Extract, Select and Rewrite: A New Modular Summarization Method

Anonymous EMNLP submission

Abstract

Prior works on supervised summarization are mainly based on end-to-end models, leading to low modularity, unfaithfulness and low interpretability. To address this, we propose a new three-phase modular abstractive sentence summarization method. We split up the summarization problem explicitly into three stages, namely knowledge extraction, content selection and rewriting. We utilize multiple knowledge extractors to obtain relation triples from the text, learn a fine-tuned classifier to select content to be included in the summary and use a fine-tuned BART rewriter to rewrite the selected triples into a natural language summary. We find our model shows good modularity as the modules can be trained separately and on different datasets.The automatic and human evaluations demonstrate that our new method is competitive with state-of-the-art methods and more faithful than end-to-end baseline models. [1]

1 Introduction

The task of summarization aims to generate a shorter version of one (or more) input documents that captures most of the salient ideas in the input. Most neural network-based approaches (Rush et al., 2015; Lewis et al., 2020) perform summarization in a single supervised step, training a model to generate summaries to documents from a paired corpus. While this results in fluent summaries, it inevitably results in unfaithfulness as summaries become more abstractive (Durmus et al., 2020).

One approach to mitigate this issues is knowledge augmented summarization. This line of work modifies the sequence-to-sequence architecture of models to incorporate information from relation triples (Cao et al., 2017), knowledge graphs (Zhu et al., 2021; Guan et al., 2021), and topics (Aralikatte et al., 2021). These methods typically augment the source document with the additional input and learn to generate the reference summary by attending to this structured information. They don't explicitly learn content selection as a standalone step so it is unclear how the structured knowledge affects the generated summary.

Another concern with this formulation is that the modularity of end-to-end models is low. These methods could not be separated into different parts explicitly, which means the models could only be trained as a whole, leading to low controllability and low interpretability. Specifically, the content selected to be in the summary is learned implicitly from the data in an end-to-end manner—there exists no formal criteria to identify relevant content within the source document. Since content selection is learned along with text generation, it does not allow for control of the summarization process—different applications and users might have different preferences of what needs to be in the summary (Cao and Wang, 2021).

In order to address these shortcomings, we propose to split the summarization task into three phases, namely knowledge extraction, content selection and rewriting. First, we utilize Information Extraction tools to extract structured knowledge in the form of relation triples from the source text. In the content selection phase, we fine-tune a RoBERTa (Liu et al., 2019a) sentence-pair classification model to select relevant triples from the extracted set. Finally we obtain the summary by rewriting the selected triples into natural language using a fine-tuned BART (Lewis et al., 2020) language model. By decoupling content selection and rewriting, we make the summaries less abstractive and hence reduce the chance of hallucination errors (Durmus et al., 2020) in the text generation phase. Another advantage of the modular setup is that the rewriter does not need paired summarization data to be trained and so for each summarization dataset we only need to train the content selection classification model.

[1] The codes and datasets will be released upon acceptance.

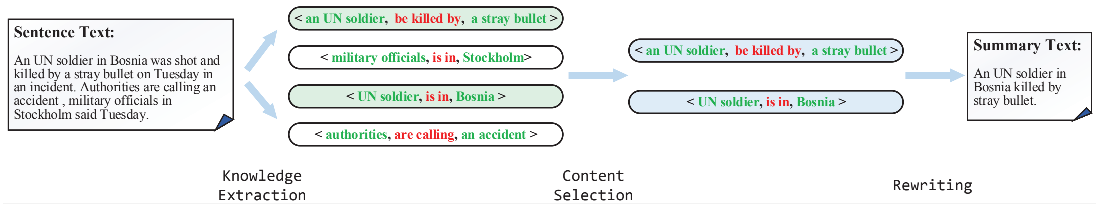

Figure 1: The overview of the three-phase summarization framework.

We run experiments on the Gigaword, DUC-2004 and Reddit-TIFU datasets and find that our approach produces summaries that are competitive to the state-of-the-art on automatic metrics. The generated summaries are more faithful to the source text by the human evaluation. We also observe that the rewriter module can be trained once on standalone text and can be reused across different datasets—a content selector trained on Reddit-TIFU paired with a rewriter from the news domain produces fluent summaries. Besides, this approach to summarization provides more well defined specification for the task allowing for more targeted and interpretable evaluation.

