The CreativeSumm 2022 Shared Task: A Two-Stage Summarization Model using Scene Attributes

EunChong Kim* **TaeWoo Yoo*** **GunHee Cho** **SuYoung Bae** **Yun-Gyung Cheong**
Department of Artificial Intelligence
Sungkyunkwan University, South Korea
{prokkec,woo990307,skate4333,sybae01,aimecca}@skku.edu

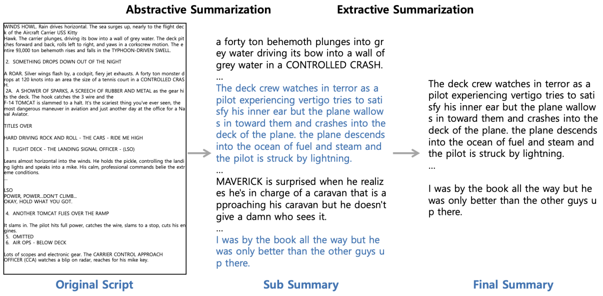
Abstract

In this paper, we describe our work for the CreativeSumm 2022 Shared Task, Automatic Summarization for Creative Writing. The task is to summarize movie scripts, which is challenging due to their long length and complex format. To tackle this problem, we present a two-stage summarization approach using both the abstractive and an extractive summarization methods. In addition, we preprocess the script to enhance summarization performance. The results of our experiment demonstrate that the presented approach outperforms baseline models in terms of standard summarization evaluation metrics.

1 Introduction

Summarization is an important task in natural language processing research area. Although many works have been conducted on summarization, little has been researched on summarizing movie scripts. It is challenging to generate a summary from movie scripts for several reasons. First, movie scripts are long. According to an analysis on Hollywood screenplays Snyder (2005), a typical script has an average page count of 110, and can even reach a few hundred pages. The long input length causes disparities when aligning its plot summary with corresponding parts in the script (Mirza et al., 2021).

Summarizing long and multi-faceted document is a classical challenge. As the document gets longer, the computational complexity of summarizing it increases dramatically (Gidiotis and Tsoumakas, 2020). Attempts have been made to address the problem of long document summarization, utilizing discourse in the document (Cohan et al., 2018) and hierarchical structures to effectively understand long sentences (Grail et al., 2021). Cohan et al. (2018) recognizes the importance of discourse in long document summarization. They develop a discourse-aware decoder to capture important points from different discourse sections. Liu et al. (2018) presents a two-stage strategy. They select important sentences in a document using an extractive summarization model, and then summarize them again using a transformer decoder.



WINDS HOWL. Rain drives horizontal. The sea surges up, nearly to the flight dec
k of the Aircraft Carrier USS Kitty
Hawk. The carrier plunges, driving its bow into a wall of grey water. The deck pit
ches forward and back, rolls left to right, and yaws in a corkscrew motion. The e
ntire 93,000 ton behemoth rises and falls in the TYPHOON-DRIVEN SWELL.

2. SOMETHING DROPS DOWN OUT OF THE NIGHT

A ROAR. Silver wings flash by, a cockpit, fiery jet exhausts. A forty ton monster d
rops at 120 knots into an area the size of a tennis court in a CONTROLLED CRAS
H.
2A. A SHOWER OF SPARKS, A SCREECH OF RUBBER AND METAL as the gear hi
ts the deck. The hook catches the 3 wire and the
F-14 TOMCAT is slammed to a halt. It's the scariest thing you've ever seen, the
most dangerous maneuver in aviation and just another day at the office for a Na
val Aviator.

TITLES OVER

HARD DRIVING ROCK AND ROLL - THE CARS - RIDE ME HIGH

3. FLIGHT DECK - THE LANDING SIGNAL OFFICER - (LSO)

Leans almost horizontal into the winds. He holds the pickle, controlling the landi
ng lights and speaks into a mike. His calm, professional commands belie the extr
eme conditions.
...

LSO
POWER, POWER...DON'T CLIMB...
OKAY, HOLD WHAT YOU GOT.

4. ANOTHER TOMCAT FLIES OVER THE RAMP

It slams in. The pilot hits full power, catches the wire, slams to a stop, cuts his en
gines.
5. OMITTED
6. AIR OPS - BELOW DECK

Lots of scopes and electronic gear. The CARRIER CONTROL APPROACH
OFFICER (CCA) watches a blip on radar, reaches for his mike key.

a forty ton behemoth plunges into gr
ey water driving its bow into a wall of
grey water in a CONTROLLED CRASH.
...
The deck crew watches in terror as a
pilot experiencing vertigo tries to sati
sfy his inner ear but the plane wallow
s in toward them and crashes into the
deck of the plane. the plane descends
into the ocean of fuel and steam and
the pilot is struck by lightning.
...
MAVERICK is surprised when he realiz
es he's in charge of a caravan that is a
pproaching his caravan but he doesn't
give a damn who sees it.
...
I was by the book all the way but he
was only better than the other guys u
p there.

