Extract, Select and Rewrite: A Modular Sentence Summarization Method

Shuo Guan
UBS AG
New York, NY 10010
shuo.guan@ubs.com

Vishakh Padmakumar
New York University
New York, NY 10012
vishakh@nyu.edu

Abstract

A modular approach has the advantage of being compositional and controllable, comparing to most end-to-end models. In this paper we propose Extract-Select-Rewrite (*ESR*), a three-phase abstractive sentence summarization method. We decompose summarization into three stages: (i) knowledge extraction, where we extract relation triples from the text using off-the-shelf tools; (ii) content selection, where a subset of triples are selected; and (iii) rewriting, where the selected triple are realized into natural language. Our results demonstrates that ESR is competitive with the best end-to-end models while being more faithful. Being modular, ESR's modules can be trained on separate data which is beneficial in low-resource settings and enhancing the style controllability on text generation.[1]

1 Introduction

While end-to-end models are dominating text generation tasks today, modular or pipelined approaches have the advantage of greater controllability and interpretability (Kedzie and McKeown, 2020). Prior work on abstractive summarization adopts a two-step process of first generating a plan (e.g., a semantic representation) of the target summary and then generating the summary conditioned on both the plan and the input document (Narayan et al., 2021, 2022). In this paper, we present a three-phase extract-select-rewrite pipeline, or ESR, for abstractive sentence summarization, where the plan is restricted to be a subset of knowledge triples extracted from the document. Specifically, we decompose the task into three subtasks: knowledge extraction, content selection and rewriting. To implement the three modules, we extract knowledge triples from the source document using off-the-shelf tools. Then, we train a classifier to select important triples representing content of the summary. Finally, we train a rewriter to convert the selected triples into natural language text (Figure 1).

There is extensive prior work that uses structured content extracted from the document to help summarization, such as relation triples (Cao et al., 2018), knowledge graphs (Zhu et al., 2021; Guan et al., 2021), and topics (Li et al., 2018, 2020; Aralikatte et al., 2021). However, these methods typically augment the source document with the extracted information and still learn to generate reference summaries from it in an *end-to-end* manner. By fully separating the modules during training, we can take a rewriter trained on a large dataset, and reuse it on a small target dataset while only training the content selector on as few as 1k examples.

We run experiments on Gigaword, DUC-2004 and Reddit-TIFU datasets and find that our approach produces summaries that are competitive to the end-to-end models in terms of automatic metrics. We also observe that a rewriter module trained on Gigaword, in the news domain, can be paired with a content selector trained on 1000 examples from Reddit-TIFU, a social media dataset, to produce high quality summaries, demonstrating the value of modularity in abstractive summarization.

Further, since our content planning is extractive in nature the summaries generated are also more faithful to the source as evidenced by a human evaluation comparing summaries from our modular approach and an end-to-end BART baseline. Lastly, We also observe that the rewriter module can be trained once on standalone text, which can enhance the controllability on the summary text generation style with minor changes of the training process.

2 Related Work

Knowledge-based Summarization Existing methods that use knowledge in summarization encodes it together with the input, e.g., Ribeiro et al. (2020) and Guan et al. (2021) introduce knowledge

[1]The codes are available on https://github.com/SeanG-325/ESR.