2 Related Work

Abstractive Sentence Summarization Abstractive sentence summarization has been intensively studied in recent years. Rush et al. (2015) proposed a seq2seq structure suitable for sentence summarization, and See et al. (2017) enhanced the model by pointer mechanism. Duan et al. (2019) introduced a transformer summarization model. Devlin et al. (2019) proposed BERT model, and Dong et al. (2019) proposed UNILM model using mask techniques. Lewis et al. (2020) proposed BART model utilizing denoising techniques.

Modular Summarization The existing approaches are two-step extractive-abstractive methods based on sentences. Pilault et al. (2020) and Chen and Bansal (2018) summarize scientific papers and general texts by first extracting sentences from it and then abstractively summarizing them. Krishna et al. (2021) proposed a medical text generation method using modular summarization techniques based on cluster. The "modularity" of these methods mainly defer to combination of neural networks implicitly instead of splitting into different modules explicitly, which is essentially different from our model.

3 Framework

We divide the summarization task explicitly into three phases—Knowledge Extraction, Content Selection and Rewriting, as shown in Figure 1.

Knowledge Extraction To enable fine grained content selection, we extract knowledge from the source documents in the form of `<entity, relation, entity>` triples. To ensure that as many potential knowledge triples in the text can be extracted, we utilize multiple extractors and merge the different triples sets. We extract the knowledge triples from the source sentences in training set as $\mathcal{S}$, triples from the corresponding summaries in training set as $\mathcal{T}$. The extracted triples will be the subtask data sets in the following two phases. Specifically, $\mathcal{S}$ is used to train the content selector, and $\mathcal{T}$ will be used for training rewriting model. $\mathcal{S}$ and $\mathcal{T}$ could be from different datasets.

The extractors used usually generate a large number of redundant triples (candidates with a large overlap with each other). To filter these prior to content selection, we use the Jaccard index on n-grams to calculate the similarity of any two triples:

$$\mathrm{Sim}(x_i, x_j) \stackrel{\text{def}}{=} \lambda_1 \mathrm{J}_{\mathrm{Uni}}(x_i, x_j) + \lambda_2 \mathrm{J}_{\mathrm{Bi}}(x_i, x_j)$$

We remove the redundant triples based on the Jaccard index thresholds, which are determined from the experiments.

Content Selection In content selection phase, we select those knowledge triples that are to be included in the summary out of the candidates generated in knowledge extraction phase. We regard this as a sentence-pair binary classifier on the source sentence and candidate knowledge triple extracted from it. If the triple is to be included in the summary of the document, the sentence-triple pair will be labeled positive, otherwise negative. In order to train this classifier, we need to obtain supervised labels for the triples in the train set, $\mathcal{S}$. For each triple in $\mathcal{S}$, we use ROUGE (Lin, 2004) to measure the similarity to the corresponding summaries, and set a threshold The threshold is determined from the experiments.

Rewriting The selected triples contain all the information to be included in the summary. In rewriting phase, we need to rewrite the content of the selected triples into natural language to produce fluent and grammatically correct summaries. We view this phase as a sequence-to-sequence text generation problem. The subtask dataset for this phase contains the concatenated selected triples from the knowledge extraction phase and their corresponding reference summaries.[2] In order to construct the subtask data set, we concatenate the selected triples in the order of the summary text as the source sequence, and set the corresponding reference summary as the target sequence.

4 Experiments

4.1 Datasets and Experiment Settings

We evaluate our approach using the annotated Gigaword corpus (Rush et al., 2015), with around 3.8M training samples, on the task of supervised sentence summarization. For training the content selection and rewriting models, we constructed the datasets of subtasks in the knowledge extraction phase as detailed in Section 3. In knowledge extraction phase, we utilized Ollie (Mausam et al., 2012), Stanford CoreNLP OpenIE (Angeli et al., 2015) and UW OpenIE (Saha and Mausam, 2018) as the extractors. We fine-tuned RoBERTa-large (Liu et al., 2019b) model as the sentence-pair classifier for content selection, and fine-tuned BART-large (Lewis et al., 2020) model from `fairseq` (Ott et al., 2019) as summary rewriter in rewriting phase. Detailed fine-tuning hyper parameters are in Appendix B.

