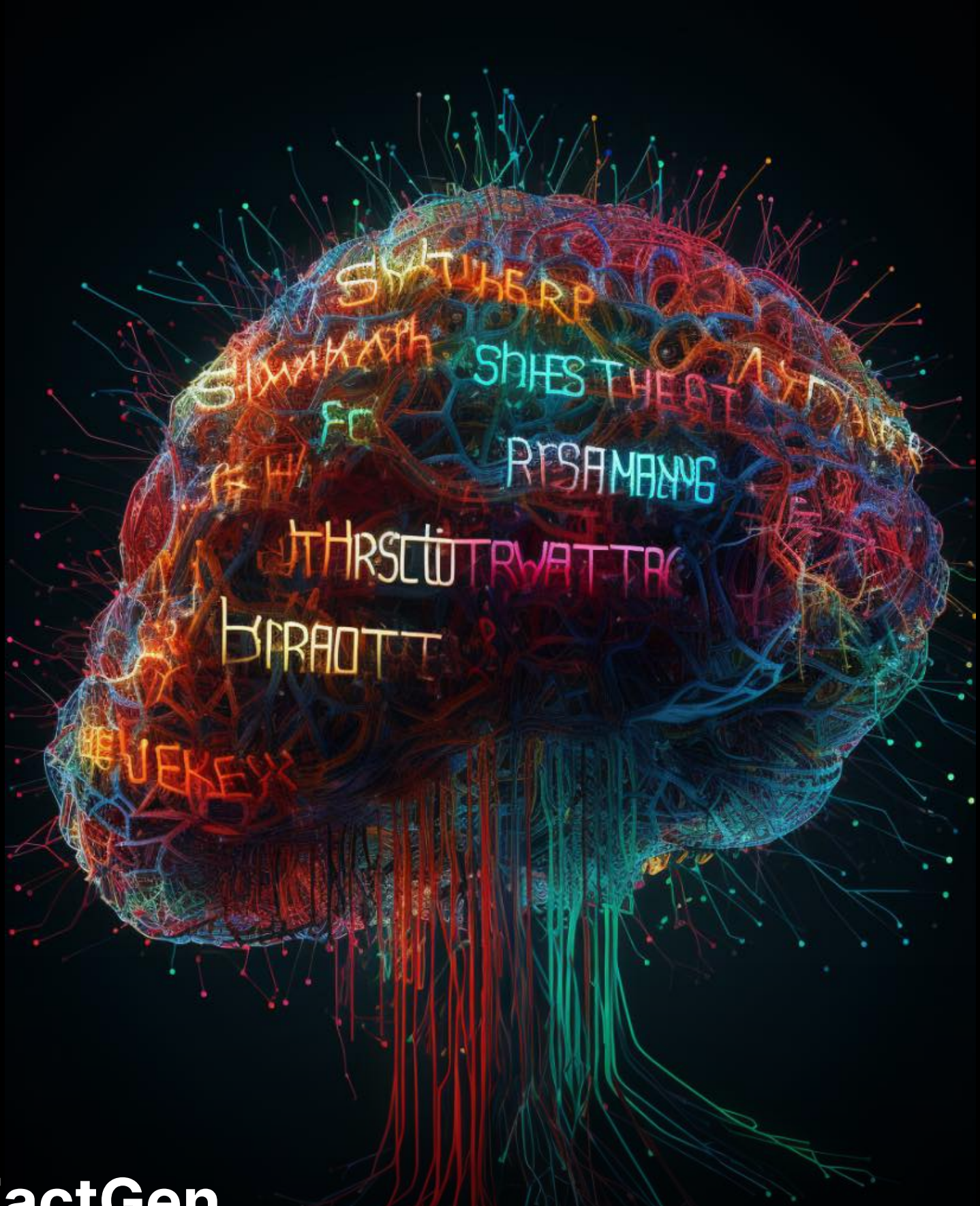


FactGen

FactGen: Faithful Text Generation by Factuality-aware Pre-training and Contrastive Ranking Fine-tuning

2023 *Z. Lan et al.*

Journal of Artificial Intelligence Research 76 (2023) 1281-1303 Submitted 09/2022; published 04/2023

FactGen: Faithful Text Generation by Factuality-aware Pre-training and Contrastive Ranking Fine-tuning

Zhibin Lan LANZHIBIN@STU.XMU.EDU.CN
School of Informatics, Xiamen University, China
Shanghai Artificial Intelligence Laboratory, China

Wei Li LIWEI85@BAIDU.COM
Baidu, Beijing, China

Jinsong Su JSSU@XMU.EDU.CN
(Corresponding author)
School of Informatics, Xiamen University, China
Shanghai Artificial Intelligence Laboratory, China

Xinyan Xiao XIAOXINYAN@BAIDU.COM
Jiachen Liu LIUJIACHEN@BAIDU.COM
Wenhao Wu WUWENHAO@BAIDU.COM
Yajuan Lyu LVYAJUAN@BAIDU.COM
Baidu, Beijing, China

Abstract

Conditional text generation is supposed to generate a fluent and coherent target text that is faithful to the source text. Although pre-trained models have achieved promising results, they still suffer from the crucial factuality problem. To deal with this issue, we propose a factuality-aware pretraining-finetuning framework named **FactGen**, which fully considers factuality during two training stages. Specifically, at the pre-training stage, we utilize a natural language inference model to construct target texts that are entailed by the source texts, resulting in a more factually consistent pre-training objective. Then, during the fine-tuning stage, we further introduce a contrastive ranking loss to encourage the model to generate factually consistent text with higher probability. Extensive experiments on three conditional text generation tasks demonstrate the effectiveness and generality of our training framework.

1. Introduction

With the rapid development of deep learning, conditional text generation has become a hot research topic in natural language processing. Current state of the art conditional text generation models achieve high levels of fluency and coherence, mostly thanks to advances in large pre-trained models (Lewis et al., 2020; Raffel et al., 2020; Zhang et al., 2020). However, current models still suffer from the crucial problem of unfaithful generation (Li et al., 2022), also known as factual inconsistency or hallucinations (Maynez et al., 2020; Pagnoni et al., 2021). For example, in text summarization, Maynez et al. (2020) reveals that hallucinations happen frequently in the model generated outputs. Similarly, in table-to-text (Dhingra et al., 2019; Wang et al., 2020) and dialogue generation (Li et al., 2016; Zhang et al., 2018; Rashkin et al., 2021a; Nie et al., 2021), existing models also suffer from

Text Summarization
Document: Equally, in less than one year, between May 30, 2009, and May 9, 2010, Lampard scored more goals for club and country than Carrick has in his entire career. **Summary:** Steven Gerrard has outscored Carrick in his entire career.
Table-to-text
Source: < Name ID > Samuel Scheidt < Occupation > Composer < Country of citizenship > Germany < Instrument > Organ (music) **Target:** Samuel Scheidt was a German Composer for Piano.
Dialogue Generation
Persona A: Fact 1: I live is a rural farming community. Fact 2: I have a german shepherd dog. Fact 3: I like to watch nhl hockey. **Dialogue History:** Agent A: Hi i am a farmer from iowa. just go in from a long day on the tractor. Agent B: Ah, we are a farming family too. you have any pets? **Generated Responses:** I have a siberian husky named bacon.

Table 1: Examples of hallucinations in different tasks by finetuned BART. Non-factual information in the output are marked in red color.

serious hallucinations. Table 1 shows some examples of hallucinations across tasks. Taking text summarization as an example, the name "Steven Gerrard" in the model-generated summary is inconsistent with the name "Lampard" in the document, although the summary is coherent. It can be said that the unfaithful generation problem severely limits the application of text generation in the real world.

