Boosting Factual Consistency and High Coverage in Unsupervised Abstractive Summarization

Yen-Hao Huang
National Tsing Hua University
yenhao0218@gmail.com

Chin-Ting Kuo
National Tsing Hua University
robin.kuo.chin.ting@gmail.com

Yi-Shin Chen*
National Tsing Hua University
yishin@gmail.com

Abstract

Abstractive summarization has gained attention because of the positive performance of large-scale, pretrained language models. However, models may generate a summary that contains information different from the original document. This phenomenon is particularly critical under the abstractive methods and is known as factual inconsistency. This study proposes an unsupervised abstractive method for improving factual consistency and coverage by adopting reinforcement learning. The proposed framework includes (1) a novel design to maintain factual consistency with an automatic question-answering process between the generated summary and original document, and (2) a novel method of ranking keywords based on word dependency, where keywords are used to examine the coverage of the key information preserved in the summary. The experimental results show that the proposed method outperforms the reinforcement learning baseline on both the evaluations for factual consistency and coverage.

1. Introduction

With the rapid growth of the Internet, increasing amounts of data have been produced. Summarization is an important technique to use in order to digest the massive amount of available information. Text summarization aims to utilize machines to condense pieces of text into a shorter version while preserving central information from the original text. There are two major categories for summarization: extractive summarization and abstractive summarization. Extractive summarization identifies the sentence that encapsulates the most important information of the document and extracts it as the summary (Gehrmann et al., 2018; Liu and Lapata, 2019; Nallapati et al., 2017; Zhong et al., 2020). This approach is limited when there is no sentence in the original document that best represents its entire content and there is no way for extractive methods to condense the information.

Abstractive summarization generates a new summary by re-writing the content to a specific length and is more similar to how humans generate summaries (Hsu et al., 2018; Laban et al., 2020; Liu and Lapata, 2019; Paulus et al., 2018; Zhang et al., 2020). However, when abstractive methods rephrase content, the meaning can be completely different, even if the machine only modifies one word; this is referred to as factual inconsistency or storytelling. Such issue is more crucial in the recently widely used pretrained models. Since these model have seen numerous documents, they may copy or produce the same content accidentally based on their memories.

Reinforcement learning (RL) is a recent popular solution to solving factual inconsistency problems. It guides the model by setting up an environment that judges the factual consistency of a summary. It allows the model to self-improve by trial and error in an automatic manner, yet the challenges of RL include designing the proper environment and obtaining desirable rewards.

To have the appropriate environment to judge the summary, this paper leverages the framework that Laban et al. (2020) proposes to address the factual inconsistency problem. This work carefully designs the environment based on the properties of a good summary: (1) ensuring **coverage** of keywords appearing in the document; (2) maintaining **factual consistency** between the summary and original document; and (3) achieving the **fluency** and **brevity** of the generated outcomes.

*The corresponding author.

URI: https://hdl.handle.net/10125/102702
978-0-9981331-6-4

Specifically, three different reword scores are designed for each property for training a RL-based abstractive summarization model.

Overall, the contributions of this work are summarized as: (1) We build a factual consistency scorer to measure the faithfulness of the generated summary based on the *Faithfulness Evaluation with Question Answering* (FEQA) (Durmus et al., 2020). The summaries generated by this method were used to obtain improvements on both the FEQA score and the human evaluation. (2) We propose a novel method of ranking keywords, namely *Top Decorated Word* (TDW), based on the amount of word dependencies. The keywords are used to guide the model regarding the information the summary should contain. The summarizer trained by TDW was able to outperform the baseline on the CNN Dailymail (CNN/DM) dataset. (3) The proposed RL framework allowed an end-to-end training and remained unsupervised.

2. Related Work

In abstractive summarization, Rush et al. (2015) is the first to apply neural networks (NNs) with the attention mechanism for generating a summary in a word-by-word manner. Cohan et al. (2018) and Nallapati et al. (2017) further adopted sequence-to-sequence recurrent NNs (RNNs) to capture hierarchical attention between words and sentences. See et al. (2017) focused on the coverage to keep track of the information been summarized. Hsu et al. (2018) maintained the inconsistency loss to ensure consistency between word and sentence attention. Gehrmann et al. (2018) adopted a bottom-up attention to constrain related content. With the development of transfer learning, recent methods adapted pre-trained language models (LMs) with fine-tuning to generate high-quality summaries (Bao et al., 2020; Lewis et al., 2020; Liu and Lapata, 2019; Radford et al., 2018; Zhang et al., 2020). The fine-tuning process required pre-defined gold summaries, thereby resulting in exposure bias and limiting the writing variety in a summary.

For summary diversity, RL was thus introduced. Paulus et al. (2018) applied RL by directly optimizing on the ROUGE score (Lin, 2004). It resulted in summaries containing a high ROUGE score, but low readability. Chen and Bansal (2018) optimized extraction on sentences with high ROUGE scores and improved readability. Different from previous approaches, Laban et al. (2020) set up several rewards that did not require golden summaries in order to allow an unsupervised training process. Yet, factual inconsistency is still a problem since there are no existing rewards designed specifically for such errors.

