text{argument}_0, \text{predicate}, \text{argument}_1)$ as the fact and ignore other spans. We generate a set of facts $\mathcal{G} = \{G_1, ..., G_m\}$ for each generated summary and $\mathcal{R} = \{R_1, ..., R_n\}$ for each reference summary.

2.3 Fact Encoder

Fact encoder embeds each fact to a continuous real space. We simply concatenate the fact triples and use sentence encoder to generate the corresponding fact embeddings. For each fact $G_i \in \mathcal{G}$, we feed $\hat{G}_i = \text{argument}_0 \circ \text{predicate} \circ \text{argument}_1$ into the sentence encoder f. Here, $\circ$ denotes the string concatenation operation, and the sentence encoder f maps the concatenated fact to its vector representation $\vec{G}_i = f(\hat{G}_i)$. The same process applies to each $R_i \in \mathcal{R}$.

2.4 Factual Scorer

Given each pair of generated and reference summary and their fact embeddings, we use cosine-similarity to estimate their relevance, and evaluate the precision, recall, and F1 by averaging across facts from generated summary and facts from reference summary. For each pair of fact embeddings $\vec{G}_i$ and $\hat{R}_j$, the similarity is computed as $s_{ij} = \frac{\vec{G}_i \cdot \vec{R}_j}{\|\vec{G}_i\| \|\vec{R}_j\|}$. The factual precision FACT-P $= \frac{\sum_{i=1}^{m} \max_{j=1}^{n} s_{ij}}{m}$, the factual recall FACT-R $= \frac{\sum_{j=1}^{n} \max_{i=1}^{m} s_{ij}}{n}$, and the FACT-$F1$ $= \frac{2\text{FACT-}P \cdot \text{FACT-}R}{\text{FACT-}P + \text{FACT-}R}$.

3 Experiments

3.1 Dataset

CNN/DM is the most commonly-used corpora for neural abstractive summarization. We preprocess the dataset followed by Paulus et al. (2018). Data statistics are listed in Table 2.

Data Split			Source	Reference	
Train	Dev	Test	#Tokens	#Tokens	#Facts
287K	13K	11K	384.0	61.3	9.7

Table 2: CNN/DM dataset statistics. Average number of tokens and extracted facts are reported.

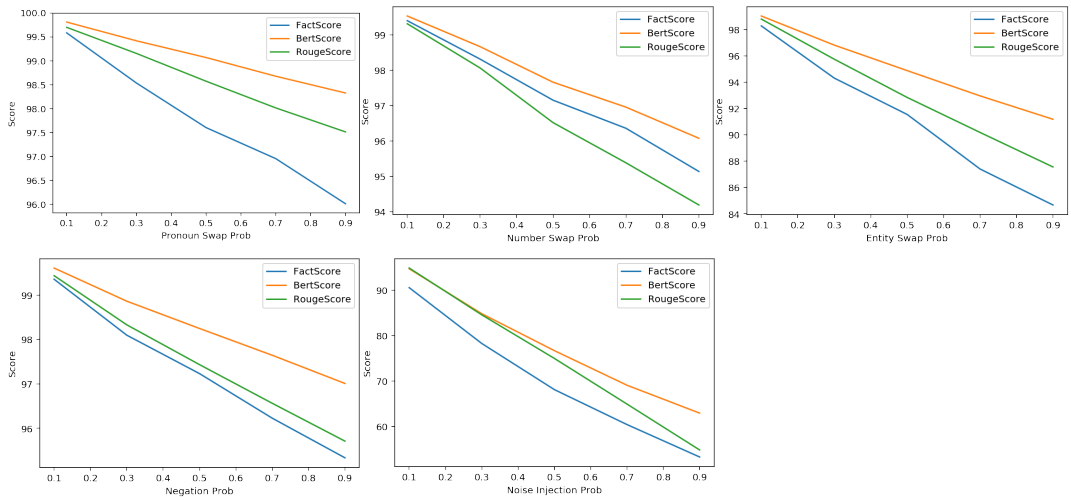

Figure 2: Falsity attack. X-axis: Text transformation probability. Y-axis: Scores.

3.2 Experimental Setup

For abstractive summarization, we implement Seq2seq and Pointer-Generator models. We use 1-layer BiLSTM (512 hidden size) as encoder and 1-layer LSTM as decoder (512 hidden size). We train the model for 10 epochs with 32 batch size. We use Adagrad with 0.15 initial learning rate. We select the best model based on the perplexity achieved on the dev set. We generate summaries using beam search with 10 beam size. We limit the minimum generation length as 35. We block repeated 3-gram. Our implementation could achieve 28.52 and 35.37 ROUGE-L score on CNN/DM for Seq2seq and Pointer-Generator, respectively. In the final experiments, we directly use around 500 sampled generated summaries from the four systems (Chaganty et al., 2018).

For factual score computation, we use the default settings of AllenNLP OpenIE system (Gardner et al., 2017) to extract facts from reference summaries and generated summaries, and we use Google universal sentence encoder (Cer et al., 2018) to generate fact embeddings.

