Using Question Answering Rewards to Improve Abstractive Summarization

Chulaka Gunasekara, Guy Feigenblat, Benjamin Sznajder, Ranit Aharonov, Sachindra Joshi
IBM Research AI
chulaka.gunasekara@ibm.com, {guyf, benjams}@il.ibm.com
{ranit.aharonov2@, jsachind@in.}ibm.com

Abstract

Neural abstractive summarization models have drastically improved in the recent years. However, the summaries generated by these models generally suffer from issues such as: not capturing the critical facts in source documents, and containing facts that are inconsistent with the source documents. In this work, we present a general framework to train abstractive summarization models to alleviate such issues. We first train a sequence-to-sequence model to summarize documents, and then further train this model in a Reinforcement Learning setting with question-answering based rewards. We evaluate the summaries generated by the this framework using multiple automatic measures and human judgements. The experimental results show that the question-answering rewards can be used as a general framework to improve neural abstractive summarization. Particularly, the results from human evaluations show that the summaries generated by our approach are preferred over 30% of the time over the summaries generated by general abstractive summarization models.

1 Introduction

Although neural abstractive summarization has seen drastic improvements over the recent years (Nallapati et al., 2016; See et al., 2017; Paulus et al., 2018; Shi et al., 2021), these systems still have multiple drawbacks. One such common drawback is that the generated summaries frequently fail to capture critical facts in source documents (low recall) (Scialom et al., 2021). On the other hand, neural abstractive summarization models are known to generate content which are inconsistent with the source document (low precision). This is commonly known as hallucination (Kryscinski et al., 2020, 2019). Some studies (Cao et al., 2018) claim that nearly 30% of the outputs of common abstractive summarization models suffer from this problem.

Original Document/Dialog	
Charlee:	I'm in class. Theatre in Portuguese lol.
Curtis:	Reallly?
Charlee:	Yes. One of my subjects at the university that I attend is portuguese theatre.
Charlee:	We are preparing for a performance.
Curtis:	What performance is this? Are you devising it?
Charlee:	A polish one translated into portuguese.
Curtis:	Thats quite cool. Who is the writer?
Charlee:	Mrożek.
Ground truth (human) summary	
	Charlee is attending Portuguese theater as a subject at university. He and other students are preparing a play by Mrożek translated into Portuguese.
Generated Summary 1: Failing to capture critical facts	
	Charlee is preparing for a performance in Portuguese. The writer is Mrożek.
Generated Summary 2: Inconsistent facts with the original document	
	Charlee and Curtis are preparing for a performance in Portuguese. The performance is a Polish one translated into Portuguese.
Generated Summary 3: A summary generated with our approach	
	Charlee is in Portuguese theater class preparing for a Portuguese translation of a Polish play. The writer is Mrożek.

Figure 1: A document, its corresponding ground truth summary and model generated summaries.

Figure 1 shows a source document, the ground truth summary and few summaries generated by neural models. In the *Generated Summary 1*, the model fails to capture some of the crucial facts in the original dialog, such as the play is translated. In *Generated Summary 2*, although the model successfully identifies the fact that the play is a translated, it incorrectly mentions that both Charlie and Curtis are performing. Due to such common factuality related issues, neural abstractive summarization models are hardly usable in real-world applications (Scialom et al., 2021).

In this work, we propose a general framework to alleviate factuality related issues and improve the quality of the abstractive summarization by using question-answering(QA) based rewards. First, we train a sequence-to-sequence(seq2seq) summary generation model to take a document as the input

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 518–526
November 7–11, 2021.