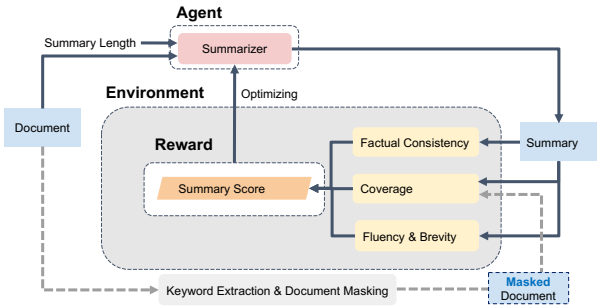


Figure 1. Overview of the proposed framework

Recently, new evaluation methods (Durmus et al., 2020; Scialom et al., 2019; Wang et al., 2020; Zhu et al., 2021) were proposed in order to evaluate the faithfulness and summary consistency of the generated summary. With these methods, this work aims to design novel rewards based on the structure of FEQA (Durmus et al., 2020) to maintain the factual consistency.

3. Framework

This work considers that a good summary only includes the information written from the input document (**factual consistency**), **covers** as much information as possible within a constrained length (**brevity**), and maintains **fluency** between words for human readers. To produce such summaries, a RL-based framework was proposed in Figure 1, which consists of two main parts: **agent** and **environment**. The agent controls the generating process of a good summary, and the environment provides rewards to improve the generated summary. Details of each component and training process are illustrated in the following sections.

3.1. Agent

To produce abstractive summaries, a summary generator, namely a *summarizer*, was set as the RL agent. Given a source document and an expected summary length, the summarizer generates words one-by-one as the document's summary. In this work, a popular text generator that Radford et al. (2018) proposed, namely the Generative Pre-trained Transfomer (GPT2), was selected as the *summarizer*. The main reasons are two-folded: (1) the GPT2 is suitable for abstractive summarization as it generates content in a word-by-word manner; and (2) its generated text relied on its prior context, which can better maintain the consistency.

3.2. Environment

To train a good summarizer, there are four different rewarding scorers in the environment, where each of them represents the standards for a good summary: *factual consistency, coverage, fluency,* and *brevity*.

3.2.1. Factual Consistency During the large-scale pretraining, the text generator, i.e. *summarizer*, has learned language usages from various documents; hence, it may generate summaries with a different meaning from the originals. For instance, it might recall/copy a piece of text from one of its pre-trained documents under a similar context of the given document, resulting in a factual inconsistency.

As our main objective, the factual consistency scorer (FactCon scorer) is designed to ensure that there is information consistency between the generated summary and the input document. One of the most common ways to measure whether there is factual consistency between the source document and summary is to compare the facts between them. Identifying and comparing the facts are the key to this measurement. This work adapted a novel metric for faithfulness by automatic question answering (Durmus et al., 2020), namely FEQA, as our FactCon scorer. An overall example to obtain a factual consistency score is shown in Figure 2. Specifically, an automatic self-question-and-answering (Q&A) process is involved to evaluate factual consistency by following four steps.

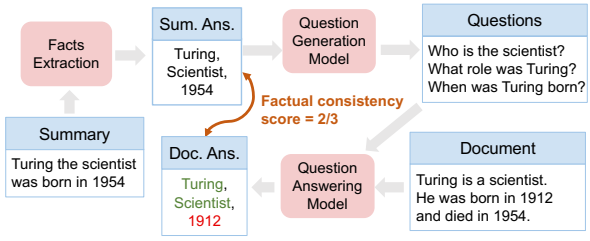


Figure 2. Example of the factual consistency score.

First, the facts are extracted from the generated summary. According to our observation, facts tend to be noun phrases, e.g. subjects and objects of sentences, and usually refer to people, places, and items. Thus, this work extracted noun phrases ($\mathbf{N}$) from the summary using a name entity recognition model from spaCy (Honnibal and Montani, 2017).

Second, a question are generated for each fact. A *question generation model* (QG) was adopted to generate a set of questions ($\mathbf{Q}$) according to the extracted noun phrases. In other words, the answers to the generated questions are the extracted noun phrases, called summary answer, i.e. the $\mathbf{N}$. As a result, the QG was a BART model (Lewis et al., 2020) fine-tuned on a Q&A dataset (Demszky et al., 2018) as in the FEQA.

Third, an answer is obtained from the document for each question. Given the source document d and the $\mathbf{Q}$, a Q&A model then read through d and generated the predicted answers ($\mathbf{A}$) for the $\mathbf{Q}$.