To deal with this issue, researchers have proposed a large number of approaches, which can be roughly classified into four categories: 1) post-processing of model-generated text (Dong et al., 2020; Cao et al., 2020); 2) using external models to extract key information in original text, and then guide the model generation (Saito et al., 2020; Dou et al., 2021) and 3) designing factual consistent training methods for specific tasks (Rebuffel et al., 2020; Li et al., 2020; Cao & Wang, 2021) 4) modifying the beam search process to force the inclusion of pre-specified words and phrases in the output (Tian et al., 2019; Balakrishnan et al., 2019; Mao et al., 2020). Despite their success, the first two types of approaches are limited by external resources or models, while the latter two usually lack generality.

In this paper, we propose a novel training framework for conditional text generation, named **FactGen**, which enhances the consistency of generated text by incorporating fac-

tuality into the whole training pipeline. Overall, our proposed training framework follows the pretrain-finetune paradigm, consisting of two stages: 1) Factuality-aware pre-training. We utilize a natural language inference (Liu et al., 2020) model to help the construction of pre-training instances where sources and targets have entailment relations, forming a pre-training objective to enhance the faithfulness of model generation. 2) Contrastive ranking fine-tuning. At this stage, we firstly use a simple finetuned model to generate multiple candidates, then rank them by factual metrics, and finally optimize the model by a contrastive ranking loss (Hopkins & May, 2011; Zhong et al., 2020; Liu et al., 2022). In this way, the model can distinguish factuality between candidates and generate more faithful outputs.

Compared with previous studies, FactGen is able to more effectively improve faithfulness of text generation due to the following advantages. First, previous studies (Lewis et al., 2020; Raffel et al., 2020) constructed corrupted text with random masks, and then trained a model to reconstruct them. However, random masks may mask important information that cannot be derived from the unmasked parts. By contrast, our factuality-aware pre-training can alleviate the above problem by masking sentences that are entailed by the remaining texts. Second, in the fine-tuning stage, due to the exposure bias problem, the model trained with negative log-likelihood loss cannot distinguish the factuality of candidates well in the inference stage. In other words, the factual consistency of generated texts does not correlate well with their generated probabilities. In contrast, we propose a factuality-aware contrastive loss to train the model to learn higher generation probability for more faithful candidate texts.

We make extensive experiments on three types of text generation tasks. First, we evaluate our method on XSum (Narayan et al., 2018) and CNN/DM (Hermann et al., 2015) datasets for text summarization, WIKIPERSON (Wang et al., 2018) dataset for table-to-text, and dialog NLI (Welleck et al., 2019) dataset for dialogue generation. Our experimental results show substantial improvements in each factuality metric for multiple tasks. Then, we conduct ablation experiments to further validate the effectiveness of different stages. Finally, we perform human evaluation to ensure the authenticity of the results. All results and in-depth analysis strongly demonstrate the effectiveness and generality of our approach.

2. Background

In this section, we firstly introduce the general model architecture for text generation, and then give a brief description of the conventional pretraining-finetuning strategy. Two related models are discussed: 1) BART (Lewis et al., 2020), which is our most important baseline; 2) PEGASUS (Zhang et al., 2020), whose pre-training objective inspires our idea.

2.1 Model Architecture

Currently, conditional text generation mainly adopts Transformer (Vaswani et al., 2017) as the backbone architecture, consisting of an encoder and a decoder.

Encoder The encoder is used to learn the semantic representations of the input text. It contains L identical layers, each of which is composed of a self-attention (SelfAtt) sublayer and a feed-forward network (FFN) sublayer. Let $\mathbf{h}_e^{(l)}$ denote the hidden states of the l-th

encoder layer, it is computed as

$$\mathbf{c}_e^{(l)} = \mathrm{LN}(\mathbf{h}_e^{(l-1)} + \mathrm{SelfAtt}(\mathbf{h}_e^{(l-1)})), \tag{1}$$

$$\mathbf{h}_e^{(l)} = \mathrm{LN}(\mathbf{c}_e^{(l)} + \mathrm{FFN}(\mathbf{c}_e^{(l)})), \tag{2}$$

where $\mathrm{LN}(*)$ denotes layer normalization. Particularly, $\mathbf{h}_e^{(0)}$ is initialized as the embeddings of input tokens.

Decoder The decoder is responsible for generating text under the guidance of the semantic representations learned from the encoder. It also consists of L identical layers. In addition to self-attention sublayer and feed-forward network sublayer, each decoder layer additionally has a cross-attention (CrossAtt) sublayer. Let $\mathbf{h}_d^{(l)}$ denote the hidden states of the l-th decoder layer, it is calculated using the following equations:

$$\mathbf{c}_d^{(l)} = \mathrm{LN}(\mathbf{h}_d^{(l-1)} + \mathrm{SelfAtt}(\mathbf{h}_d^{(l-1)})), \tag{3}$$

$$\mathbf{z}_d^{(l)} = \mathrm{LN}(\mathbf{c}_d^{(l)} + \mathrm{CrossAtt}(\mathbf{c}_d^{(l)}, \mathbf{h}_e^{(L)})), \tag{4}$$

$$\mathbf{h}_d^{(l)} = \mathrm{LN}(\mathbf{z}_d^{(l)} + \mathrm{FFN}(\mathbf{z}_d^{(l)})). \tag{5}$$

Given a training data $D = \{(x, y)\}$, where x is the input text and y is the output text. At each decoding timestep t, the decoder hidden state $h_{d,t}^{(l)}$ is fed into a linear transformation (Linear) layer and then a softmax function, producing the probability distribution of target tokens:

$$p(y_t|y_{<t}, x) = \mathrm{softmax}(\mathrm{Linear}(h_{d,t}^{(L)})). \tag{6}$$

2.2 Two-stage Training

Usually, conditional text generation employs a two-stage training strategy, including pre-training and fine-tuning.

Pre-training Commonly, general-domain or unlabeled data is firstly used to train the model, obtaining initialized parameters.

Typically, at the pre-training stage, BART (Lewis et al., 2020) introduces a denoising autoencoder to pretrain sequence-to-sequence models. In particular, a masking strategy of text infilling is used to sample multiple spans whose lengths are drawn from a Poisson distribution, and then replace each span with a single mask token. The model is trained to reconstruct the original document.

Different from BART, PEGASUS (Zhang et al., 2020) masks multiple whole sentences rather than small text spans to formulate its pre-training objective. Specifically, it designs a gap sentence generation (GSG) pre-training objective, which uses ROUGE-1 as the criterion to select important sentences as target, and treat the remaining sentences as source.

Similar to BART, most pre-trained models (Dong et al., 2019; Raffel et al., 2020) apply random masking to construct their pre-training data, however, the masked information usually cannot be correctly reconstructed from the remaining texts, resulting in unfaithful generation during the pre-training stage. Although PEGASUS selects target sentences according to the ratio of unigram overlapping (i.e. ROUGE-1), there is no guarantee that the selected sentences are faithful to the rest of the document.