4.2 Summarization Evaluation

We evaluated the three phases and the quality of the final generated summaries separately.

Phases Evaluations We extract triples from Gigaword dataset for detailed statistics. Table 1 shows the detailed statistics in training and test set. We then create the datasets for fine-tuning the content selector and rewriter. The number of sentence-triple-pair samples is 400k, which is for selector. The size of the rewriting data set is 2M, which is for rewriter fine-tuning. The accuracy of selector is

	Extracted	Valid	Redun.	Pos/Neg
Train Sent	6.34	2.53	60.1%	0.91
Train Summ	4.51	1.76	61.0%	
Test Sent	6.19	2.42	60.9%	

Table 1: Statistics of triples on training and test sets. "*Extracted*" and "*Valid*" are the mean number of the the extracted and valid tripletts (redundance removed). "*Redun.*" is the redundance rate. "*Pos/Neg*" is the positive and negative sample ratio of the constructed data set in selection phase.

Case Study

ST: Zairean president Mobutu Sese Seko will stay at his French Riviera residence until at least the middle of the week because of an increase in diplomatic activity, a Mobutu aide said on Sunday.
Selected Triples:
(Zairean president Mobutu Sese Seko, will stay at, his French Riviera residence)
(Zairean president Mobutu Sese Seko, will stay until, the middle of the week)
Our Model: **Zairean president Mobutu** will stay at his **French Riviera residence** until the middle of week
BART: **Tanzania's** Mobutu to stay at **Riviera residence** until middle of week
Ref: Zairean president Mobutu to stay in France till mid-week

Figure 2: A case study on the Gigaword testset. **ST** is the source text; **Ref** is the reference summary; **BART** is the BART baseline summary; **Selected Triples** is the triples selected in the content selection phase; **Our Model** is the generated summary of our model triples.

84.6%. The ROUGE scores increased more than 1 point after being rewrited comparing to the concatenated selected triples. The detailed metrics of the phases evaluations are showed in the Appendix A.

Automatic and Human Evaluations The final performance is evaluated with the standard ROUGE metrics. We conducted the automatic evaluation on Gigaword test set and DUC-2004 dataset, 1951 and 500 samples separately. We choose some strong sentence summarization models as the comparison baselines. The performances are shown in Table 2 and Table 3 separately.

To test the modularity of our framework, we use a different dataset Reddit TIFU (Kim et al., 2019) for training content selector and rewriter. We perform an ablation where the rewriter is trained on text from Reddit-TIFU and Gigaword and report performance on Reddit-TIFU—the key is that the rewriter does not need paired text to be trained, it can be reused for multiple summarization tasks. We further subsampled 1k samples from Reddit TIFU and Gigaword for training the modules to see how performance varies in the small data regime. The results are showed in Table 4.

To verify that our approach produces more faithful summaries, we ran a user study on Amazon MTurk where crowdworkers annotated summaries to 100 randomly sampled texts from the Gigaword

[2]We are not using paired summarization data. Specifically, any text data will suffice for this phase, even just Wikitext.

Model	R-1	R-2	R-L
PEGASUS (Zhang et al., 2020)	39.12	19.86	36.24
BRXF (Aghajanyan et al., 2021)	**40.45**	**20.69**	36.56
BART (Baseline)	37.28	18.58	34.53
Our Model	39.51	20.07	**36.67**

Table 2: ROUGE F1 scores on the Gigaword test set. Our modular approach outperforms a baseline BART model trained to perform summarization in an end-to-end manner. We also report values from recent works that show that our ROUGE scores are competitive with the supervised state-of-the-art on this dataset. **Bold** indicates the best score in each of R-1, R-2 and R-L.

Model	R-1	R-2	R-L
RT+Conv (Wang et al., 2018)	31.15	10.85	27.68
ALONE (Takase et al., 2020)	32.57	11.63	28.24
WDROP (Takase et al., 2021)	**33.06**	11.45	28.51
BART (Baseline)	31.36	11.40	28.02
Our Model	32.98	**11.82**	**28.74**

Table 3: ROUGE F1 scores on DUC-2004 dataset. Our modular approach outperforms a baseline BART model trained to perform summarization in an end-to-end manner. We also report values from recent works that show that our ROUGE scores are competitive with the state-of-the-art on this dataset. All modules are trained on Gigaword before evaluation on DUC-2004 since DUC-2004 is purely a test set. **Bold** indicates the best score.