Last, the similarity between the answers of summary and document are calculated and was considered as the factual consistency score. For each question $q_m \in \mathbf{Q}$, it is able to obtain an answer pair of the noun phrase answer and a *generated answer* using the Q&A model, denoted by (n_m, a_m) where $n_m \in \mathbf{N}$ and $a_m \in \mathbf{A}$. Formally, for generated question q_m, its overlap score $sc_m^{\texttt{overlap}}$ was derived using Equation 3.

$$P_m^{\texttt{overlap}} = \frac{\sum_{w \in a_m} \Gamma[w \in a_m]}{|n_m|} \tag{1}$$

$$R_m^{\texttt{overlap}} = \frac{\sum_{w \in n_m} \Gamma[w \in a_m]}{|a_m|} \tag{2}$$

$$sc_m^{\texttt{overlap}} = \frac{2 \times P_m^{\texttt{overlap}} \times R_m^{\texttt{overlap}}}{P_m^{\texttt{overlap}} + R_m^{\texttt{overlap}}} \tag{3}$$

where w denotes the word in answer a_m or n_m, and $\Gamma[.]$ is the indicator function that returns 1 if there is an overlapping; otherwise, it returns 0.

Finally, the factual consistency score $sc_{\texttt{fact}}$ for the generated summary s is is the average of the scores from all of the questions. A higher factual consistency score represents a more reliable generated summary.

$$sc_{\texttt{fact}} = \frac{\sum_{q_m \in \mathbf{Q}} sc_m^{\texttt{overlap}}}{|\mathbf{Q}|} \tag{4}$$

3.2.2. Coverage To evaluate the amount of information covered, this work first generates a masked source document and then answers/recovers the masked words by reading through the generated summary. The scores of coverage are determined based on the number of masks being filled-in successfully. The framework for measuring coverage is illustrated in an example in Figure 3 by an example. Overall, there are two main parts: the *masking source document* and *filling-in thllllle masked document*.

For the **masking source document**, it is important to select the meaningful words to mask. Such target masking words are usually the keywords of a source document since they carry the most important information. Instead of adopting the traditional keywords extraction methods (Laban et al., 2020), e.g. term frequency-inverse document frequency (TFIDF)

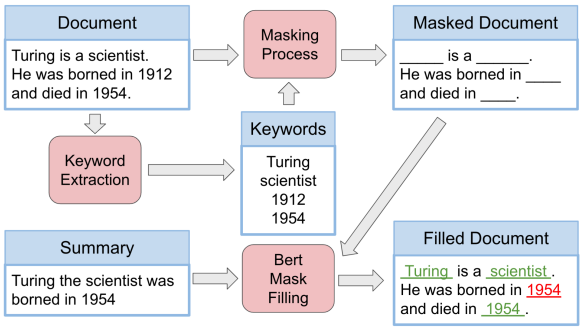


Figure 3. Illustration of the coverage score.

and TextRank (Mihalcea and Tarau, 2004), although they can represent important words relevant to the document, they are limited to extract keywords that have a lower frequency.

Considering that key ideas/words are usually repeatedly emphasized in a document, many other words are used to complement or decorate such keywords. Furthermore, the goal of a sentence is usually to describe the relation of *who did what* or *one thing affects/happened to another*, where such relations could usually be captured by word dependencies (e.g. a sentence's subject, verb, and object). This work, then, proposes a dependency-based method for extracting such keywords, called Top Decorated Words (TDWs). This is obtained by ranking the words in the document according to their number of linked dependencies. Given the set of extracted TDW keywords, a masked source document is obtained by masking the TDW keywords in the source document. Note that the masked keywords are selected based on the top-k on the ranking score since it is too difficult to recover if too many words are masked.

For **recovering the masked document**, a BERT-based (Devlin et al., 2019) model was adapted as in Laban et al. (2020). Let $d^{(\mathtt{m})}$, $\mathtt{BERT}$, $\mathbf{W}_d^{(\mathtt{m})} = \{w_{d,i}^{(\mathtt{m})}\}$, be the masked document, recover model, and set of target masking words, respectively. We can obtain two sets of filling words $\mathbf{W}_d^{(\mathtt{f})} = \{w_{d,i}^{(\mathtt{f})}\} = \mathtt{BERT}(d^{(\mathtt{m})} \oplus s)$ and $\hat{\mathbf{W}}_d^{(\mathtt{f})} = \{\hat{w}_{d,i}^{(\mathtt{f})}\} = \mathtt{BERT}(d^{(\mathtt{m})})$, where $\oplus$ denotes a text concatenation. Finally, the coverage score ($sc_{\mathtt{cover}}$) of the summary is compute as follows:

$$sim(w_1, w_2) = \begin{cases} 1, & \text{if } w_1 = w_2, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

$$sc_{\mathtt{cover}} = \frac{\sum_{i=1}^{|\mathbf{W}_d^{(\mathtt{m})}|} \left(sim(w_{d,i}^{(\mathtt{f})}, w_{d,i}^{(\mathtt{m})}) - sim(\hat{w}_{d,i}^{(\mathtt{f})}, w_{d,i}^{(\mathtt{m})})\right)}{|\mathbf{W}_d^{(\mathtt{m})}|} \tag{6}$$

It is worth noting that the recover model was able to fill in some mask tokens based on the masked document alone without reading the generated summary. The coverage score was, thus, calculated by the score that was boosted by the aids of summary, such that it can indeed reflect *the number of keywords that "summary" covered*. The more masked words that are recovered, the higher the coverage score which the summary obtains.