Proceedings of The 4th New Frontiers in Summarization Workshop, pages 41–48

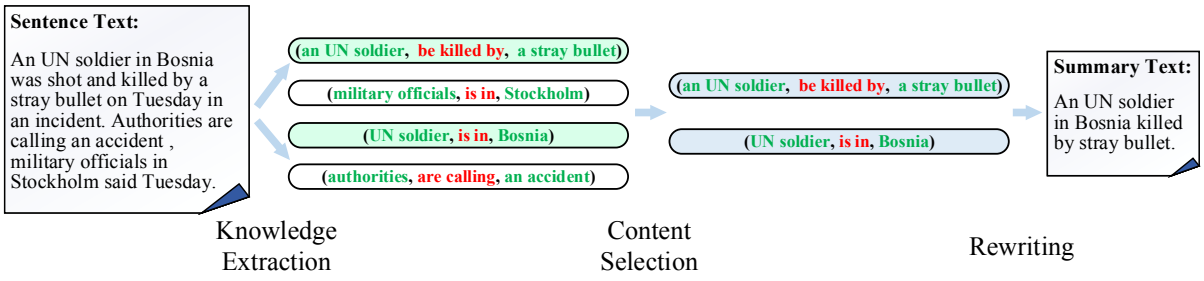

Figure 1: An overview of the three-phase summarization framework ESR.

graph encoding strategies for the graph-to-text generation model. Koncel-Kedziorski et al. (2019) and Wu et al. (2021) use a graph transformer encoder to consume knowledge and semantic graph. Huang et al. (2020) propose a model integrated with the GAT (Veličković et al., 2018) encoding knowledge graphs of the documents.

Modular Summarization Castro Ferreira et al. (2019) and Khot et al. (2021) showed the advantages of the modularity on text generation and question answering comparing to the end-to-end models. Pilault et al. (2020) and Chen and Bansal (2018) first extract sentences from the document and then perform abstractive summarization on them. Krishna et al. (2021) proposed a medical text generation method using modular summarization techniques based on clustering of utterances in sentences. However, the "modularity" in these methods rely on the neural networks to take in additional knowledge such as knowledge graphs, as opposed to splitting the model into different modules explicitly, which is where ESR differs.

3 Method

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 and rewriting, we turn all documents into structured content representation. We adopt knowledge triples that can be extracted by off-the-shelf tools (Section 4.1). The knowledge triples are in the form of `<entity 1, relation, entity 2>`. The extractors usually generate a large number of redundant triples (i.e. triples with large overlap with each other.[2] To delete the overlapping things, we use the Jaccard distance on n-grams (J_{Uni}, J_{Bi}) of between any pairs of triples (x_1, x_2) to calculate their similarity:

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

Here λ_1, λ_2 are hyperparameters determined on the validation data. We filter triples such that no pair of triples has a similarity score higher than the threshold. If the similarity between two tripes are larger than the threshold, the triple that has the larger length will be kept. The details of the threshold are in Section 4.1.

Content Selection The content selector selects the triples that are to be included in the summary out of the candidates. We train it as a sentence-pair classifier with two inputs, the *document* and the *candidate knowledge triple* extracted from it, and an output of whether to select the triple. If the triple is to be included in the summary of the document, the document-triple pair will be labeled positive, otherwise negative. We need to obtain supervised labels for the triples in the training set for training the content selector. For each triple in the training set, we use ROUGE (Lin, 2004) to measure the similarity to the corresponding summaries, if it is higher than a threshold then we label that triple as a positive example. Some representative examples of these sentence pairs and the details for selecting the threshold can be found in Section 4.1.

Rewriting The rewriter converts the selected triples into fluent summaries, where the triples serve as a content plan. We train a sequence-to-sequence text generation model, similar to converting meaning-representation to natural language text (Kedzie and McKeown, 2020). The train data for this phase contains the texts and the triples extracted from them. To train the generation model,

[2]For example, given the sentence "German chemical giant Hoechst Group announced plans wednesday to invest over a million dollars in China next year" our extractors might generate two candidates `<German chemical giant Hoechst Group, announced, plans>` and `<chemical giant Hoechst group, announced, plans>` which are clearly redundant.

	Ext.	Valid	Redun.	Pos/Neg
Train Articles	6.34	2.53	60.1%	0.91
Train Summaries	4.51	1.76	62.0%	-
Test Articles	6.19	2.42	60.9%	-

Table 1: Triple statistics in train and test sets. "*Ext.*" (Extracted) and "*Valid*" are the mean numbers of the the extracted and valid triplets (redundance removed). "*Redun.*" is the redundancy rate. "*Pos/Neg*" is the positive and negative sample ratio of the constructed data set in the content selection phase.

we concatenate the extracted triples from the document as the source sequence, and use the text as the target sequence. Note that training the rewriter only requires a piece of text and knowledge triples extracted from it. Therefore it can be potentially trained on much larger data (like Wiki text).