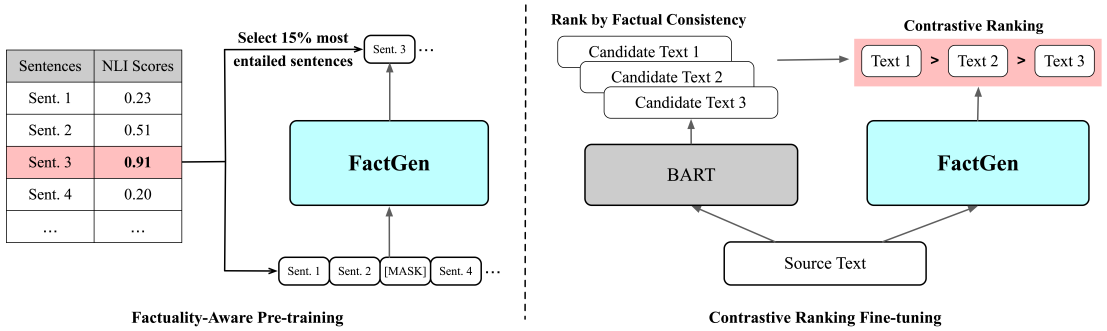


Figure 1: Overview of the FactGen training framework. During the pre-training stage, 15% of sentences with the highest NLI scores will be masked in the input document and concatenated sequentially as the target text. Note that the sentences with the highest scores have a 15% probability of not being masked. In the fine-tuning stage, we just take three candidate texts for instance, they are generated by a simple fine-tuned pre-trained model, then sorted according to their factual consistency scores, and finally let the pre-trained model learn to distinguish them.

Fine-tuning After general pre-training, the model is further fine-tuned on the downstream dataset D_{dt}. The conventional learning objective for a conditional text generation task is to minimize the negative log-likelihood loss (NLL) defined on D_{dt}. Formally, given a training instance $(x, y) \in D_{dt}$, the NLL loss is defined as follows:

$$\mathcal{L}_{nll}(x, y) = \sum_{t=1}^{|y|} -\log p\left(y_t \mid x, y_{<t}\right). \tag{7}$$

However, a series of previous studies (Wang & Sennrich, 2020; Maynez et al., 2020) also found that the exposure bias problem (Ranzato et al., 2016), a discrepancy between training and inference stages, is partially to blame for hallucinations. Furthermore, the model is only optimized to minimize the NLL loss of the reference at token level, which does not explicitly encourage the model to be faithful.

3. Our FactGen Framework

As shown in Figure 1, our training framework is also a two-stage one, including *factuality-aware pre-training* and *contrastive ranking fine-tuning*. Unlike the conventional pretraining-finetuning paradigm, our training framework takes factuality into account at both the pre-training and fine-tuning stages.

3.1 Factuality-aware Pre-training

At this stage, we leverage a general natural language inference (NLI) model to construct pre-training instances where the target is more faithful to the source. By doing so, our strategy only modifies the pre-training target without any impact on the model architecture, thus is model independent and easy to apply in different tasks.

To construct the above-mentioned instances for factuality-aware model training, we must compute the factual consistency score between the source and target for each training instance. Their factual consistency is usually accessed by judging their entailment relationship, and the intuition is that sentences with higher entailment scores have better factual consistency. Thus, we use an NLI model (Liu et al., 2020) to identify targets that are more faithful to the source. Notably, in order to make the pre-trained model general to all tasks, we do not use an NLI model trained on any task-specific data, but a general model that minimizes task-specific modeling.

As shown in the left part of Figure 1, we first employ an NLI model to calculate the entailment score of each sentence and the rest of sentences in the input document, and then select 15% sentences with the highest scores as the target text, which will be masked in the original document. Furthermore, to encourage the model to fully exploit the source text, we keep 15% of the selected sentences unmasked during pre-training.

3.2 Contrastive Ranking Fine-tuning

Given a task-specific training instance (x, y), we then fine-tune the model using the following objective function:

$$\mathcal{L}(x, y) = \mathcal{L}_{nll}(x, y) + \gamma \mathcal{L}_{cr}(x, y), \tag{8}$$

where $\mathcal{L}_{nll}$ is the conventional negative log-likelihood loss mentioned in Section 2.2, $\mathcal{L}_{cr}$ is a contrastive ranking loss that allows the model to distinguish factual consistency across texts, and γ is a hyper-parameter that balances the impacts of these two losses.

Here, we firstly introduce the basic motivation behind our objective function. Intuitively, an ideal model should be able to assign higher probability of generation to candidate texts with better factual consistency. However, using only $\mathcal{L}_{nll}$ cannot guarantee this since it does not consider factual consistency during training.

To deal with this issue, as shown on the right part of Figure 1, we first use the pre-trained model fine-tuned on the task-specific training data to generate N candidate texts via diverse beam search (Vijayakumar et al., 2018). Then, the candidate texts are ranked in a descending order according to task-specific factual evaluation metrics, obtaining the candidate text sequence $y^1, y^2, ..., y^N$. Given a candidate text y^i, we directly consider those $y^1, y^2, ..., y^{i-1}$ ranked above it as positive examples and those $y^{i+1}, y^{i+2}, ..., y^N$ ranked below it as negative examples. Finally, we construct several pseudo training instances to define the contrastive ranking loss as follows:

$$\mathcal{L}_{cr}(x, y) = \sum_i \sum_{j>i} \max(0, f(y^j) - f(y^i) + \lambda_{ij}), \tag{9}$$

where λ_{ij} is the margin multiplied by the rank difference between the candidate texts, i.e., $\lambda_{ij} = (j - i) * \lambda$. $f(y)$ is the estimated log-probability under length normalization, shown as follows:

$$f(y) = \frac{\sum_{t=1}^{|y|} - \log p(y_t | x, y_{<t})}{|y|^\alpha}, \tag{10}$$

where α is the length penalty hyperparameter commonly used in the text generation task.

Apparently, by learning with $\mathcal{L}_{cr}$, the model is trained to minimize the generated probability of text with low factual consistency, while maximizing the generated probability

of text with high factual consistency. In this way, the model can generate more factually consistent text during inference. Please note that these additional models and metrics are not involved during inference.

4. Experiments

In this section, we conduct extensive experiments on three types of text generation tasks.

4.1 Experimental Settings

Dataset We pre-train the model on the RealNews-like (Raffel et al., 2020) dataset (about 35G corpus) and evaluate it on three text generation tasks: text summarization, table-to-text generation, and dialogue generation. For the text summarization task, we evaluate the performance of the model on XSum (Narayan et al., 2018) and CNN/DM (Hermann et al., 2015), both of which are the most commonly-used datasets. For the table-to-text task, we follow Liu et al. (2021a) to conduct experiments on WIKIPERSON (Wang et al., 2018) dataset. In the experiments of dialogue generation, we use the dialogue NLI (Welleck et al., 2019) as our evaluation dataset. In the dialogue NLI dataset, each example contains several kinds of candidates, including a ground-truth utterance, 10 entailment candidates, 10 contradicting candidates and 10 random candidates. Each type of candidate is fed into the model for ranking according to perplexity. Details of these datasets are shown in Table 2.

Implementation Details We use the same architecture as BART-large. Specifically, the model has $L = 12$, $H = 1024$, $F = 4096$, $A = 16$, where L denotes the number of layers for encoder and decoder, H is the hidden size, F is the feed-forward layer size, and A denotes the number of self-attention heads. We use mixed-precision floating point training in both the pre-training stage and the fine-tuning stage. We conduct our experiments on the V100 GPU with 32GB memory. We develop our model based on the open-source toolkit *Transformers*[1].

Factuality-aware Pre-training We use Adam as the optimizer with linear scheduled learning rate 2e-5, a weight decay of 0.01, and set the maximum number of input tokens to be 512 and a maximum number of output tokens to be 256. We use a batch size of 2048. We post-pretrain the full model for 20,000 steps with a warmup of 7,500 steps based on BART.