Model	R-1	R-2	R-L
PEGASUS (Zhang et al., 2020)	26.63	9.01	21.60
BR3F (Aghajanyan et al., 2021)	*30.31*	*10.98*	*24.74*
BART (Baseline)	24.19	8.12	21.31
Our Model			
$S_R + R_G$	**29.23**	**10.32**	**24.48**
$S_R + R_R$	29.02	10.11	24.06
$S_{R1k} + R_{G1k}$	28.67	9.89	23.80
$S_{R1k} + R_{R1k}$	28.98	10.02	23.90
$S_{R1k} + R_G$	29.01	10.07	23.97

Table 4: ROUGE F1 scores on Reddit TIFU dataset. S_R means the content selector was trained on Reddit TIFU, R_G and R_R mean rewriter trained on Gigaword and Reddit TIFU respectively. 1k means that the module is trained on 1000 randomly sampled article-summary pairs. We see that the rewriter can be trained on text from a larger dataset to enhance performance, indicating that inference on new datasets only requires training a new content selector. We see that our content selector can be trained with a much smaller amount of data to outperform the BART baseline. **Bold** means the best.

Summaries	Sup.	Unsup.	Incoh.	Inconc.
Human-Written	96	3	0	1
BART (Baseline)	90	6	2	2
Our Model	94	3	2	1

Table 5: Human Evaluation of Summaries for Faithfulness from AMT. The summaries from the dataset (Human-Written) and those generated by our model and the BART baseline are annotated by 3 crowdworkers. Summaries are marked as Supported (by the source), Unsupported or Incoherent by each crowdworker. The final label is decided by a majority vote. It is labeled Inconlcusive if there is no agreement. Our model produces more faithful summaries than the baseline.

test set. For each article, we ask crowdworkers to rate summaries of our approach and the baseline (BART), along with outputs by human-written summaries from the original dataset. The results are reported in Table 5. A representative example from Gigaword is shown in Figure 2.

5 Analysis

Automatic evaluation shows that our three-phase model can achieve or approach the state-of-the-art performance on multiple summarization datasets. Also human evaluation shows that our model can enhance the quality of summaries in terms of improving faithfulness. The main reason is that our three-stage model can limit the content of the generated summary in the content selection stage, and then rewrite only selected content. So text generation will introduce less hallucination. In addition, our model has structural advantages. First, our model has a better modularity than other summarization models, as the modules can be trained on different datasets separately to enhance the performance. This means we can modify the modules of the framework to enhance the performance instead of redesigning the entire model. Our model also provides better defined subtask specifications and more transparent evaluations (i.e. evaluate content selection and rewriting separately) for summarization.

6 Conclusion

We propose a three-phase modular abstractive sentence summarization method that obtains competitive performance on automatic metrics while producing more faithful summaries. The modular aspect allows us to train the content selection and rewriting models separately and reuse them on multiple datasets. By decoupling text generation and content selection, we are able to provide a well defined task specification for summation as well. In the future, we are aiming to experiment with more task specific content selectors and adapt our framework to multi-document summarization.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Rahul Aralikatte, Shashi Narayan, Joshua Maynez, Sascha Rothe, and Ryan McDonald. 2021. Focus attention: Promoting faithfulness and diversity in summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6078–6095, Online. Association for Computational Linguistics.

Shuyang Cao and Lu Wang. 2021. Inference time style control for summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5942–5953, Online. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization.

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

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xiangyu Duan, Hongfei Yu, Mingming Yin, Min Zhang, Weihua Luo, and Yue Zhang. 2019. Contrastive attention mechanism for abstractive sentence summarization. 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 3044–3053, Hong Kong, China. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Shuo Guan, Ping Zhu, and Zhihua Wei. 2021. Knowledge and keywords augmented abstractive sentence summarization. In *EMNLP 2021 Workshop on New Frontiers in Summarization*, pages 25–32, Online and in Dominican Republic. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. 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 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Kundan Krishna, Sopan Khosla, Jeffrey Bigham, and Zachary C. Lipton. 2021. Generating SOAP notes from doctor-patient conversations using modular summarization techniques. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4958–4972, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv e-prints*, arXiv:1907.11692.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.