and generate a summary as the output. Next, we improve the precision and recall of the summary generation model using a QA framework as follows. To improve the precision of the model, we first generate questions and corresponding answers for each generated summary. Next, we evaluate the answers that we get for the same questions from the ground truth summaries. If a generated summary contains factually incorrect information, this would lead to having different answers from the ground truth summary for some of the generated questions. We use the similarity of answers to calculate a reward to improve the precision. Similarly, to improve the recall of the summarization model, we generate questions and corresponding answers from the ground truth summaries and evaluate the answers we obtain for the same questions from the generated summaries. If the generated summary does not contain some key information as captured in the ground truth summary, then this would lead to obtaining different answers from the ground truth summary for some of the generated questions. We use the similarity of answers to calculate a reward to improve the recall. The calculated rewards were used in a Reinforcement Learning (RL) based framework to improve the summary generation model. In Figure 1 we show an example output from our approach, which does not contain the factuality related issues shown above. We evaluate the summaries generated by our approach using multiple automatic measures and human judgements, and show that the QA can be used as a general framework to improve abstractive summarization.

In summary, our key contributions are: (1) We introduce a Reinforcement Learning framework, which uses QA rewards to improve the recall and precision of abstractive summarization. (2) The framework is evaluated on three commonly used transformer based summarization models on two public datasets. (3) The evaluation of generated summaries on several automatic measures and human judgements show the effectiveness of our method. In particular, the human judges prefer summaries generated by our approach more than 30% of the time, over the summaries generated by general abstractive summarization models.

2 Related Work

There have been previous work on improving the factual consistency of abstractive summarization models. Cao et al. (2018) used an approach with two encoders, one to encode the source document, and another to encode the facts, and a decoder to attend to the outputs of the two encoders when generating the summary. Zhu et al. (2020) used OpenIE to extract facts and used them in the form of knowledge graphs to improve abstractive summarization. Arumae and Liu (2019) used facts obtained from question-answering rewards to improve extractive summarization. Huang et al. (2020) used multi-choice cloze rewards, in addition to the knowledge graphs to improve the factual consistency. Li et al. (2018) incorporated entailment knowledge into abstractive summarization to improve factual correctness.

There have been several work proposed to evaluate the factuality of summarization algorithms, as more common n-gram based metrics, such as ROUGE (Lin, 2004), are known to perform poorly for this purpose. Most recent approaches proposed for evaluating the factuality are based on QA frameworks (Chen et al., 2018; Eyal et al., 2019; Wang et al., 2020; Deutsch et al., 2020; Durmus et al., 2020; Scialom et al., 2021). The evaluation metrics proposed by the the above studies measure to which extent a generated summary provides sufficient information to answer questions posed on its ground truth summary and whether the questions generated on the generated summary can be answered by the ground truth summary.

3 Improving Summarization with QA Rewards

In general, abstractive summarization models are trained to minimize the cross entropy loss of the reference summary at the word-level, which does not necessarily reward models for being factually accurate with high precision and recall (Maynez et al., 2020). Hence, to improve the factual accuracy of abstractive summarization, we propose a general framework which uses QA based rewards and RL based training. Our proposed framework is illustrated in Figure 2, and below we describe the critical components of the framework.

3.1 Summary Generator

Recent work have leveraged pre-trained Transformer (Vaswani et al., 2017) models for abstractive summarization (Lewis et al., 2019; Zhang et al., 2020). In this work, as the first step of summary generation, we train a transformer based seq2seq model (S), where the source document is fed as

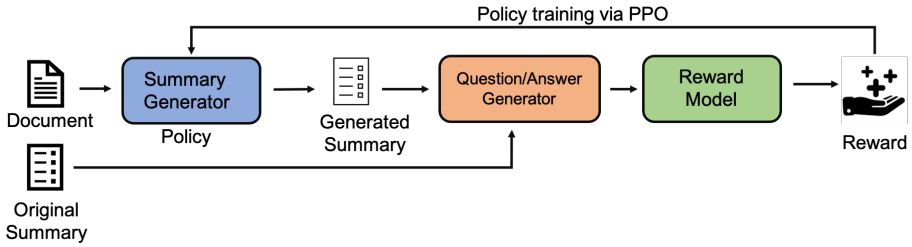

Figure 2: The training process for the summarization framework with QA rewards

the input, and the model is trained to generate the summary token-by-token. The model is trained to optimize the cross entropy loss. During inference, we use top-p nucleus sampling (Holtzman et al., 2019) as the decoding mechanism, with p=0.95.