3.2.3. Fluency and Brevity Since a summary's goal is to condense information and also maintain readability, fluency and brevity are important features to consider. Previous scorers did not focus on the correctness of word usages (e.g. the tense of word and the sentence structure), nor information density. To handle these problems, this work leverages the designs of fluency scores and brevity scores from Laban et al. (2020).

For **fluency**, it is referred to a sequence of words that are well-organized and easily to read. To measure fluency, this work considered when a sequence of words is commonly used in various and multiple documents, as regarded this as a fluent text sequence.

With pretraining on the large corpus, GPT2 learns the proper next coming word given its prior context. In other words, the next word that GPT2 predicts is the most commonly and likely appearing next word based on its training corpus, i.e. the most fluent word. Thus, GPT2 was selected as the *fluency scorer* and was fine-tuned with the target domain data. Finally, the fluency score, $sc_{\mathtt{fluency}}$, is calculated as follows:

$$sc_{\mathtt{fluency}} = 1 - \frac{LM(s) - LP_{low}}{LP_{high} - LP_{low}} \tag{7}$$

where $sc_{\mathtt{fluency}} \in [0, 1]$ (the higher the score, the better its fluency) and $LM(s)$ returns the log-probability of the given summary between the lower/maximum values LP_{low}/LP_{high}, which are empirically selected for GPT2 to control the magnitudes of $sc_{\mathtt{fluency}}$.

The **brevity** score $sc_{\mathtt{brevity}}$ represents the efficiency of presenting information in fewer words. The training guard rails method was adopted from Laban et al. (2020), and it contains three sub-scorers: *No-repetition*, *Finish-your-sentence*, and *No-frame-filling*. Each sub-scorer has a binary effect that if the generated summary does not match its criteria, the sub-scorer then returns 0 and, otherwise, returns 1.

4. Reinforcement Learning

4.1. Rewards and Training Objective

In this work, the final reward, sc, is a weighted aggregated score over the four scorers, calculated by

$$sc = \frac{\alpha\ sc_{\texttt{fact}} + \beta\ sc_{\texttt{cover}} + \gamma\ sc_{\texttt{fluency}} + \delta\ sc_{\texttt{brevity}}}{|\text{ActivatedScorers}|} \tag{8}$$

The Monte-Carlo model was selected as the RL model since its optimization is based purely on the reward.

For the training of RL, the Self-critical Sequence Training (SCST) method (Rennie et al., 2017) was empirically selected since it has shown a performance on the text generation tasks by training models with the cross-entropy-like loss but evaluating with ROUGE or BLEU metrics. With SCST, the summarizer component generates two summaries of document d: (1) the *greedy method*, which always picks vocabulary with the highest probability when decoding; and (2) the *sampling method*, which picks words by *sampling over the probability distribution* on all the vocabulary. Based on the above generated summaries, denoted as $s^{\texttt{greedy}}$ and $s^{\texttt{sample}}$, their corresponding rewards could be obtained as $sc^{\texttt{greedy}}$ and $sc^{\texttt{sample}}$, respectively. Finally, the training objective is to minimize the following loss:

$$\mathcal{L} = (sc^{\texttt{greedy}} - sc^{\texttt{sample}}) \sum_{i=0}^{K} logp(w_i|(w_1, ..., w_{i-1}), d) \tag{9}$$

where K is the target length of the summary and $p(w_i|...)$ represents the probability of the summarizer generating the first word to the ith word. In other words, the summarizer generates a simple summary $s^{\texttt{greedy}}$ and compares it to the sample summary $s^{\texttt{sample}}$. Note that minimizing $\mathcal{L}$ actually maximizes the likelihood of a sample summary. The sample summary should eventually outperform the greedy summary.

4.2. Training Order

Since the summarizer was initialized with GPT2, it is first fine-tuned with the *leading sentence* in order to fit the domain of the downstream task and accelerate the training process. The leading sentence is the first sentence of the document, which is a strong baseline for summarization. The scorers are individually trained. If the parameter of the scorers can update with the summarizer, the scorers could cheat by always returning a high score, regardless of the quality of the summary.

For training the RL, this work adapts a *two-phase training procedure*. For the *first phase*, it is worth noting that the FactCon scorer is not activated at the beginning of the training since the summarizer tends to copy the first few sentences of the original document. The factual consistency score is based on the copied sentence not yet judging the real summary. Therefore, the FactCon scorer is employed at the *second phase* after the summarizer is able to produce high-quality summaries.