Contrastive Ranking Fine-tuning We use FactCC (Kryscinski et al., 2020) as the ranking metric in the text summarization task and dialogue generation task, and PARENT (Dhingra et al., 2019) as the ranking metric in the table-to-text generation task. For all datasets, we use Adam as the optimizer with polynomial scheduled learning rate 3e-5, label smoothing of 0.1, training epoch of 5, batch size of 64 and warmup of 10000. All models are simply fine-tuned on their respective datasets before fine-tuning with contrastive ranking loss. Task-specific hyper-parameters are shown in Table 3. We use diverse beam search (Vijayakumar et al., 2018) to generate 16 candidates for each data sample and set γ in Equation 8 to 100 when calculating the combined loss.

1. https://github.com/huggingface/transformers

Datasets	#Train	#Valid	#Test
XSum	204,045	11,332	11,334
CNN/DM	287,227	13,368	11,490
WIKIPERSON	250,186	30,487	29,982
Dialogue NLI	310,110	16,500	12,376

Table 2: Statistics of datasets for evaluation.

Dataset	γ (Eq.8)	α(Eq.10)	λ (Eq.9)
CNN/DM	100	2.0	0.01
XSum	100	1.0	0.1
WIKIPERSON	100	2.0	0.01
Dialogue NLI	100	1.0	0.1

Table 3: Hyper-parameter settings for different datasets.

4.2 Evaluation Metrics

We use different evaluation metrics on three tasks to evaluate the quality of generated texts in two aspects: 1) informativeness. We evaluate the capability of the model in generating non-redundant, meaningful and rich content. 2) factuality. We investigate whether the generated texts are consistent with the input and do not have non-factual errors.

Text Summarization We report ROUGE-1, ROUGE-2, ROUGE-L (Lin, 2004) to evaluate our generated summaries against the reference. In addition, to evaluate factuality, we also report FactCC (Kryscinski et al., 2020), QuestEval (Scialom et al., 2021), and SummacC (Laban et al., 2022), all of which are mainly used in evaluating factual consistency. Among them, FactCC is a weakly-supervised, model-based metric, QuestEval is a QA-based metric, and SummaC is an NLI-based metric achieving SOTA results.

Table-to-text Generation We report BLEU (Papineni et al., 2002) to evaluate informativeness, and PARENT (Dhingra et al., 2019) to evaluate factuality following Liu et al. (2021a).

Dialogue Generation Following Welleck et al. (2019), we report Ent@1 and Con@1 to evaluate factuality and Hit@1 to mainly evaluate informativeness, where both Ent@1 and Con@1 are the variants of the ranking metric Hits@1. They measure the proportions of top-1 candidates returned by the model that are entailment, contradictory or ground-truth, respectively. The models rank the candidates by perplexity.

4.3 Baseline

Since we use BART as our basic model, and thus it is considered as our baseline in all experiments. In addition, we include the following competitive models as baselines.

- **Text Summarization**
  - **PEGASUS** (Zhang et al., 2020). As mentioned in Section 2.2, it masks sentences with largest ROUGE scores from input document and generates them from the remaining sentences. It is important to note that PEGASUS has 1.5x larger parameters than other models. It has two versions, PEGASUS(C4) pre-trained

on 750GB C4 corpus, and PEGASUS(mixed) pre-trained on mixed corpus of C4 and 3.8TB HugeNews. Our pre-training corpus, the RealNews-like dataset, is only part of the C4 corpus. So, PEGASUS(C4)[2] is relatively more comparable to our model than PEGASUS(mixed).

- **CLIFF** (Cao & Wang, 2021). It targets the abstractive summarization task, which firstly takes the reference summary and its back-translation results as positive examples, and constructs multiple negative ones by heuristics, and then learns to distinguish them through contrastive learning.
- **Unlikelihood Training**(**UT**) (Li et al., 2020). It penalizes the probabilities of all tokens in a negative example. Particularly, we reimplement this method on text summarization task and employ back translation to generate positive examples for better training.
- **Loss Truncation**(**LT**) (Kang & Hashimoto, 2020). In a simple and scalable manner, this method adaptively removes examples with high log-loss to improve factual consistency. Note that it can also be applied generically to other tasks.

- **Table-to-text Generation**
  - **Aug-plan** (Liu et al., 2021a). Typically, it incorporates the auxiliary entity information into the model training, including both an augmented plan-based model and an unsupervised model.

    Following Liu et al. (2021a), we also report the performance of **PG-NET** (See et al., 2017) and **Content Matching** (Wang et al., 2020) in table-to-text generation task.
- **Dialogue Generation**
  - **Controllable Features Guidance(CFG)** (Rashkin et al., 2021b). This method proposes three evaluation measures including Objective Voice, Lexical Precision, and Entailment, to distinguish different styles of responses. Different control codes are added during the training based on these metrics. During inference, these control codes act as stylistic controls that encourage the model to generate responses that are faithful to the input.

4.4 Main Results

Text Summarization Table 4 shows the main results of two datasets on the text summarization task. Overall, FactGen outperforms almost all models in terms of all three factual metrics. In particular, FactGen significantly outperforms the basic BART model, achieving improvements of 14.88 and 16.91 FactCC scores on the XSum and CNN/DM datasets, respectively. Although PEGASUS has 1.5x larger model parameters and is pre-trained on a much larger corpus, FactGen achieves better performance on most factuality metrics (except QE on XSUM for PEGASUS(mixed)). It is worth noting that our model also maintains the informativeness of the model-generated results well compared to baseline methods, i.e., the ROUGE score of our model is not severely degraded, and even slightly improved on the CNN/DM dataset.

2. Because PEGASUS(C4) is not open released, we cannot evaluate its factuality performance.

Dataset	Model	R-1	R-2	R-L	CC	QE	SC
XSum	BART	45.10	22.12	36.97	22.54	45.86	37.23
	PEG.(C4)	45.20	22.06	36.99	-	-	-
	PEG.(mixed)	**47.09**	**24.53**	**39.27**	24.40	**47.82**	38.10
	CLIFF	44.63	21.39	36.43	23.51	45.45	**40.29**
	UT	45.56	22.33	37.41	23.88	46.09	37.21
	LT	45.05	22.01	36.96	23.65	45.93	37.88
	FactGen	45.03	21.82	36.86	**37.42**†	45.63	40.25†
CNN/DM	BART	44.30	21.16	41.16	72.95	50.69	78.86
	PEG.(C4)	43.90	21.20	40.76	-	-	-
	PEG.(mixed)	44.08	21.44	40.90	73.36	50.78	79.98
	CLIFF	44.29	21.14	41.02	75.66	50.47	84.24
	UT	43.92	20.91	40.73	75.31	50.45	81.95
	LT	44.16	21.01	41.02	73.00	50.65	78.85
	FactGen	**44.73**	**21.49**	**41.43**	**89.86**†	**51.14**†	**85.68**†

Table 4: Results on the text summarization task. R-1, R-2, R-L denote ROUGE-1, ROUGE-2, ROUGE-L, respectively, for evaluating informativeness. CC, QE, SC denote FacCC, QuestEval, SummaC metrics are used to evaluate factuality. PEG.(C4) and PEG.(mixed) denote PEGASUS(C4) and PEGASUS(mixed), respectively. We use T-test to evaluate the significance of the improvements compared to baseline BART, where † denotes $p < 0.01$.

Model	**BLEU**	**PARENT**
BART	**31.93**	53.06
PG-Net	23.56	50.14
Content Matching	24.56	53.06
LT	31.01	52.65
Aug-plan	17.12	**56.75**
FactGen	29.97	56.72†

Table 5: Results on the WIKIPERSON dataset for the table-to-text generation task. †: significantly better than the baseline BART ($p < 0.01$).