The deck crew watches in terror as a
pilot experiencing vertigo tries to sati
sfy his inner ear but the plane wallow
s in toward them and crashes into the
deck of the plane. the plane descends
into the ocean of fuel and steam and
the pilot is struck by lightning.
...

I was by the book all the way but he
was only better than the other guys u
p there.

Figure 1: A sample illustration of our two-stage summarization approach with the movie script of the film 'Top Gun'. We first create a scene summary from the script using an abstractive summarization model. We then select important sentences using an extractive summarization model.

Moreover, movie scripts have a complex format, consisting of different components such as dialogues, action directions, scene description, cut transitions, film editing instructions, etc (Feng et al., 2021). Movie transcripts are similar to TV scripts, in that they contain dialogues as well as action and filming directions and editing descriptions. Various studies have been conducted to extract summaries from TV scripts (Liu et al., 2021). Zhong et al. (2022) constructs a dialogue summarization model trained with TV transcripts datasets by crawling them from the websites such as Forever Dreaming and TVMegaSite dataset (Chen et al., 2021a). Although movie scripts tend to be longer than TV scripts, we can refer to previous studies in TV script summarization to deal with the complex format.

Summarization methods can be divided into two types: abstractive and extractive summarization. When a document is given as a source text, an abstractive summarization generates a summary that

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 51–56
July 5–10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.26615/978-954-452-056-4_008

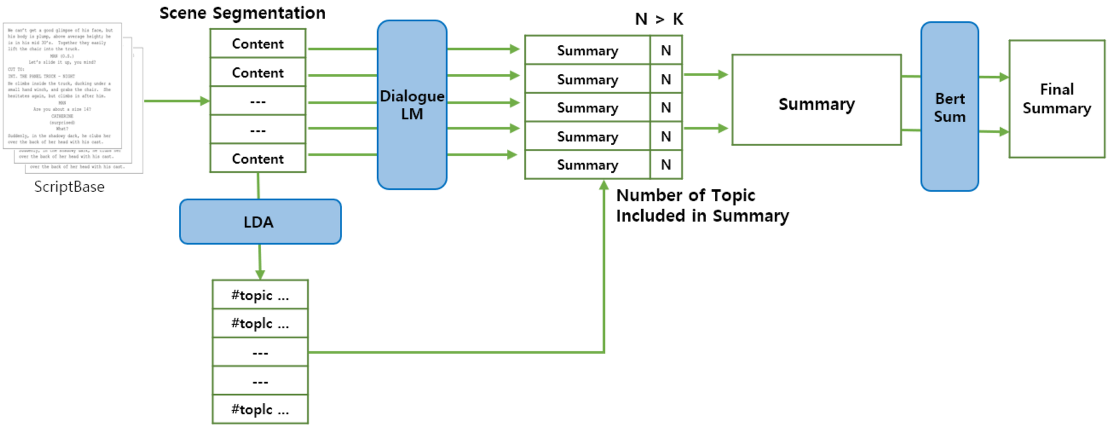


Figure 2: Framework of our model contains scene segmentation, abstractive summarization using Dialogue LM, important scene selection, and extractive summarization using BERTSum.

contains text that are not in the source text, whereas extractive summarization re-uses the words in the source text. Specifically, the task of extractive summarization is choosing salient words and sentences.

The goal of our work is to summarize movie scripts, which is one of the shared task of CreativeSumm 2022, the Automatic Summarization for Creative Writing workshop, at COLING 2022. This paper describes a two-stage summarization approach employing both the abstractive and extractive summarization methods to summarize movie scripts.

Figure 1 illustrates our framework. First, our preprocessing stage merges similar scenes based on character information to enhance the summarization performance. Then, the abstractive summarization method produces a summary for each scene-based unit of the script. In particular, we use the DialogLM model (Zhong et al., 2022) since dialogues in the script are essential for understanding the story. Finally, these scene summaries are summarized again by selecting salient scenes via topic modeling and the extractive summarization method.

This paper is structured as follows. Section 2 describes our method in detail. Section 3 reports the results of the experiment. Finally, we end with conclusion.

2 Method

Our approach is composed of four steps: scene segmentation, dialogue abstraction, salient scene selection, and extractive summarization. Figure 2 briefly illustrate our method.