4 Experiments

4.1 Experiment Settings

Datasets Our main results are based on 2 news summarization datasets: (i) the Gigaword corpus (Rush et al., 2015), with around 3.8M summaries of single sentence news documents; (ii) DUC-2004, another test set in the news domain (Over et al., 2007)[3] To evaluate the modularity of our method, we reuse the rewriter trained on Gigaword and pair it with the content selector trained on another dataset from a different domain, Reddit TIFU (Kim et al., 2019); Gigaword contains news text while Reddit TIFU contains text from social media.

Training Details We used OLLIE (Mausam et al., 2012), two OpenIE tools (Angeli et al., 2015; Saha and Mausam, 2018) as the triple extractors. The triples from each of these are combined and then filtered for redundancy (Section 3). In order to ensure the quality of the triplet to the greatest extent, the methods such as co-reference resolution will be required. We fine-tuned the RoBERTa-large (Liu et al., 2019) as the content selector and fine-tuned the BART-large (Lewis et al., 2020) from `fairseq` (Ott et al., 2019) as the rewriter. All models are trained and fine-tuned on 2 NVIDIA RTX 2080 Ti GPUs. The detailed hyperparameters for three modules are in Appendix B.

4.2 Results

Intrinsic Evaluation of Each Module We first evaluate each of the three modules separately. Table 1 shows the detailed statistics of knowledge extraction based on Gigaword. The number of sentence-triple pairs is 400k, which are used to train the content selector. The accuracy of our fine-tuned RoBERTa content selector on this dataset is 88.9%. The details of the metrics are in Appendix Table 6. The size of the rewriting data set is 2M. We ablate the effect of the rewriting phase by comparing ROUGE scores before and after rewriting the triples in Appendix Table 7.

Model	R-1	R-2	R-L
BART (2020)	37.28	18.58	34.53
BART-RXF (2021)	40.45	20.69	36.56
PEGASUS+Dot (2021)	40.60	**21.00**	37.00
OFA (2022)	39.81	20.66	37.11
ESR	**40.63**	20.62	**37.14**

Table 2: ROUGE F1 on the Gigaword testset. It shows that ESR achieves or is competitive with the state-of-the-art on this dataset. **Bold** indicates the best score.

Model	R-1	R-2	R-L
RT+Conv (2018)	31.15	10.85	27.68
BART (2020)	31.36	11.40	28.02
ALONE (2020)	32.57	11.63	28.24
WDROP (2021)	33.06	11.45	28.51
ESR	**33.08**	**11.52**	**28.74**

Table 3: ROUGE F1 on DUC-2004 dataset. It shows ESR's performance achieved the SOTA on this dataset. **Bold** indicates the best score.

Automatic Evaluation Next, we evaluate the whole system on new summarization datasets. We report ROUGE score (Lin, 2004) on the Gigaword test set and the DUC-2004 dataset, containing 1951 and 500 samples respectively. We compare our ESR to a BART baseline that is fine-tuned in a single supervised step to generate the summary from the source documents. and some other strong models on the datasets.[4] The performance is shown in Table 2 and Table 3. On Gigaword and DUC2004, our approach outperforms the BART baseline and is within half the point of the SOTA results.[5]

Modularity One advantage of ESR is that training the rewriter does not require document-summary pairs and we can train it on any generic text. To test the modularity of ESR, we report the ROUGE on Reddit TIFU reusing a rewriter trained

[3]We use the DUC 2004 Task 1 which requires you to generate a sentence summary to a short article.

[4]These are typically modified variants of end-to-end models.We report the results from the PapersWithCode leaderboard and cite the corresponding works in the results table.