3.2 Question-Answer Generator

The QA Generator is utilized to generate questions and answers from the original and generated summaries. We generate questions and corresponding answers from the original summary and evaluate the answers obtained for those questions from the generated summary. Similarly, we generate questions and corresponding answers from the generated summary and evaluate the answers obtained for those questions from the original summary. The functionality of the QA framework is explained in Algorithm 1. To generate questions and corresponding answers, we use an answer aware question generation model[1], which is fine-tuned on t5-base (Raffel et al., 2020) model. To identify the answer for a generated question from a summary, we use a extractive QA model[2], which is trained on the SQuAD task (Rajpurkar et al., 2018).

3.3 Reward Model

We use the similarity between the answers obtained by generated and ground truth summaries as the reward function. A generated summary is considered relevant if the questions posed by the ground truth summary can be answered correctly by the generated summary, as this shows the critical information queried by the question is present in the generated summary. Similarly, a generated summary is considered factual if a question generated on the generated summary can be correctly answered by the ground truth summary, as the questions generated on a hallucinated summary will not be correctly answered by the original summary. In this study, use

Algorithm 1: QA Framework for factuality based reward calculation

Input: Trained Summarization Model (S), Question-Answer Generation Model (QA), Answer Generation Model (A), Input Document (D), Ground Truth Summary (G_t), Textual Similarity Function (T)

Output: Reward value(R) for Generated Summary (G_a)

1 Obtain the Generated Summary $G_a = S(D)$

2 Generate the questions and the corresponding answers from G_a, G_t.
(I) $Q_{G_a}, A_{G_a} = QA(G_a)$
(II) $Q_{G_t}, A_{G_t} = QA(G_t)$
where, Q_{G_a} represents the question set generated for the text G_a and A_{G_a} represents the corresponding answer set.

3 Ask the Q_{G_a} from the G_t, and obtain the corresponding answer set ${AG_a}'$ using A. Similarly, ask Q_{G_t} from the G_a, and obtain the corresponding answer set ${AG_t}'$ using A.
(I) ${AG_a}' = A(G_t, Q_{G_a})$
(II) ${AG_t}' = A(G_a, Q_{G_t})$

4 Calculate the reward for G_a by the similarity between ${AG_a}'$ and AG_a as well as similarity between ${AG_t}'$ and AG_t.
$R = Average[T({AG_a}', AG_a) + T({AG_t}', AG_t)]$

the Normalized Levenshtein distance (Yujian and Bo, 2007) as the similarity measure [3]. An example for using QA for reward calculation is provided in Section B of the appendix. The reward 1 is used by the RL framework (shown in Figure 2) to further train the summary generation model S.

3.4 Policy training

We use proximal policy optimization (PPO) (Schulman et al., 2017) as the optimizer for the policy training, as it prevents the generator from moving too far away from the pretrained language model (Wu et al., 2020). We used a publicly available PPO implementation[4] in this study. This approach of

[1] https://huggingface.co/valhalla/t5-base-qg-hl

[2] https://huggingface.co/distilbert-base-cased-distilled-squad

[3] We also considered cosine similarity of BERT embeddings as a distance measure. However the results were not significantly better than using Normalized Levenshtein distance.

[4] https://github.com/lvwerra/trl

QA based optimization following general seq2seq training was used to make this framework applicable across different abstractive summarization models.

4 Evaluation and Results

We evaluate our QA based summarization framework on three common neural abstractive summarization models: **GPT-2** (Radford et al., 2019), **BART** (Lewis et al., 2019) and **PEGASUS** (Zhang et al., 2020). The experiments are performed on two different abstractive summarization datasets: (1) **XSUM** (Narayan et al., 2018): consists of 227k news articles covering a wide variety of subjects along with human written single-sentence summaries, (2) **SAMSUM** (Gliwa et al., 2019): conversation summarization dataset, containing over 13k open-domain conversations and summaries created by humans.