2.1 Scene Segmentation

Before putting the script into the abstract summarization model, we preprocess the script to group scenes with similar context. Generally, one sentence of a plot summary can be mapped to several scenes in the script (Mirza et al., 2021). Hence, we first divide the script into scenes, using scene headings that describe the location and time of the day of a scene, such as ’INT’, ’EXT’, ’-DAY’, and ’-NIGHT’, following (Mirza et al., 2021).

Then, we group a number of scene based on the main characters, as illustrated in Figure 3. First, we identify main characters based on the number of scenes they appear. For this, we set the total number of scenes as the threshold value. In the example, the script contains 100 scenes which serves as the threshold. We count the number of scenes that a character appears. Starting from the highest value, we accumulate the numbers of character appearance until their sum exceeds the threshold. As a result, Woody, Buzz, and Andy are identified as the main characters. We classify the characters contributing to the summation as main characters. Therefore, the main characters in our approach can be different from the actual main characters in the movie.

Then, we merge subsequent scenes where their main characters are identical. For instance, scenes 10 and 11 are merged since they share Woody and Buzz as the main characters, however, scenes 11 and 12 are not merged since Woody is not present in scene 12.

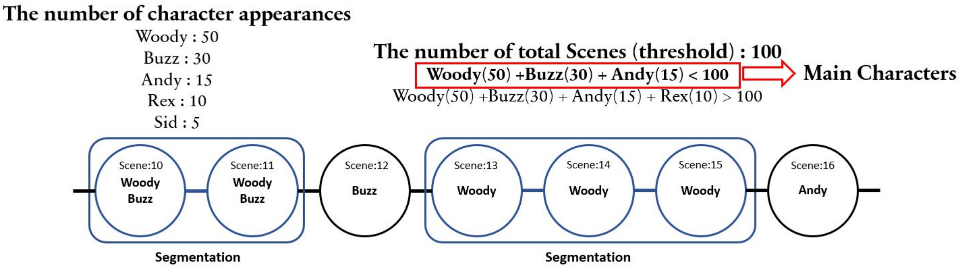
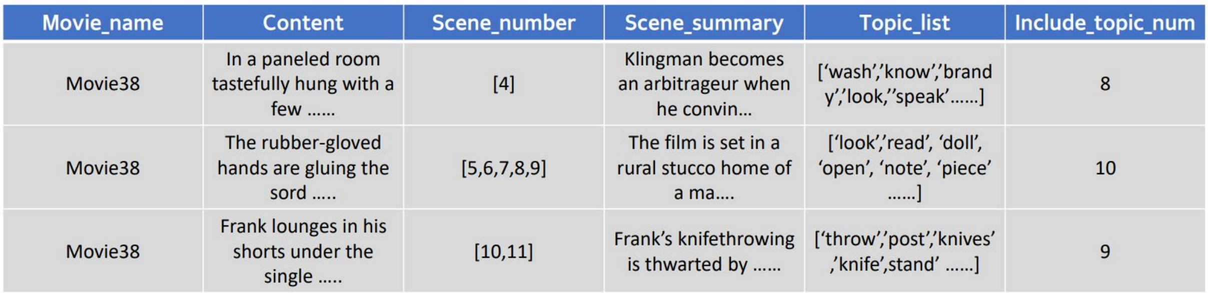


Figure 3: An example of scene merging process. The first step identifies main characters. The second step merges the consecutive scenes into one if their main characters are identical.



Movie_name	Content	Scene_number	Scene_summary	Topic_list	Include_topic_num
Movie38	In a paneled room tastefully hung with a few	[4]	Klingman becomes an arbitrageur when he convin...	['wash','know','brand y','look,''speak'......]	8
Movie38	The rubber-gloved hands are gluing the sord	[5,6,7,8,9]	The film is set in a rural stucco home of a ma....	['look','read', 'doll', 'open', 'note', 'piece' ......]	10
Movie38	Frank lounges in his shorts under the single	[10,11]	Frank's knifethrowing is thwarted by	['throw','post','knives' ,'knife',stand' ......]	9

Figure 4: An example of important scene selection using topic modeling. We compute the scene's salience score as the number of keywords that are associated with the scene's main topic. In this case, the summary of the scenes [5,6,7,8] scores highest.

2.2 Dialogue Abstraction

This step uses an abstractive summarization model to summarize a scene. In a script, a scene has dialogues between characters mixed with various information such as action direction and film editing instructions. Since dialogues are essential for story comprehension, we believe dialogue summarization models are appropriate for abstracting a scene.