[5]State-of-the-art as of the date of submission per the leaderboard on PapersWithCode

Model	R-1	R-2	R-L
BART (2020)	24.19	8.12	21.31
PEGASUS+Sum (2022)	29.83	9.50	23.47
BART-R3F (2021)	30.31	**10.98**	24.74
ESR			
$S_R + R_G$	**30.63**	*10.82*	**24.78**
$S_R + R_R$	29.92	10.51	24.26
$S_{R1k} + R_{G1k}$	29.67	10.09	24.00
$S_{R1k} + R_{R1k}$	29.38	10.02	23.90
$S_{R1k} + R_G$	29.09	10.07	23.86

Table 4: ROUGE F1 on R-TIFU (Reddit-TIFU). S_R means the content selector was trained on R-TIFU, R_G and R_R mean rewriter trained on Gigaword and R-TIFU respectively. 1k means that the module is trained on 1k randomly sampled subset. The content selector can be trained with low-resourced data without large dropping. **Bold** means the best and *Italics* means the best in ESR.

on Gigaword in Table 4. The best ROUGE is obtained when using the Reddit TIFU content selector coupled with the Gigaword rewriter, highlighting the benefit of training the modules separately. One advantage of such decoupling is that we can train the rewriter on high resource domains and reuse it in low resource 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. We see that training a content selector on only 1k examples and reusing the rewriter from Gigaword is on-par with using the entire Reddit TIFU. Further, the modularity makes ESR able to control the text style, as in Figure 2.

Human Evaluation We conducted a user study on Amazon MTurk where annotators rated summaries of 100 randomly sampled texts from the Gigaword test set on faithfulness. We asked the annotators to rate summaries of our approach and BART, together with the results of the gold summaries of the data set. Each crowdworker was shown the source document and three summaries and asked to decide if each summary is individually supported by the text in the source. We collect three annotations for each example and decide the judgement via a majority vote. It is labeled inconclusive if there is no agreement. The results are in Table 5. We see that ESR is rated to be more faithful than the baseline and almost as good as the human-written summaries. A representative case is shown in Figure 2. It shows that ESR can eliminate the hallucination and control the summarization styles with different rewriter modules.

Summaries	Sup.	Unsup.	Incoh.	Inconc.
Human-Written	96	3	0	1
BART	90	6	2	2
ESR	94	3	2	1

Table 5: Human evaluation on faithfulness. The summaries from the dataset (Human-Written) and those from ESR and the BART are annotated by 3 annotators. Crowd workers find ESR to be more faithful than BART.

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)
Ref: Zairean president Mobutu to stay in France till mid-week

BART: **Tanzania's** Mobutu to stay at **Riviera residence** until middle of week

ESR (Gigaword content selector):
- **Gigaword rewriter:** **Zairean president Mobutu** will stay at his **French Riviera residence** until the middle of week
- **Reddit-TIFU rewriter:** **Zairean Mobutu** will stay at his **French Riviera president residence**... it's said that he will stay until the middle of week

Figure 2: A case on the Gigaword testset. **ST**: source text; **Ref**: reference summary; **Selected Triples**: triples selected by the content selector. With the rewriter module trained on different dataets, the text style of ESR can be controlled. The green shows the factual correctness and the red shows the error.

4.3 Analysis

The evaluations show that ESR can achieve or approach SOTA performance on multiple datasets and can enhance the faithfulness of summaries. We found ESR can limit the content of the generated summary in the content selection stage, and then rewrite only selected content. Therefore, text generation will introduce less hallucination. In addition, ESR has better modularity than other models, as the selector and rewriter can be trained separately on different data to enhance performance and controllability on summarization. This means that we can modify the modules to enhance performance rather than redesign the entire framework.

5 Conclusion

We propose ESR, a three-phase modular abstractive summarization method. It obtains competitive performance on automatic metrics while producing more faithful summaries, and its modularity makes it have a good controllability on summary generation, and maintains a good performance on

low resource data. In the future, we are adapting the ESR method to multi-document summarization datasets.