3.3 Result

We evaluate each model by ROUGE-L score (Lin, 2004), BERT score (Zhang et al., 2019) and our proposed factual score (Table 3.3). ROUGE-L score evaluates the n-gram hard-match between generated summaries and reference summaries, while BERT score measures word soft-match using contextualized word embeddings provided by BERT. Factual score evaluates the factual consistencies between generated and reference summaries. Factual score, as well as ROUGE score, ranks ML+RL > Pointer-Generator > ML > Seq2seq, which is consistent with the human evaluation result.

System	ROUGE Score		BERT Score		FACT Score	
	Mean	Std	Mean	Std	Mean	Std
Seq2seq	19.94	10.89	55.01	6.97	39.61	12.45
Pointer-Generator	27.62	13.68	60.20	7.70	43.49	12.11
ML	26.57	11.67	60.35	6.19	42.83	10.33
ML+RL	28.63	12.16	61.72	6.40	45.13	9.89

Table 3: Evaluations of generated summaries from different models.

4 Analysis

4.1 Falsity Attack

To investigate the sensitivity of the factual score to factual inconsistencies, we define 5 types of common falsities (Kryściński et al., 2019b) that would occur in summarization and generate false examples from reference summaries with these semantically variant transformations (Table 4).

Pronoun Swap	We define 4 groups of pronouns: subject personal pronouns (e.g., *he*), object personal pronouns (e.g., *him*), reflexive pronouns (e.g., *himself*), and possessive pronouns (e.g., *his*). For each pronoun in the reference summary, we replace it under a predefined probability with another pronoun randomly sampled from the same group.
Number Swap	For each number entity in the reference summary, we replace it under a predefined probability with another number entity randomly sampled from the source text. Number entities are detected with NER tools.
Entity Swap	Same as number swap. Replace named entity with another named entity randomly sampled from the source text.
Negation	We define a set of auxiliary verbs with its negations (e.g., *do, don't/do not*). For each auxiliary verb in the reference summary, we randomly flip it with a predefined probability.
Noise Injection	For each token in the reference summary, we randomly duplicate or delete it with a predefined probability.

Table 4: Five types of semantically variant transformations for false example generation.

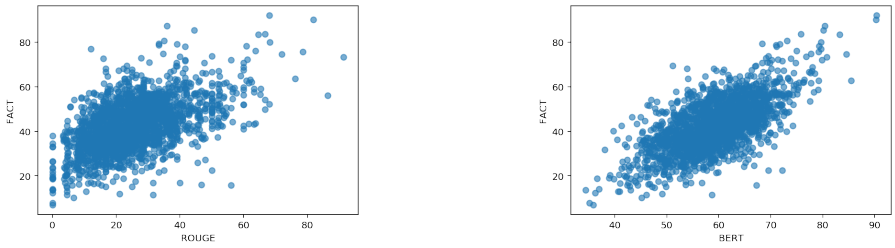

Figure 3: Correlations between factual score and ROUGE score (left) & BERT score (right).

With the predefined probability ranging from 0.1 to 0.9, we evaluate the factual score, as well as ROUGE score and BERT score, using these generated false examples. From Figure 2, we find the factual score has the most sensitivity to falsities compared with ROUGE score and BERT score (except for number swap). This demonstrates that the factual score better captures semantic variances.

It is also worth noting that all the evaluation metrics are much more sensitive to noun phrases transformation (i.e. noise injection, entity swap) than numbers, pronouns and negation transformation. This is the inherent flaw of neural-based evaluation metrics, as those tokens are mapped to locations close together in the embedding space. For example, in number swap, most numbers are treated as special "unknown (UNK)" token in neural models.

4.2 Metric Correlation

To get better tuitions on how the factual score correlates with other evaluation metrics, we investigate the correlation of factual score with ROUGE score and BERT score, respectively (Figure 3).

We find the factual score has strong correlations with both ROUGE score and BERT score, and its correlation with BERT score is stronger than that with ROUGE score. As BERT score is demonstrated to correlate with human evaluation better than ROUGE score (Zhang et al., 2019), it indicates that our factual score may also be more consistent with human evaluation.

5 Discussion and Future Work

From the falsity attack experiment, we conclude that the factual score lacks sensitivity to number swapping, pronoun swapping and negation. In contrast, the encoder is much more sensitive to noun phrases than numbers, pronouns, and negations, which inspires us to design better fact encoder architecture. We also experimented with InferSent (Conneau et al., 2017) as the fact encoder, and it is much less sensitive to each of the transformation than Google universal sentence encoder.

OpenIE models extract facts via sequence tagging, and its output contains duplicated and noisy facts. Future efforts may be devoted to denoising and coreference resolution for OpenIE extractions.

On the other side, the factual score may serve as a novel reward function that could be optimized using reinforcement learning approaches. Future experiments may help to investigate its effectiveness on improving factual correctness in abstractive summarization tasks.

We hope this work would shed light on evaluation metrics of factual correctness for abstractive summarization.

Acknowledgement Yuhui Zhang conducted all the experiments and finished writings independently. Yuhao Zhang (yuhao.zhang@stanford.edu) and Christopher D Manning (manning@stanford.edu) supervised and contributed many valuable ideas for the project. Zhengping Zhou helped proofread and improve the English writings and adjust LaTeXfor page compression.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

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

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019a. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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.

Gabriel Stanovsky, Julian Michael, Luke S. Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *NAACL-HLT*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.