The documents in the XSUM data are fed to the models unaltered. For SAMSUM data, we first preprocess the conversations by replacing the personal names (ex: John) with unique tags (ex:<person_0 >), and then accumulate the utterances in each conversation as follows before feeding them to the models: *<person_1>utterance_1 <person_2>utterance_2 <person_1>utterance_3 ....* In this implementation, we generate one QA pair per sentence in a summary. In addition to that, we also filter out answers that are long (over 5 words), as we believe such long answers do not correspond to the factuality, which is the focus of this study. The average number of QA pairs per summary are 2.5 and 1.4 for SAMSUM and XSUM datasets respectively. The QA based reward process is less expensive in this study, since the number of QA pairs generated are low compared to the studies that generate QA pairs on source documents (not on summaries). We evaluate each model, first, with general method of training: generate the summary given the document, then, with further RL based training with QA rewards that we propose. The hyper-parameters used in training are available in the Section A of the appendix.

Evaluation with ROUGE scores: We first evaluate the models using the ROUGE scores. The obtained results are reported in Tables 1 and 2. Each table contains two sections, where the first section shows the accuracy before training with QA based rewards, and the second section shows the results after RL based training with QA rewards. The results suggest that for both datasets, each model significantly improves ($p < 0.05$) its summarization accuracy using our QA framework.

Factuality based evaluation: We evaluate the results obtained from our models using the factuality based evaluation framework proposed by Scialom et al. (2021). This measure provides better correlation with human judgments over four evaluation dimensions (consistency, coherence, fluency, and relevance) (Scialom et al., 2021), and provides precision, recall and F1 for a generated summary given a reference. The results obtained on the two datasets are shown in Table 3. Similar to the ROUGE based evaluation, the results here clearly indicate that the for both datasets, each model improves its accuracy using our QA framework.

Human Evaluation: We further conducted human evaluations to study the quality of the models. We focused on the two models that obtained the best scores in our automatic evaluations: PEGASUS and BART, and compared the quality of summaries between the original model to our model optimized with QA rewards. For this assessment we first randomly sampled 30 records from the test sets of SAMSUM and XSUM (overall 60 records). Then, we generated 4 types of summaries: PEGASUS, PEGASUS-QA, BART, BART-QA. We followed the evaluation protocol similar to (Wang et al., 2020), in which, the annotators were presented with a document, a ground truth summary and a model summary, and were asked to make two decisions: (1) which model summary is more factual consistent with the given document, and (2) which model summary is of a higher quality, taking into account *Informativeness*, *Fluency*, and *Succinctness*. The annotators were instructed to select one summary or indicate that both summaries are equally good or bad. To achieve a high quality standard we recruited 6 NLP experts, and collected 3 human judgments per each summary. To obtain a single score per summary, we took the majority vote of the collected assessments. More details about human evaluation is available at Section C of the Appendix.

Table 4 describes the results of this assessment. The values represent the number of times that a model was selected as strictly better than its counterpart out of 30 annotated summaries. Differences between QA based reward generation model to the original model is statistically significant (with

Model	R-1	R-2	R-L	R-SU4
GPT-2	42.90	20.75	33.94	19.97
BART	52.85	32.05	44.06	29.58
PEGASUS	52.86	32.36	44.76	30.28
GPT-2-QA	44.94	22.27	35.24	21.46
BART-QA	55.50	33.91	46.20	31.75
PEGASUS-QA	55.43	34.81	47.04	32.46

Table 1: Abstractive summarizers on SAMSUM

Model	R-1	R-2	R-L	R-SU4
GPT-2	25.30	5.61	18.87	8.11
BART	45.58	22.47	37.61	22.38
PEGASUS	47.33	24.59	39.43	24.16
GPT-2-QA	28.73	7.41	21.01	9.85
BART-QA	46.98	23.14	38.31	23.96
PEGASUS-QA	48.11	25.13	41.06	25.28