Table-to-text Generation Table 5 provides the evaluation results on the table-to-text task. Likewise, FactGen outperforms almost all previous models in terms of PARENT and is on par with the SOTA Aug-plan model. However, our model achieves a much higher BLEU score than Aug-plan, with an improvement of 12.85. Meanwhile, compared with the BART model, although FactGen exhibits a decrease of 1.96 BLEU scores, it has an improvement of 3.66 PARENT scores, demonstrating that the texts generated by FactGen have stronger factual consistency. Note that in this dataset, the reference text may be noisy due to the hallucinated content, so BLEU is unable to measure the fidelity of the generated text.

Dialogue Generation Table 6 reports the results on the dialogue generation task. As described in Section 4.2, the considered model is required to rank multiple candidates based on perplexity. We can find that although the Hit@1 score of FactGen decreases, compared

Model	Ent@1↑	Con@1↓	Hit@1↑
BART	30.81%	40.41%	16.61%
LT	32.47%	37.27%	**18.45**%
CFG	**37.45**%	36.72%	14.57%
FactGen	37.27%	**30.63**%	14.21%

Table 6: Results on the Dialogue NLI dataset for the dialogue generation task. Higher Ent@1 and lower Con@1 mean better factual consistency.

Model	R-1	R-2	R-L	CC	QE	SC
FactGen	45.03	21.82	36.86	**37.42**†	45.63	**40.25**†
w/o CRL	**45.52**	22.39	**37.45**	23.02	**46.28**†	37.76
w/o pre-training	43.35	20.02	34.94	**40.96**†	44.39	39.19†
w/o CRL and pre-training	45.10	22.12	36.97	22.54	45.86	37.23

Table 7: Ablation study of our framework on the XSum dataset. CRL denotes the contrastive ranking loss defined in Equation 9. †: significantly better than the baseline "w/o CRL and pre-training" ($p < 0.01$).

with other models, it can select the entailment candidate with a higher probability and contradict candidate with a lower probability. Particularly, compared with the BART model, FactGen obtains an improvement of 6.46% Ent@1 point, while a drop of 9.78% Con@1 point.

4.5 Ablation Study

To investigate the effectiveness of different components, we report the performance of variants of our model in Table 7.

We first remove the contrastive ranking loss from the fine-tuning training objective to inspect the performance change of our model. As shown in Line 2 and Line 3, the factual consistency of our model drops, proving that this loss is indeed important for training a model with factual consistency. Then, by comparing the results of Line 3 and Line 5, we can observe that factuality-aware pre-training can significantly improve the model performance in terms of ROUGE and all factual metrics. Besides, from Line 4 and Line 5 we can observe that the contrastive ranking loss still greatly boosts the factual consistency of the model without factuality-aware pre-training, demonstrating the generality of our contrastive ranking fine-tuning method. Finally, from Line 2 and Line 4, we find that factuality-aware pre-training can improve factuality while ensuring that the informativeness (e.g., Rouge-1, Rouge-2, and Rouge-L) will not be greatly reduced. All of these results demonstrate the effectiveness of our method.

4.6 Human Evaluation

Besides, we invite four graduate students with linguistic background to evaluate the informativeness and factuality of several baseline models on two types of generation tasks: 1) XSum for text summarization, and 2) WIKIPERSON for table-to-text generation. From

Model	Factuality			Informativeness		
	Win	Tie	Lose	Win	Tie	Lose
PEGASUS	20.0	60.5	19.5	**19.5**	61.0	19.5
CLIFF	24.5	58.5	17.0	19.0	59.0	22.0
LT	16.0	69.5	14.5	12.0	77.5	**10.5**
FactGen	**28.5**	59.5	**12.0**	18.5	63.5	18.0

(a) XSum

Model	Factuality			Informativeness		
	Win	Tie	Lose	Win	Tie	Lose
LT	26.5	53.5	20.0	19.5	67.0	**13.5**
Aug-plan	24.5	32.5	43.0	23.5	38.0	38.5
FactGen	**39.5**	46.5	**14.0**	**24.0**	59.0	17.0

(b) WIKIPERSON

Table 8: Percentages of generated text that are better than, tied with, or worse than baseline BART in factuality and Informativeness. The Krippendorff's α are 0.64 and 0.36 for the two aspects on XSum, and 0.65 and 0.43 on WIKIPERSON.

XSum Example
Document: Police said they were called to Wingfield Road, Alfreton, at about 03:55 BST on Monday, where the body was discovered. Detectives said the man had suffered head injuries and has not yet been formally identified. The road was closed in both directions and police have appealed for people who were in the area between 03:00 and 04:00 to contact them. Officers also want to hear from anyone who have noticed any damage to a car that could have happened overnight.
BART: A murder investigation has been launched after a man's body was found at a house in Nottinghamshire.
CLIFF: A man has been found dead in a car in Nottinghamshire.
LT: A murder investigation has been launched after a man's body was found on a road in Nottinghamshire.
PEGASUS: A murder investigation has been launched after a man's body was found in a car boot in Derbyshire.
FactGen: The body of a man has been found in a car in a road.

Table 9: Examples of summaries generated by different models on the XSum dataset for text summarization. Non-factual information in the output are marked in red color.

each dataset, we randomly sample 100 examples for evaluation, which are annotated by two students. For each example, we provide the annotators with results generated by the BART model and other comparison models, all models except the BART model are anonymous. We ask annotators to carefully judge whether the texts generated by other models are better than, tie with, or worse than the BART model according to two criterion: 1) factuality, annotators have to judge whether the factual information of the generated text is accurate and faithful to the original text, and 2) informativeness, annotators will judge

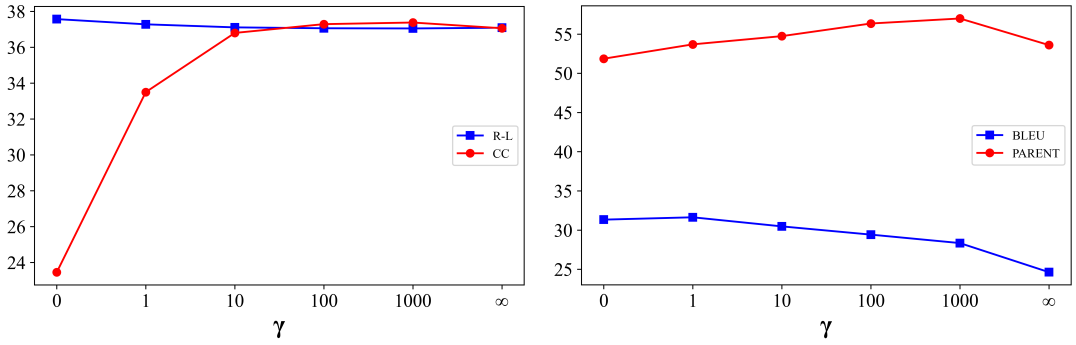
WIKIPERSON Example
Source: < Name ID > Walter William Skeat < place of birth > London < place of death > Cambridge < country of citizenship > United Kingdom < date of birth > 21 November 1835 < date of death > 6 October 1912 < educated at > Christ's College, Cambridge < educated at > Highgate School < educated at > King's College School < languages spoken, written or signed > English language
BART: Walter William Skeat (21 November 1835 – 6 October 1912) was a United Kingdom classical scholar. he was educated at Highgate School King's College School and Christ's College, Cambridge where he graduated B.A. he died in 1912 in Cambridge. skeat was born in 1835 in London the son of William Skeat a merchant and his wife Elizabeth. he was a professor of classical philology at the University of Cambridge from 1876 to his death. he is best known for his work on the grammar of the English language and for his work on the grammar of Latin. **Aug-plan:** Walter William Skeat place of London country of Cambridge country of the United Kingdom (21 November 1835 – 6 October 1912) was a United Kingdom English language poet. he was educated at Highgate School and King's College School languages spoken and Christ's College. **LT:** Walter William Skeat (21 November 1835 – 6 October 1912) was a United Kingdom classical scholar. he was educated at Highgate School King's College School and Christ's College, Cambridge where he graduated B.A. he died in 1912 in Cambridge. he was born in 1835 in London and educated at King's College School and Christ's College, Cambridge where he graduated B.A. he was the author of a number of books on the history of the English language. **FactGen:** Walter William Skeat (21 November 1835 – 6 October 1912) was a United Kingdom classical scholar. he died in Cambridge on 6 October 1912. skeat was born in London on 21 November 1835. he was educated at Highgate School King's College School and Christ's College, Cambridge.