Table 1. Dataset statistics of CNN/DM.

	Doc. #	Doc. len.	Sum. len.	Sent. len.
Train	286,817	799.4	59.1	23.8
Valid	13,368	782.3	66.1	24.6
Test	11,490	791.7	58.5	23.4
All	311,675	798.4	59.3	23.8

5. Experimental Setup

In this work, we aim to investigate two research questions (RQs) on the proposed RL framework:

- **RQ1**: How much key information can our generated summary **cover**?
- **RQ2**: Can our method maintain **factual consistency** between the generated summary and original document?

5.1. Dataset

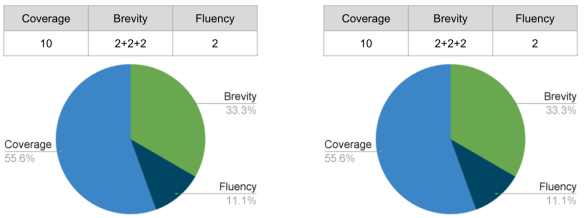
A widely used CNN News and Daily Mail News dataset, CNN/DM (Hermann et al., 2015), modified for summarization (Nallapati et al., 2017) was adopted. Each article contains a pair of news content and highlight. The new highlight was treated as the reference summary of the content. Details are shown in Table 1.

5.2. Model Settings

To train the summarizer, the weight for each scorer in Equation 8 has been considered as an important factors.

At the first phase of training, the weights for coverage, brevity, and fluency scorers were empirically set as shown in Figure 4a (the FactCon scorer is not activated yet). In this phase, the summarizer should mainly focus on summarizing the important content, but also maintain the summary's brevity and fluency. The weight of the coverage, brevity, and fluency scorers were set as 10 (more than 50% of the total weight), 6 (each sub-scorer set to 2), and 2, respectively.

In the second phase, the FactCon scorer is employed whenever the summarizer is able to include key information. Empirically, it is better to activate the FactCon scorer after 30k iterations when the summary obtains both $sc_{\texttt{fluency}}, sc_{\texttt{coverage}} > 0.3$. The updated weights are shown in Figure 4b, where the FactCon and coverage scorer were set to 5 and 15, in order to maintain a balanced and stable training between the factual consistency and the coverage of the generated summary. If the weight of the factual consistency



(a) Weights for first phase. (b) Weights for second phase.

Figure 4. Weight distribution of Equation 8.

score is too high, the coverage score could significantly decrease.

5.3. Baselines

To evaluate the results of the proposed RL framework, this work mainly focused on the comparisons of reward scorers, especially as regards the coverage and factual consistency. Since the framework of the current work was built upon a RL framework, coined as *Summary Loop* (Laban et al., 2020), the *Summary Loop* was considered as our baseline. Overall, four summarizers with different reward designs are trained based on the same RL framework as follows.

For **RQ1**, the RL framework of *Summary Loop* (the baseline) was trained following its original procedure. The baseline utilizes GPT2 as its summarizer and TFIDF for keyword extraction, which is denoted as GPT2+TFIDF hereinafter. This work adapts GPT2 and the proposed TDW as summarizer and keyword extraction method (denoted as GPT2+TDW). In addition to the baseline, *Summary Loop*, the proposed framework was also compared with other non-RL-based summarization methods by obtaining their results from their own reports for a fair comparison.

For **RQ2**, the same setting as RQ1 was adopted but with an additional FactCon scorer during the training for both our method (denoted as GPT2+TDW+FactCon) and the baseline (denoted as GPT2+TFIDF+FactCon).

Each model was expected to generate a one-sentence summary that contained 24 words according to the average sentence length, as in Table 1. All the models were trained using the training dataset. The best performing checkpoint on the validation dataset was selected to be evaluated on the testing dataset. The model was trained with GeForce RTX 2080Ti GPU for more than 80 hours with a batch size of 4.

5.4. Evaluation Metrics

For automatic evaluation, the results were measured by the popular ROUGE metric (Lin, 2004) for **coverage** and using the FEQA (Durmus et al., 2020) for **factual consistencies**. Moreover, a human evaluation was also conduct on the generated summary.

For ROUGE, the standard F1 score with ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L) was adopted on the generated summary for each model. The higher ROUGE score represents the higher coverage between the generated and reference summary.

The FEQA was adopted for factual consistency, as it is an automatic Q&A based metric leveraging the pre-trained LMs for reading comprehension. Given question answer pairs generated from the summary and a Q&A model searching answers from the document, if the answer pair of both sources does not match, this indicates inconsistent information in the summary. The higher the FEQA score, the more factual consistency there is between the generated summary and document.