Table 2: Abstractive summarizers on XSUM

Model	SAMSUM			XSUM		
	P	R	F-1	P	R	F-1
GPT-2	27.88	24.64	26.26	11.52	10.17	10.85
BART	40.93	35.98	38.46	35.40	29.36	32.38
PEGASUS	46.64	36.89	41.77	38.12	32.69	35.40
GPT-2-QA	28.79	28.47	28.63	14.11	13.55	13.82
BART-QA	43.10	41.56	42.33	39.30	31.96	35.63
PEGASUS-QA	47.89	38.96	42.93	41.30	34.24	37.77

Table 3: Results of QA based evaluation

Model	Factual Consistency		Quality	
	SAMSUM	XSUM	SAMSUM	XSUM
BART	6 (20%)	1 (3%)	6 (20%)	4 (13%)
BART-QA	**16 (53%)**	**15 (50%)**	**16 (53%)**	**18 (60%)**
PEGASUS	7 (23%)	5 (17%)	4 (13%)	5 (17%)
PEGASUS-QA	**16 (53%)**	**14 (47%)**	**13 (43%)**	**14 (47%)**

Table 4: Results of human evaluation

$p < 0.05$). These results indicate that QA based rewards helps to significantly improve summary generation model, considering both factual consistency and general quality aspects.

Examples: In Figure 3 we show some examples of model improvements after RL based training with QA rewards. For each model, we show as *Original*, the summary produced by the model before RL training and, as *After RL*, the summary produced by the model after RL training.

5 Conclusion

We investigated the problem of low recall and precision of factuality in neural abstractive summarization models, and proposed a framework to alleviate this issue which uses QA based rewards. The proposed framework is evaluated on three commonly used transformer based summarization models and on two publicly available datasets. The automatic evaluations were performed using ROUGE scores, as well as question answering based evaluation framework and the results suggest that the our method improves the summarization accuracy and factuality. The human evaluation on the generated summaries also suggest that our approach produces summaries with significantly high factual consistency and quality.

Original Document/Dialog	
person_0:	hey babe, what do you want for dinner tonight?
person_1:	gah, don't even worry about it tonight
person_0:	what do you mean? everything ok?
person_1:	not really, but it's ok, don't worry about cooking though, I'm not hungry
person_0:	Well what time will you be home?
person_1:	soon, hopefully
person_0:	you sure? Maybe you want me to pick you up?
person_1:	no no it's alright. I'll be home soon, i'll tell you when I get home.
person_0:	Alright, love you.
person_1:	love you too.
Ground truth summary	
	person_1 will be home soon and she will let person_0 know.
GPT-2 Model	
Original	person_1 wants to grab something for dinner with person_0. person_0 is not hungry. She will pick up something for dinner when she gets home.
After RL	person_1 is away for the evening. person_0 wants to pick him up and person_1 will let him know when he gets home.
BART Model	
Original	person_1 is not hungry tonight. She will be home soon.
After RL	person_1 doesn't want person_0 to cook anything for dinner tonight. She will be home soon and will tell person_0 when she gets home.
Pegasus Model	
Original	person_1 will be home soon. person_0 will pick her up.
After RL	person_1 will tell person_0 when he gets home.

Figure 3: Model improvements after QA based rewards - SAMSUM data

6 Ethics

In this study we used the publicly available SAMSUM (https://huggingface.co/datasets/samsum) and XSUM (https://github.com/EdinburghNLP/XSum) datasets. For the human evaluation, in order to meet a high quality standard, we recruited 6 NLP researchers, who have graduate degree in NLP and Machine Learning. Before the official evaluation started, we sampled 10 tasks to get an estimate of the duration of the task and to make sure the instructions are clear enough.

References

Kristjan Arumae and Fei Liu. 2019. Guiding extractive summarization with question-answering rewards. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2566–2577.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ping Chen, Fei Wu, Tong Wang, and Wei Ding. 2018. A semantic qa-based approach for text summarization evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2020. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *arXiv preprint arXiv:2010.00490*.

Esin Durmus, He He, and Mona Diab. 2020. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *EMNLP-IJCNLP 2019*, page 70.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441.