Table 10: Examples of texts generated by different models on the WIKIPERSON dataset for table-to-text generation. Non-factual information in the output are marked in red color.

whether the generated text contains important content in the source text. Particularly, before evaluation, annotators are required to pre-annotate some identical examples and provide the reason behind their annotations. Subsequently, the inconsistent annotations are corrected and the explanations for the corrections are supplied. Through the above pre-annotation, annotators are able to attain a high level of consistency in their annotations. Table 8 shows the results of human evaluation. Comparing with these baselines, *FactGen is more frequently rated as being more faithful and more informative.*

4.7 Case Study

Table 9 and Table 10 show the generation results of different models on the text summarization task and the table-to-text task, respectively. In Table 9, we can observe that the baseline model is prone to produce content that is not mentioned in the input text, e.g. "Nottinghamshire" appears in three baselines, but in fact "Wingfield Road, Alfreton" is



(a) XSum (b) WIKIPERSON

Figure 2: Results on the validation set of XSum and WIKIPERSON with different γ. ∞ denotes removing negative log-likelihood loss $\mathcal{L}_{nll}$ in Equation 8.

not in "Nottinghamshire". Likewise, in Table 10, the baselines also generate some content that is neither supported by the source nor consistent with the facts. Instead, our model does not make such hallucinations but generates concise outputs that are more faithful to the input text. This reveals that our model can reduce extrinsic hallucinations effectively. Other examples can be found in the appendix. A.

4.8 Effects of Hyper-parameters

In our proposed framework, the contrastive ranking loss weight γ (Equation 8) during fine-tuning is an important hyper-parameter, which balances the roles of the two loss items. We perform an in-depth analysis to further understand its impact on faithfulness and informativeness of trained models.

Figure 2 shows the effect of different γ on the XSum and WIKIPERSON validation sets. As γ increases, the factual consistency of the model improves, but the trend stops when γ reaches a certain level, since the NLL loss will be nearly ignored. Besides, we note that there is a slight decrease in informativeness metrics as the factual consistency rises. The negative relationship between factuality and informative metrics is also consistent with prior works, most likely due to hallucinations in the dataset (Kryscinski et al., 2019; Maynez et al., 2020).

4.9 Abstractiveness Analysis

Ladhak et al. (2022) show that the improved factuality of recently proposed methods comes mainly from an increased extractiveness. To further investigate whether our framework improves factuality at the expense of abstractiveness, we analyze the degree of extraction of summaries generated by each model in the XSum dataset. Following Grusky et al. (2018), we compute extractive fragment coverage and extractive fragment density to measure the extractiveness of summaries. Coverage measures the proportion of words extracted from the document. Density measures the average length of the extracted sequence to which each word in the summary belongs. As shown in Table 11, we can find that LT and CLIFF have a higher degree of extraction, implying that they are likely to improve factuality because

Model	Coverage	Density
Reference	0.623	**0.973**
BART	0.700	1.369
PEG.(mixed)	0.689	1.412
CLIFF	0.719	1.491
UT	0.693	1.320
LT	0.721	1.560
FactGen	**0.617**	1.024

Table 11: The coverage and density of the summaries generated by each model on the XSum dataset. Lower coverage and density mean more abstractive.

of an increase in extractiveness. In contrast, except for reference, FactGen has the lowest coverage and density, which means that FactGen does not improve factuality at the expense of Abstractiveness.

5. Related Work

Our related work mainly include three aspects: faithfulness in NLG, contrastive learning for faithful NLG and pretraining for faithful NLG.

Faithfulness in NLG Recently, The faithfulness problem has become the one of the biggest challenges in Natural Language Generation (NLG), which seriously limits the applicability of NLG in practical scenarios. To deal with this issue, four types of approaches are proposed. The first type is post-processing based methods, which introduce a corrector to boost the factuality of output text (Dong et al., 2020; Cao et al., 2020; Song et al., 2020) or utilize an additional scoring module to rerank the candidate outputs obtained via beam search (Zhao et al., 2020; Harkous et al., 2020; Chen et al., 2021). The second type aims to utilize external models to obtain relation triplets (Cao et al., 2018), key information (Saito et al., 2020; Wu et al., 2021) or graph structures (Zhu et al., 2021) from the source text, and then use them to guide model generation. The third type mainly resorts to various learning methods, such as using unlikelihood training (Li et al., 2020) in dialogue generation, reinforcement learning (Rebuffel et al., 2020) in table-to-text generation and contrastive learning (Cao & Wang, 2021) in text summarization. The fourth category focuses on modifying beam search by incorporating specific words or content into the generated output. In this aspect, Balakrishnan et al. (2019) design a constrained decoding strategy that requires the meaning representations of generated text to be not conflict with the input. Tian et al. (2019) propose a confident decoding method, where the low-confidence generated tokens will be skipped. Mao et al. (2020) construct constrained token sets during decoding, where the generation process will continue until all constraints are satisfied. Although the above methods have achieved great success, the first two types are still limited by external models, which the additional models are also involved during inference. By contrast, our model only uses these models during training. The latter two types usually are task specific, whereas our training framework takes factuality into account at both the pre-training and fine-tuning stages. Therefore, our training framework can be applied to a wide variety of tasks.

Contrastive Learning for Faithful NLG Recently, it is common to apply contrastive learning to improve factual consistency for the text summarization task. For example, Cao and Wang (2021) design a task-specific contrastive learning formulation that can help the model better distinguish between positive and negative examples, where positive examples are the reference and its back translation, while negative examples are constructed by heuristic methods. Liu et al. (2021b) make a further step by adding negation words or replacing opposite meaning words to generate more diverse negative examples. Liu et al. (2021b) propose a contrastive summarization framework CO2Sum. This framework applies contrastive learning to both encoder and decoder, allowing the model to be aware of the factual information of input document and generate factual summary. However, they construct negative examples by a heuristic method and only focus on the text summarization task. By contrast, our training framework utilizes the model to generate negative examples and can be applied to multiple text generation tasks.