For human evaluation on factual consistency, five documents were randomly selected from the testing set and the corresponding summaries were generated by four different models as illustrated in Section 5.3. In the questionnaire, the participants were first introduced to the definition of factual consistency in order to ensure their understanding. Then, the participants read the original article and rated each summary individually based on factual consistency (the order of each version was shuffled). Finally, the participants were asked to rank each summary by their factual consistency. Noting that, if the ranking order did not match with the rating order, the reply was considered an invalid sample and discarded. The performance of each model was presented by its averaged *rating* (from 1 to 5) and *ranking* (from 1 to 4) scores of the generated summary. The higher the two scores, the more consistent the facts between the generated summary and document.

6. Result and Analysis

The results of the experiments are divided into two separate sections for evaluating *RQ1* on coverage and *RQ2* on factual consistency. Overall, our model outperformed the baseline in both coverage and factual consistency. Details are shown in the following sections.

6.1. RQ1: Coverage Evaluation

For RQ1, Tables 2 and 3 present the ROUGE results on CNN/DM on the maximum and average values over multiple rounds, respectively. In Table 4, the length constraint of the generated summary is extended to 60 words (averaged length of a summary) in order to compare with other related works.

6.1.1. ROUGE Score Table 2 shows the highest ROUGE score out of 10 experiments. By training

Table 2. **Best ROUGE scores over 10 exps.**

Method	Best ROUGE		
	R-1	R-2	R-L
Summarizer initialized with copy leading sentence			
GPT2+TFIDF (Baseline)	18.495	6.287	19.017
GPT2+TDW (Ours)	**21.020**	**7.319**	**21.254**
GPT2+TFIDF+FactCon	19.901	6.360	19.650
GPT2+TDW+FactCon	19.707	7.010	20.151
Summarizer initialized with the checkpoint from baseline			
GPT2+TFIDF (Baseline)	22.050	7.445	22.334
GPT2+TDW (Ours)	**23.663**	**8.639**	**23.719**
GPT2+TFIDF+FactCon	20.926	6.552	21.026
GPT2+TDW+FactCon	22.091	7.880	22.505

Table 3. **Averaged ROUGE scores for # exps.**

Method	# of Exp.	Avg. ROUGE		
		R-1	R-2	R-L
Summarizer initialized with copy leading sentence				
GPT2+TFIDF (Baseline)	#=5	18.448	6.236	18.963
	#=10	18.419	6.219	18.937
GPT2+TDW (Ours)	#=5	20.920	7.241	21.156
	#=10	**20.942**	**7.259**	**21.167**
GPT2+TFIDF+FactCon	#=5	19.879	6.329	19.620
	#=10	19.886	6.335	19.626
GPT2+TDW+FactCon	#=5	19.661	6.975	20.126
	#=10	19.656	6.963	20.105
Summarizer initialized with checkpoint from baseline				
GPT2+TFIDF (Baseline)	#=5	21.982	7.438	22.271
	#=10	21.969	7.471	22.263
GPT2+TDW (Ours)	#=5	23.636	**8.624**	**23.654**
	#=10	**23.650**	8.617	23.631
GPT2+TFIDF+FactCon	#=5	20.851	6.490	21.010
	#=10	20.850	6.490	21.002
GPT2+TDW+FactCon	#=5	22.021	7.818	22.656
	#=10	22.021	7.817	22.465

baseline model using its suggesting settings (Laban et al., 2020), we found that there is a difficulty reproducing the baseline's reported results. Our reproduced baseline result can only obtain 19.017 on R-L, which was worse than 22.334 by its released checkpoint. Nevertheless, with the keywords selected as the proposed TDW, the same summarizer can cover more important information on different initializing methods and obtain higher ROUGE scores. The proposed method (GPT2+TDW) outperforms the baseline (GPT2+TFIDF) on ROUGE-1/-2/-L by more than 2/1/0.6 points, respectively, when the summarizer (GPT2) pretrained on the leading sentence. When the summarizer was initialized as the checkpoint from the baseline, our method further improved more than one point on all of the ROUGE scores.

In terms of the stability of different models, Table 3 shows the average performance of each model by averaging the ROUGE scores from 5 and 10 experiments separately. Comparing to the baseline, our method obtain the highest average ROUGE score for both initialization cases. These show that by adopting the proposed TDW keywords for Q&A as the RL reward, the summarizer can obtain better and more robust ROUGE scores than utilizing TFIDF.

With the FactCon scorer included, the best and averaged ROUGE scores decreased, as shown in Tables 2 and 3. The main reason is that the ROUGE score is calculated based on the n-gram overlap between the reference and the generated text. In contrast, FactCon may not follow such measures in order to ensure factual consistency. Although the ROUGE score decreased as compared to GPT2+TDW , our GPT2+TDW+FactCon still obtain higher scores than the baselines that without the FactCon scorer. This implies that the proposed GPT2+TDW+FactCon was able to take care of the factual consistency and coverage at the same time, and still perform better than the baseline (GPT2+TFIDF) that was optimized for converage.

6.1.2. Comparing to Non-RL Methods Other works did not generate summary of a sentence's length. To enable a more fair comparison, GPT+TDW was extended to generate 60 words per summary, the average summary length as in Table 1. Table 4 shows the ROUGE scores compared with other popular related works. The proposed GPT+TDW was able to outperform other unsupervised methods on both R-1/-2.