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

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

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

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. 2021. Questeval: Summarization asks for fact-based evaluation. *arXiv preprint arXiv:2103.12693*.

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

Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K Reddy. 2021. Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science*, 2(1):1–37.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Qingyang Wu, Lei Li, and Zhou Yu. 2020. Textgail: Generative adversarial imitation learning for text generation. *arXiv preprint arXiv:2004.13796*.

Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

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

A Model Training and Hyperparameter Details

In this section, we elaborate the training processes and the hyperparameters used by the models used in this study. Each experiment was run on 2 V100 GPUs (on a single machine).

A.1 GPT2 model

We fine-tune a GPT-2 language model (Radford et al., 2019) for this task by using the implementation available at HuggingFace (Wolf et al., 2019). The hyper-parameters used during training and inference are shown below. The model takes around 3 hours to train for the SAMSUM data and approximately 24 hours to train on the XSUM data. We finetune this on XSUM and SAMSUM datasets in respective applications.

```
model_name: gpt2
per_gpu_train_batch_size: 4
per_gpu_eval_batch_size: 4
gradient_accumulation_steps: 4
learning_rate: 6.25e-5
adam_epsilon: 1e-8
max_grad_norm: 1.0
num_train_epochs: 10
warmup_steps: 500
max_input_tokens: 512
```

A.2 BART model

We used a BART model (Lewis et al., 2019) provided by HuggingFace (Wolf et al., 2019) library[5], which is fine-tuned on the extreme summarization (XSUM) task. During the evaluation with SAMSUM dataset, we further fine-tune this model on SAMSUM data. This model takes around 6 hours to finetune on the SAMSUM data. The code used for the fine-tuning is publicly available[6]. The hyperparameters used for training the BART model are as follows:

```
train_batch_size=4
eval_batch_size=4
num_train_epochs=10
model_name=facebook/bart-large-xsum
learning_rate=3e-5
val_check_interval=0.1
max_source_length=512
max_target_length=80
```

[5] https://huggingface.co/facebook/bart-large-xsum

[6] https://github.com/huggingface/transformers/tree/master/examples/pytorch/summarization

A.3 PEGASUS model

Similar to the BART experiments, we use a PEGASUS model (Zhang et al., 2020) provided by HuggingFace (Wolf et al., 2019) library[7], which is fine-tuned on the extreme summarization (XSUM) task. During the evaluation with SAMSUM dataset, we further fine-tune this model on SAMSUM data. This model takes around 7 hours to finetune on the SAMSUM data. The code used for the fine-tuning is publicly available[8]. The hyperparameters used for training the PEGASUS model are as follows:

```
train_batch_size=4
eval_batch_size=4
num_train_epochs=10
model_name=google/pegasus-xsum
learning_rate=3e-5
val_check_interval=0.1
max_source_length=512
max_target_length=80
```

A.4 Reinforced Learning model with QA rewards

We adapted a publicly available Proximal Policy Optimization (PPO) implementation [9] for the RL model with QA rewards. The model was trained for 10000 steps and takes around 12 hours to train. Following hyper-parameters were used to train the model.

```
steps: 100000
batch_size: 16
forward_batch_size: 4
learning_rate: 1.41e-5
init_kl_coef:0.2
target: 6
horizon:10000
gamma:1
lam:0.95
cliprange: 0.2
cliprange_value: 0.2
vf_coef: 0.1
```

B Example - Reward calculation with Question-Answers

In Figure 4, we provide an example for calculating rewards with QA. The figure first shows a document, with its corresponding ground truth (GT) summary and abstractive summary generated (GEN) by the BART based summarization model. Then the next section shows the QA pairs generated by the GT summary and the answers obtained by the GEN summary for the same questions. For example, for the question 'Who will visit person_1's grandma tonight?', the answer from the GT summary is 'person_1 and person_0' while the answer from the GEN summary is only 'person_1'. Since the model failed to capture the fact that both persons will be visiting grandma, the model will receive a lower reward for this case. Next section shows the questions and answers generated from the GEN summary. For example, for the quesion 'What will person_0 buy for her?', the GEN summary produces the answer 'chocolate and cake' while the GT summary produces 'chocolate' as the answer. This mismatch occurs since GEN summary has some hallucinated content (cake), and this will be penalized with a lower reward during the RL model training.