Pretraining for Faithful NLG Previous studies have explored many effective pre-training objectives, often in the form of masking certain parts of the input. Typically, BART (Lewis et al., 2020) corrupted text with an arbitrary noising function and learns to reconstruct the original text. T5 (Raffel et al., 2020) randomly replaces some spans with single sentinel tokens, and then reconstructs only these replaced spans. PEGASUS (Zhang et al., 2020) selects and masks whole sentences from input document, and concatenates these sentences into a target text. Although these pre-trained models achieve promising performance on text generation tasks, they ignore the factuality of generated texts. Recently, we found a contemporary work FACTPEGASUS (Wan & Bansal, 2022), a model for abstractive summarization consisting of factuality-aware pre-training and modules for enhancing factuality during fine-tuning. Concretely, FACTPEGASUS uses FactCC (Kryscinski et al., 2020) to augment the sentence selection strategy of PEGASUS's pre-training objective. Then it introduces three complementary components (connector, corrector, and contrastor) for improving factuality during fine-tuning. This work targets at the abstractive summarization task, while our model is applicable to different text generation tasks.

6. Conclusion

In this paper, we have studied the factuality of conditional text generation and point out that the conventional pre-training and fine-tuning paradigm do not consider the issue of factuality. To deal with this issue, we have proposed a training framework FactGen, which takes the factuality into account at both the pre-training and fine-tuning stages. The effectiveness and generality of our framework are demonstrated by evaluations on three conditional text generation tasks including text summarization, table-to-text generation and dialogue generation. Moreover, human evaluations also confirm the effectiveness of our proposed framework. In the future, we plan to apply our framework to a wider range of text generation tasks, such as machine translation (Liu et al., 2021a), commonsense generation (Liu et al., 2022) and dialogue generation (Hu et al., 2023). Besides, we will leverage external knowledge to boost our framework.

Acknowledgments

The project was supported by National Key RD Program of China (No. 2022ZD0160501), National Natural Science Foundation of China (No. 62276219), and Natural Science Foundation of Fujian Province of China (No. 2020J06001). This work is also supported by Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University). We also thank the reviewers for their insightful comments.

Appendix A. Sample Outputs

We show some outputs of our model and several baselines on XSum for text summarization in Table 12, and on WIKIPERSON for table-to-text generation in Table 13.

XSum Example 2

Document: Revised growth estimates now suggest the construction industry shrank in the first quarter of 2012, but by less than previously thought. Analysts say the revision may be enough to mean the overall economy narrowly avoided falling into recession for a second time. The ONS is due to give official confirmation of this in June. The revised figures show the construction sector shrank by 5% in the first three months of 2012, less than the 5.4% contraction initially reported. The ONS gives its final estimates for growth in June, and if other parts of the economy remain unchanged, the economy as a whole would register zero growth, rather than a contraction of 0.1%. The economy needs to register two consecutive quarters of negative growth to be in recession. By Stephanie Flanders Economics editor. However, the ONS may also revise the growth of other parts of the economy when it publishes its final estimates in June, such as the much larger services sector, which may offset the gains in construction. Construction accounts for less than 7% of the UK economy, whereas the services sector makes up about three quarters of output. Despite this, the construction sector's weakness has been blamed for dragging down economic growth. The ONS figures showed that construction output fell again in the first three months of this year, reaching its lowest level in 15 years.It was down 2.4% compared the last quarter of 2012. Output has fallen by nearly a fifth since the height of the construction boom in 2008.The ONS will publish the Quarterly National Accounts on 27 June.

BART: The, the Office for National Statistics (ONS) has said.
CLIFF: The UK's construction sector has been revised down by the Office for National Statistics (ONS).
LT: The UK's construction sector may have avoided a recession, according to the Office for National Statistics (ONS).
PEGASUS: The UK economy shrank by 0.1% in the first three months of the year, according to the Office for National Statistics (ONS).
FactGen: The UK economy may have avoided a recession, according to the latest figures from the Office for National Statistics.

Table 12: Examples of summaries generated by different models. Non-factual information in the output are marked in red color

WIKIPERSON Example 2

Source: < Name ID > Tobias Nickenig < date of birth > 1 August 1984 < country of citizenship > Germany < member of sports team > FC Vaduz < start time > 1 January 2009 < member of sports team > 1. FC Köln < member of sports team > FC Erzgebirge Aue < place of birth > Neuwied < position played on team / speciality > Defender (association football) < sport > Association football < league > Bundesliga

BART: Tobias Nickenig (born 1 August 1984 in Neuwied) is a Germany Association football who plays as a Defender (association football) for FC Erzgebirge Aue. nickenig made his debut on the professional league level in the Bundesliga for 1. FC Köln on 1 January 2009 when he came on as a substitute in the 67th minute in a game against VfL Bochum. on 1 January 2009 Nickenig signed a one-year contract with Swiss Super League side FC Vaduz.

Aug-plan: Toby Nickenig (born 1 August 1984) is a Germany footballer who last played for FC Vaduz on 1 January 2009 Steam- 1 January 2008 Toby Nickenigs was born on 1 August 1984 in Neuwied Germany. nickenig began his career with FC Erzgebirge Aue in his home town of NeuWied where he played as a Defender (association football) and made his professional debut in the Bundesliga in the 2002 – 03 season.

LT: Tobias Nickenig (born 1 August 1984 in Neuwied) is a Germany Association football who plays as a Defender (association football) for FC Erzgebirge Aue. nickenig made his Bundesliga debut for 1. FC Köln on 1 January 2009 in a game against 1. FSV Mainz 05. on 1 January 2009 Nickenig signed for Swiss Super League side FC Vaduz on a two-year contract.

FactGen: Tobias Nickenig (born 1 August 1984 in Neuwied) is a Germany Association football who plays as a Defender (association football) for FC Erzgebirge Aue. on 1 January 2009 he signed for FC Vaduz. nickenig made his debut on the professional league level in the Bundesliga for 1. FC Köln on 1 January 2009.

Table 13: Examples of texts generated by different models. Non-factual information in the output are marked in red color

References

Balakrishnan, A., Rao, J., Upasani, K., White, M., & Subba, R. (2019). Constrained decoding for neural NLG from compositional representations in task-oriented dialogue. In *Proceedings of ACL*, pp. 831–844.

Cao, M., Dong, Y., Wu, J., & Cheung, J. C. K. (2020). Factual error correction for abstractive summarization models. In *Proceedings of EMNLP*, pp. 6251–6258.

Cao, S., & Wang, L. (2021). CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of EMNLP*, pp. 6633–6649.

Cao, Z., Wei, F., Li, W., & Li, S. (2018). Faithful to the original: Fact-aware neural abstractive summarization. In *Proceedings of AAAI*, p. 4784–4791.

Chen, S., Zhang, F., Sone, K., & Roth, D. (2021). Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of NAACL*, pp. 5935–5941.

Dhingra, B., Faruqui, M., Parikh, A., Chang, M.-W., Das, D., & Cohen, W. (2019). Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of ACL*, pp. 4884–4895.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., & Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation. In *Proceedings of NIPS*, pp. 13042–13054.

Dong, Y., Wang, S., Gan, Z., Cheng, Y., Cheung, J. C. K., & Liu, J. (2020). Multi-fact correction in abstractive text summarization. In *Proceedings of EMNLP*, pp. 9320–9331.

Dou, Z.-Y., Liu, P., Hayashi, H., Jiang, Z., & Neubig, G. (2021). GSum: A general framework for guided neural abstractive summarization. In *Proceedings of NAACL*, pp. 4830–4842.

Grusky, M., Naaman, M., & Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of NAACL*, pp. 708–719.

Harkous, H., Groves, I., & Saffari, A. (2020). Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of COLING*, pp. 2410–2424.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In *Proceedings of NIPS*, pp. 1693–1701.

Hopkins, M., & May, J. (2011). Tuning as ranking. In *Proceedings of EMNLP*, pp. 1352–1362.

Hu, Z., Cao, Z., Chan, H. P., Liu, J., Xiao, X., Su, J., & Wu, H. (2023). Controllable dialogue generation with disentangled multi-grained style specification and attribute consistency reward. *TASLP*, 188–199.