As compared to supervised methods, recent unsupervised methods still performed worse than the recent NN-based methods, especially for the recent breakthroughs in extremely large-scale, pre-trained transformer-based models. This also indicates that it is beneficial to leverage the golden summary (label) in order to enable a supervised learning. The proposed method demonstrated a possibility to summarize the document when there is no available human label, as is the case in many applications and domains.

This work further studied the impacts that ROUGE-1 had on with different lengths since the ROUGE scores were found improved by increasing the summary length from 24 to 60 words. Table 5 shows the results with summarizers initialized with copy leading the sentence and without a FactCon scorer. An incremental trend on the ROUGE-1 score was found by increasing its generating length. This shows that the summarizer can successfully summarize important information if an easier constrain is set. It also implies a possible training procedure by allowing a longer summary at first, and then by gradually reducing its length.

6.1.3. Keyword Selection for Coverage Scorer To construct the coverage scorer, the proper keywords are crucial for masking the document. Except for the proposed TDW, methods that can capture keywords can

Table 4. ROUGE score comparison with supervised and unsupervised methods.

Method	Type	Avg. ROUGE		
		ROUGE-1	ROUGE-2	ROUGE-L
Pointer Generator (See et al., 2017)	Sup.	36.44	15.66	33.42
PG + Coverage (See et al., 2017)		39.53	17.28	36.38
Bottom-Up (Gehrmann et al., 2018)		41.22	18.68	38.34
BERTSUM$_{ABS}$ (Liu and Lapata, 2019)		41.72	19.39	38.76
UNILMv2$_{BASE}$ (Bao et al., 2020)		43.16	20.42	40.14
ERNIE-GEN$_{LARGE}$ (Xiao et al., 2021)		44.02	21.17	41.26
PEGASUS$_{LARGE}$ (Zhang et al., 2020)		44.17	**21.47**	41.11
BART$_{LARGE}$ (Lewis et al., 2020)		44.16	21.28	**40.90**
ProphetNet (Qi et al., 2020)		**44.20**	21.17	41.30
TextRank (Mihalcea and Tarau, 2004)	Unsup.	35.20	12.90	28.70
GPT2 Zero-Shot (Radford et al., 2019)		29.34	8.27	26.58
Summary Loop (Laban et al., 2020)		37.70	14.80	**34.70**
Ours (GPT2+TDW)		**37.71**	**15.12**	34.23

Table 5. ROUGE-1 score of summary length.

	L24	L45	L50	L60
GPT2+TFIDF (Baseline)	18.49	27.34	-	-
GPT2+TDW (Ours)	21.02	29.83	33.19	37.71

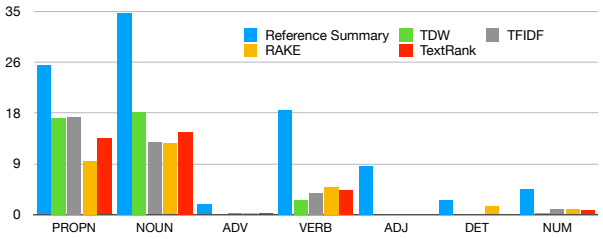


Figure 5. POS percentages of top-20 keywords.

Table 6. Factual consistency score

Method	FEQA-Faith.	# of Ques.
GPT2+TFIDF (Baseline)	39.681	7.029
GPT2+TFIDF+FactCon	43.528	7.489
GPT2+TDW (Ours)	43.172	8.109
GPT2+TDW+FactCon	**49.192**	8.540

be adopted to our framework. This analysis compared the overlaps between the keywords from common extracting methods and the words that appeared in the reference summary. In particular, this work focused on the Part-of-Speech (POS) tags of the top-20 selected keywords from the summary. The result was shown in Figure 5, where all stop words were removed. The figure shows that nouns and proper noun (PROPN) have the most proportion (34.74% and 25.72%) of all POS tags in the summary (blue bars). This is also aligned with our assumption that nouns are usually the subject and object of the sentence. Therefore, the proposed dependency-based TDW can capture the most keywords that overlap with the summary.

6.2. RQ2: Factual Consistency Evaluation

To evaluate factual consistency, we focus on the summarizer's improvement, with or without the FactCon scorer. The results of FEQA and human evaluation are shown in Tables 6 and 7, respectively.

6.2.1. FEQA Score In Table 6, the scores of both baseline and our method improved by more than 3 FEQA points with the aids of FactCon. Meanwhile, without the FactCon scorer, the proposed GPT2+TDW has obtained a higher score than the baseline. With the FactCon scorer, our performance further improves more than six points. It indicates that, with FactCon scorer, the summarizer can successfully preserve faithfulness from the original document to its summary.