Original Document/Dialog

person_0: Hi person_1!
person_1: Hello
person_0: Do u have any plans for tonight?
person_1: I'm going to visit my grandma.
person_1: You can go with me.
person_1: She likes u very much.
person_0: Good idea, i'll buy some chocolate for her.

Ground truth summary (GT)

person_1 and person_0 are going to visit person_1's grandma tonight. person_0 will buy her some chocolate.

Generated summary (GEN)

person_1 is going to visit her grandma tonight. person_0 will buy chocolate and cake for her.

Questions/Answers generated on GT

Question:	Who will visit person_1's grandma tonight?
Answer:	person_1 and person_0
GEN answer:	person_1
Similarity:	0.381

Question:	Who will buy her some chocolate?
Answer:	person_0
GEN answer:	person_0
Similarity:	1.0

Questions/Answers generated on GEN

Question:	When will person_1 visit her grandma?
Answer:	tonight
GT answer:	tonight
Similarity:	1.0

Question:	What will person_0 buy for her?
Answer:	chocolate and cake
GT answer:	chocolate
Similarity:	0.5

Reward (Average similarity) = (0.381+1+1+0.5)/4 = 0.72

Figure 4: Reward calculation with Question-Answer pairs

[7] https://huggingface.co/google/pegasus-xsum
[8] https://github.com/huggingface/transformers/tree/master/examples/pytorch/summarization
[9] https://github.com/lvwerra/trl

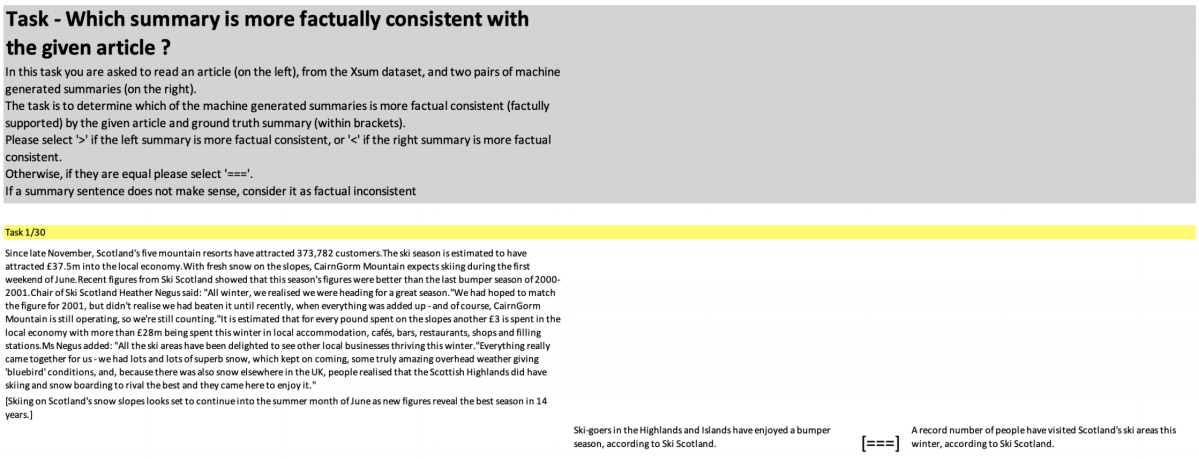

Figure 5: The interface used for human evaluation of the summaries.

C Interface and Instructions for Human Evaluation

Figure 5 shows the annotation interface and instructions that were given to the annotators while working on the factual-consistency human evaluation task. Annotators used a drop-down list to select their judgments ([===],[>],[<]) Notice that following (Wang et al., 2020), ground-truth summaries were prepended back onto the source article (within square brackets).