Kang, D., & Hashimoto, T. B. (2020). Improved natural language generation via loss truncation. In *Proceedings of ACL*, pp. 718–731.

Kryscinski, W., Keskar, N. S., McCann, B., Xiong, C., & Socher, R. (2019). Neural text summarization: A critical evaluation. In Inui, K., Jiang, J., Ng, V., & Wan, X. (Eds.), *Proceedings of EMNLP-IJCNLP*, pp. 540–551.

Kryscinski, W., McCann, B., Xiong, C., & Socher, R. (2020). Evaluating the factual consistency of abstractive text summarization. In *Proceedings of EMNLP*, pp. 9332–9346.

Laban, P., Schnabel, T., Bennett, P. N., & Hearst, M. A. (2022). SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *TACL*, 163–177.

Ladhak, F., Durmus, E., He, H., Cardie, C., & McKeown, K. R. (2022). Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proceedings of ACL*, pp. 1410–1421.

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. In *Proceedings of ACL*, pp. 7871–7880.

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., & Dolan, B. (2016). A persona-based neural conversation model. In *Proceedings of ACL*, pp. 994–1003.

Li, M., Roller, S., Kulikov, I., Welleck, S., Boureau, Y.-L., Cho, K., & Weston, J. (2020). Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of ACL*, pp. 4715–4728.

Li, W., Wu, W., Chen, M., Liu, J., Xiao, X., & Wu, H. (2022). Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods. In *arXiv:2203.05227*.

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

Liu, T., Zheng, X., Chang, B., & Sui, Z. (2021a). Towards faithfulness in open domain table-to-text generation from an entity-centric view. In *Proceedings of AAAI*, pp. 13415–13423.

Liu, W., Wu, H., Mu, W., Li, Z., Chen, T., & Nie, D. (2021b). Co2sum:contrastive learning for factual-consistent abstractive summarization. In *arXiv:2112.01147*.

Liu, X., Liu, D., Yang, B., Zhang, H., Ding, J., Yao, W., Luo, W., Zhang, H., & Su, J. (2022). KGR4: retrieval, retrospect, refine and rethink for commonsense generation. In *Proceedings of AAAI*, pp. 11029–11037.

Liu, X., Yang, B., Liu, D., Zhang, H., Luo, W., Zhang, M., Zhang, H., & Su, J. (2021a). Bridging subword gaps in pretrain-finetune paradigm for natural language generation. In *Proceedings of ACL*, pp. 6001–6011. Association for Computational Linguistics.

Liu, Y., Sun, Y., & Gao, V. (2021b). Improving factual consistency of abstractive summarization on customer feedback. In *Proceedings of the 4th Workshop on e-Commerce and NLP*, pp. 158–163.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2020). Roberta: A robustly optimized bert pretraining approach. In *Proceedings of ICLR*.

Liu, Y., Liu, P., Radev, D., & Neubig, G. (2022). BRIO: Bringing order to abstractive summarization. In *Proceedings of ACL*, pp. 2890–2903.

Mao, Y., Ren, X., Ji, H., & Han, J. (2020). Constrained abstractive summarization: Preserving factual consistency with constrained generation. In *arXiv:2010.12723*.

Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In *Proceedings of ACL*, pp. 1906–1919.

Narayan, S., Cohen, S. B., & Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of EMNLP*, pp. 1797–1807.

Nie, Y., Williamson, M., Bansal, M., Kiela, D., & Weston, J. (2021). I like fish, especially dolphins: Addressing contradictions in dialogue modeling. In *Proceedings of ACL*, pp. 1699–1713.

Pagnoni, A., Balachandran, V., & Tsvetkov, Y. (2021). Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of NAACL*, pp. 4812–4829.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pp. 311–318.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, pp. 1–67.

Ranzato, M., Chopra, S., Auli, M., & Zaremba, W. (2016). Sequence level training with recurrent neural networks. In *Proceedings of ICLR*.

Rashkin, H., Reitter, D., Tomar, G. S., & Das, D. (2021a). Increasing faithfulness in knowledge-grounded dialogue with controllable features. In *Proceedings of ACL*, pp. 704–718.

Rashkin, H., Reitter, D., Tomar, G. S., & Das, D. (2021b). Increasing faithfulness in knowledge-grounded dialogue with controllable features. In Zong, C., Xia, F., Li, W., & Navigli, R. (Eds.), *Proceedings of ACL*, pp. 704–718.

Rebuffel, C., Soulier, L., Scoutheeten, G., & Gallinari, P. (2020). PARENTing via model-agnostic reinforcement learning to correct pathological behaviors in data-to-text generation. In *Proceedings of INLG*, pp. 120–130.

Saito, I., Nishida, K., Nishida, K., & Tomita, J. (2020). Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models. In *arXiv:2003.13028*.

Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., Staiano, J., Wang, A., & Gallinari, P. (2021). QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of EMNLP*, pp. 6594–6604.

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *arXiv:1704.04368*.

Song, H., Wang, Y., Zhang, W.-N., Liu, X., & Liu, T. (2020). Generate, delete and rewrite: A three-stage framework for improving persona consistency of dialogue generation. In *Proceedings of ACL*, pp. 5821–5831.

Tian, R., Narayan, S., Sellam, T., & Parikh, A. P. (2019). Sticking to the facts: Confident decoding for faithful data-to-text generation. In *arXiv:1910.08684*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of NIPS*, pp. 5998–6008.

Vijayakumar, A., Cogswell, M., Selvaraju, R., Sun, Q., Lee, S., Crandall, D., & Batra, D. (2018). Diverse beam search for improved description of complex scenes. In *Proceedings of AAAI*, pp. 7371–7379.

Wan, D., & Bansal, M. (2022). Factpegasus: Factuality-aware pre-training and fine-tuning for abstractive summarization. In *arXiv:2205.07830*.

Wang, A., Cho, K., & Lewis, M. (2020). Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of ACL*, pp. 5008–5020.

Wang, C., & Sennrich, R. (2020). On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of ACL*, pp. 3544–3552.

Wang, Q., Pan, X., Huang, L., Zhang, B., Jiang, Z., Ji, H., & Knight, K. (2018). Describing a knowledge base. In *Proceedings of INLG*, pp. 10–21.

Wang, Z., Wang, X., An, B., Yu, D., & Chen, C. (2020). Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of ACL*, pp. 1072–1086.

Welleck, S., Weston, J., Szlam, A., & Cho, K. (2019). Dialogue natural language inference. In *Proceedings of ACL*, pp. 3731–3741.

Wu, Z., Galley, M., Brockett, C., Zhang, Y., Gao, X., Quirk, C., Koncel-Kedziorski, R., Gao, J., Hajishirzi, H., Ostendorf, M., & Dolan, B. (2021). A controllable model of grounded response generation. In *Proceedings of AAAI*, pp. 14085–14093.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of ICML*, pp. 11328–11339.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., & Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too?. In *Proceedings of ACL*, pp. 2204–2213.

Zhao, Z., Cohen, S. B., & Webber, B. (2020). Reducing quantity hallucinations in abstractive summarization. In *Proceedings of EMNLP Findings*, pp. 2237–2249.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., & Huang, X. (2020). Extractive summarization as text matching. In *Proceedings of ACL*, pp. 6197–6208.

Zhu, C., Hinthorn, W., Xu, R., Zeng, Q., Zeng, M., Huang, X., & Jiang, M. (2021). Enhancing factual consistency of abstractive summarization. In *Proceedings of NAACL*, pp. 718–733.