Acknowledgement

We would like to appreciate Professor He He for her input and guidance at various stages of the project. This work is supported by the Samsung Advanced Institute of Technology (Next Generation Deep Learning: From Pattern Recognition to AI) and the National Science Foundation under Grant No.1922658. The computation resource of the work is supported by NYU Courant Institute of Mathematical Sciences.

Limitation

One limitation of our method is the reliance on off-the-shelf tools for the extraction phase. These tools are sometimes not able to successfully obtain triples from the source sentences, which results in empty summaries, and at others they returns multiple redundant candidates which makes selection challenging. We attempt to address the former by aggregating results from multiple extractors and the latter by filtering candidates through overlap based heuristics.

Ethical Consideration

One ethical consideration for the modular summarization method is that we are essentially using two different deep learning steps, content selection followed by text generation. There is a chance for model bias to have an impact at either stage. Additionally, we note that one of the features of modular summarization is that different applications can select different content to be relevant to a summary. Improper content selection here could exacerbate issues such as misinformation when used in real-world applications. We do however note that this is not isolated to our modular summarization approach, but is also the case even when the model is learned end-to-end.

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.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact-aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562, Hong Kong, China. Association for Computational Linguistics.

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.

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.

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, Online. Association for Computational Linguistics.

Akhil Kedia, Sai Chetan Chinthakindi, and Wonho Ryu. 2021. Beyond reptile: Meta-learned dot-product maximization between gradients for improved single-task regularization. In *Findings of the*

Association for Computational Linguistics: EMNLP 2021, pages 407–420, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chris Kedzie and Kathleen McKeown. 2020. Controllable meaning representation to text generation: Linearization and data augmentation strategies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5160–5185, Online. Association for Computational Linguistics.

Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2021. Text modular networks: Learning to decompose tasks in the language of existing models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1264–1279, Online. 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.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. 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 2284–2293, 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.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 55–60, New Orleans, Louisiana. Association for Computational Linguistics.

Haoran Li, Junnan Zhu, Jiajun Zhang, Chengqing Zong, and Xiaodong He. 2020. Keywords-guided abstractive sentence summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8196–8203.

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. 2019. 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.

Shashi Narayan, Gonçalo Simões, Yao Zhao, Joshua Maynez, Dipanjan Das, Michael Collins, and Mirella Lapata. 2022. A well-composed text is half done! composition sampling for diverse conditional generation. *arXiv preprint arXiv:2203.15108*.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

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.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.

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.

Mathieu Ravaut, Shafiq Joty, and Nancy Chen. 2022. SummaReranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4504–4524, Dublin, Ireland. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling global and local node contexts for text generation from knowledge graphs. *Transactions of the Association for Computational Linguistics*, 8:589–604.

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.

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lió, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

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.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *CoRR*, abs/2202.03052.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu, and Haifeng Wang. 2021. BASS: Boosting abstractive summarization with unified semantic graph. 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 6052–6067, Online. Association for Computational Linguistics.

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

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

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.

B Hyper Parameters

The hyper parameters for fine-tuning RoBERTa-large in content selection phase, and BART-large model in rewriting phase are listed.

B.1 Knowledge Extraction

The hyperparameters in Jaccard similarity are $\lambda_1 = 0.75$ and $\lambda_2 = 0.25$. The threshold for similarity is 0.7.

B.2 Content Selection

TOTAL_NUM_UPDATES=3000
WARMUP_UPDATES=500
LR=1e-05
NUM_CLASSES=2
MAX_SENTENCES=8

Acc.	Rec.	Prec.	F1
88.9%	88.6%	88.1%	88.4%

Table 6: Sentence-pair classification performance of the content selector.

B.3 Rewriting

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

	R-1	R-2	R-L
Concatenated Triples	38.98	18.12	35.76
Rewritten Summaries	40.63	20.62	36.71

Table 7: ROUGE comparing Concatenated Triples (aren't rewritten) and Rewritten Summaries (rewritten).

Statistics	Articles	Ref.	Our Model
Avg Len	30.9	9.1	12.3

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