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

Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Association for Computational Linguistics (ACL)*.

Sho Takase, Shun Kiyono, and Sho Takase. 2021. Rethinking perturbations in encoder-decoders for fast training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5767–5780, Online. Association for Computational Linguistics.

Sho Takase, Sosuke Kobayashi, and Sho Takase. 2020. All word embeddings from one embedding. In *Advances in Neural Information Processing Systems*, volume 33, pages 3775–3785. Curran Associates, Inc.

Li Wang, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, and Qiang Du. 2018. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pages 4453–4460. AAAI Press.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

Appendices

A Details of the Generated Summaries

As mentioned in the paper, the summary generation of our model is based on triples extracted from the original text. Therefore, the quality of the extracted triples during inference will affect the quality of the generated abstracts to a certain extent. For example, the length of the final generated summaries will depend on the text length of the triples. In order to ensure the quality of the triplet to the greatest extent, methods such as co-reference resolution will be required.

The metrics for the content selector fine-tuning is showed in Table 6.

In order to evaluate the performance of the rewrite model and verify that the rewrite model can effectively enhance the quality of the generated summary, we compared the ROUGE scores of the concatenated triples (before being rewritten) and the summaries generated by our BART rewriter comparing to the reference summaries. Table Table 7 shows the comparison of ROUGE scores, which verified the rewriting phase enhance the quality of generated summaries.

The length statistics of the generated summaries of our model on Gigaword test set is showed in Table 8.

B Hyper Parameters

The hyper parameters for fine-tuning RoBERTa-large in content selection phase, and BART-large model in rewriting phase are listed. All models are trained and fine-tuned on 2 NVIDIA RTX 2080 Ti GPUs.

B.1 Content Selection

TOTAL_NUM_UPDATES = 3000
WARMUP_UPDATES = 500
LR = 1e-5
NUM_CLASSES = 2
MAX_SENTENCES = 8

B.2 Rewriting

TOTAL_NUM_UPDATES = 10000
WARMUP_UPDATES = 500
MAX_TOKENS = 256
UPDATE_FREQ = 2
LR = 3e-5

Acc.	Rec.	Prec.	F1
84.6%	83.5%	83.7%	83.3%

Table 6: Sentence-pair (article text and triple) binary classification metrics of content selection phase.

	R-1	R-2	R-L
Concat Triples	38.98	18.12	35.76
Rewrite Summary	39.51	20.07	36.82

Table 7: Performance enhancement of the rewriter comparing to the directly concatenated triples on Gigaword datatset.

C The Human Evaluation on Other Indicators

For the human evaluation on other indicators, we randomly sample 100 articles from Gigaword test set and ask 3 annotators to rate summaries of our systems and the baseline (BART), along with outputs by human-written summaries, showing in Table 9. We consider two types of unfaithful errors: (i) *hallucination error* (HErr.) and (ii) *logical error* (LErr.). We ask the annotators to label each type as 1 for existence of errors and 0 otherwise, and to score summaries on a Likert scale from 1 (worst) to 5 (best) on *informativeness* (Info.).

Informativeness It is the indicator reflecting whether the generated summary covers all important information points in the input text.

Logical Error The error for model of generating summaries whose logic structures contradicting with which in the original text (such as summarizing "A is B's dog" as "B is A's dog").

Hallucination Error The error for model of generating summaries containing the facts that are not in or cannot be inferred from original text.

Statistics	Articles	Ref.	Our Model
Avg Len	30.9	9.1	12.3

Table 8: Sentence-pair classification metrics of content selection phase.

Models	Info.↑	HErr.↓	LErr.↓
BART	3.76	12%	14%
Our Model	3.91	**7**%	**9**%
HUMAN	4.57	5%	2%

Table 9: Human evaluation on informativeness (Info.) (1-to-5), and hallucination error(HErr.) and logical error (LErr.) (0-to-1). **Bold** means it is significantly increased comparing to other models. ($p < 0.05$)