To measure factual consistency, FEQA generates questions from the generated summary based on the noun phrases as depicted in Section 3.2.1. The statistics of the amount of questions generated per summary were shown at the right of Table 6. It is observed that the proposed GPT2+TDW could generate one more question than baseline since most of the TDW keywords are nouns and noun phrases. It results in more questions being generated for FactCon scorer to optimize the factual consistency and, thus, perform better. In addition, with FactCon, the number of generated questions increases by 0.4 for both our method and the baseline. Both methods tended to increase the number of noun phrases to reduce the inconsistency. Both FEQA scores were, thus, improved.

Although the FactCon scorer can address the inconsistency, the generated summary is not guaranteed to maintain the same information as the original document. First, FEQA is not available when there are multiple answers since the noun phrase is treated as the only answer. Second, the reliability of the Q&A model is still questionable since it is not 100% accurate. Third, not all questions can be answered within a few words. These are the potential mistakes by the FactCon scorer which could be improved in the future.

Table 7. Human evaluation on factual consistency (higher better).

Method	Rating (0-5)	Ranking (0-4)
GPT2+TFIDF (Baseline)	3.02	2.5
GPT2+TFIDF+FactCon	**3.7**	**2.9**
GPT2+TDW (Ours)	2.5	1.92
GPT2+TDW+FactCon	**3.78**	**2.94**

6.2.2. Human Evaluation To ensure the factual consistency of the generated summary, a human evaluation was also conducted. As shown in Table 7, the rating score improves for more than 0.6 points for more than 0.6 points for GPT2+TFIDF and more than one point for GPT2+TDW. Additionally, the rating score is aligned with the ranking score where the ones with FactCon obtain a higher ranking score. This implies that the participants' replies are reliable.

The result of GPT2+TDW+FactCon is slightly better than GPT2+TFIDF+FactCon. The possible reason for this is that the participants were asked to judge the summary based on factual consistency. As the result, adding the FactCon scorer, which can ensure higher factual consistency, can achieve better evaluations.

7. Conclusion

This work presented a novel, unsupervised abstractive summarization method with RL to improve factual consistency and coverage. The adoption of QG and Q&A models helps maintain faithfulness between the generated summary and the original document. A novel keyword extraction method (TDW) was proposed to maintain coverage of our generated summaries. Our model improved on the baseline RL model by Laban et al. (2020) on the CNN/DM dataset in both factual consistency and coverage evaluation.

For future works, we aim to design the scorer on various aspects for different purpose. For instance, a easy-to-read summary focuses on its generalizability for readers from different knowledge domain.

Acknowledgements

This work is funded by the National Science and Technology Council in Taiwan (NTSC 111-2221-E-007-110-MY3; NTSC 110-2221-E-007-085-MY3; NTSC 108-2221-E-007-064-MY3).

References

Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Gao, J., Piao, S., Zhou, M., et al. (2020). Unilmv2: Pseudo-masked language models for unified language model pre-training. *International Conference on Machine Learning*, 642–652.

Chen, Y.-C., & Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 675–686.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (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*, 615–621.

Demszky, D., Guu, K., & Liang, P. (2018). Transforming question answering datasets into natural language inference datasets.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Durmus, E., He, H., & Diab, M. (2020). FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5055–5070.

Gehrmann, S., Deng, Y., & Rush, A. (2018). Bottom-up abstractive summarization. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4098–4109.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in neural information processing systems*, *28*, 1693–1701.

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.

Hsu, W.-T., Lin, C.-K., Lee, M.-Y., Min, K., Tang, J., & Sun, M. (2018). A unified model for extractive and abstractive summarization using inconsistency loss. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 132–141.

Laban, P., Hsi, A., Canny, J., & Hearst, M. A. (2020). The summary loop: Learning to

write abstractive summaries without examples. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5135–5150.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 74–81.

Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 3730–3740.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411.

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *Thirty-first AAAI conference on artificial intelligence*.

Paulus, R., Xiong, C., & Socher, R. (2018). A deep reinforced model for abstractive summarization. *International Conference on Learning Representations*.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., & Zhou, M. (2020). ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2401–2410.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI blog*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 1179–1195.

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 379–389.

Scialom, T., Lamprier, S., Piwowarski, B., & Staiano, J. (2019). Answers unite! unsupervised metrics for reinforced summarization models. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 3246–3256.

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1073–1083.

Wang, A., Cho, K., & Lewis, M. (2020). Asking and answering questions to evaluate the factual consistency of summaries. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5008–5020.

Xiao, D., Zhang, H., Li, Y., Sun, Y., Tian, H., Wu, H., & Wang, H. (2021). Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*, 3997–4003.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. *Proceedings of the 37th International Conference on Machine Learning*, *119*, 11328–11339.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., & Huang, X. (2020). Extractive summarization as text matching. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6197–6208.

Zhu, C., Hinthorn, W., Xu, R., Zeng, Q., Zeng, M., Huang, X., & Jiang, M. (2021). Enhancing factual consistency of abstractive summarization. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 718–733.