Various studies have been conducted on dialogue summarization tasks (Gliwa et al., 2019; Chen et al., 2021b; Feng et al., 2022): a study on dialogue summarization using a graph with topic words (Zhao et al., 2020), a study using a model with sparse attention technology and pre-training with a new masking skill to improve dialogue summarization performance (Zhong et al., 2022), etc.

In this study, we apply the abstractive summarization model DialogLM (Zhong et al., 2022) to each scene to generate a scene summary. We did not eliminate other supplemental components such as film editing instructions and action descriptions. In our work, we observe that a scene summary typically consists of 80 tokens, while the maximum length of a summary is set to 280 tokens.

2.3 Important Scene Selection

The previous step creates scene summaries. Since not all the scenes contribute equally to story comprehension, we select important scenes using a topic modeling approach. We leverage LDA to find topics that are associated with a particular scene summary where individual topic is also associated with N(= 10 in our work) topic. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is a probabilistic topic modeling method that infers latent topics from a corpus of documents.

We select a single keyword from each scene summary with the highest proportion as its main topic. For example, if a scene summary is associated with three topics, such that topic 1 occupies 74%, topic 2 occupies 20%, and topic 3 occupies 6% of the summary, we select topic 1 as the main topic of that scene. We compute the salience score of the scene summary based on the number of keywords that are associated with the scene's main topic (see Figure 4). If these keywords appear in a scene summary less than the pre-defined threshold value, we eliminate the summary from the scene summaries set.

Table 1: Results of experiments to test the impact of scene merging and replacing extractive summary with abstractive summary. AS denotes abstracive summarization, and ES denotes extractive summarization. The best performance is shown in bold.

	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore-P	BERTScore-R	BERTScore-F1
AS + AS (w.o scene merge)	0.3003	0.0490	0.1311	0.6185	0.6611	0.6385
AS + AS	0.3226	0.0578	0.1375	0.6434	0.6643	0.6488
AS + ES (w.o scene merge)	0.3975	0.0788	0.1529	0.6780	0.6932	0.6854
AS + ES	**0.4010**	**0.0788**	**0.1580**	**0.6835**	**0.6953**	**0.6892**

2.4 Extractive Summarization

At this stage, only important scene summaries remain. The final step leverages an extractive summarization model to create the final summary of the film.

We use the BERTSum model(Liu, 2019), which uses BERT(Devlin et al., 2018) as the embedding model. In BERTSum, the input document is encoded with multiple sentences and then used as the input for BERT. Then, use the output of the BERT as the input to the summarization layers of the Transformer 2-layers. Finally, Using the sigmoid function to classify each sentence as class 0 or class 1. The scene summaries are given as input to BERTSum, and only the scene summaries classified as class 1 are included in the final summary.

3 Evaluation

3.1 Dataset

The goal of our work is to summarize movie scripts, which is one of the CreativeSumm 2022 shared tasks. We are provided with ScriptBase (Gorinski and Lapata, 2015), a collection of 1,276 movie scripts and their corresponding wikiplot summaries, as the training and development dataset. An additional dataset is provided as the test dataset for evaluating our approach using standard automatic evaluation metrics including ROUGE, BERTScore, LitePyramid, and SummaCZS.

3.2 Model selection

To find the best setting for our method, we conducted several experiments with various settings using 100 randomly selected movies from the dataset. First, we test whether the scene grouping method enhances the summarization performance or not. We simply remove scene segmentation process and check how this impacts the model performance. Table 1 shows that we obtained the best performance when using an extractive summarization model with the scene merging strategy.

Second, we test if the extractive summarization method can replace abstractive summarization. We use Primer (Xiao et al., 2021) as the abstractive summarization model which specializes summarizing long/multi documents. As described above, BertSum (Liu, 2019) is used as the extractive summarization model. Table 1 shows that using the abstractive model throughout the summarization process results in performance deterioration regardless of the scene merging strategy. Therefore, we use the scene merging strategy and the combination of abstractive and extractive summarizaion for the evaluation.

3.3 Model settings

We leverage DialogLM and BertSum models for abstractive and extractive summarization respectively. DialogLM, used in this experiment, is DialogLED-base-16384, a larger version of DialogLM. We fine-tune the pre-trained DialogLM model on the FD dataset (Chen et al., 2021a), which has transcripts of 88 TV shows. This model accepts up to 16,384 tokens as input and outputs a summary consisting of up to 280 tokens. We use the BertSum model that employs the Bert model pre-trained with the pytorch-pre-trained-BERT version. We constrain the summary length not to exceed 4500 tokens.

3.4 Results

Table 2 shows the evaluation metrics (ROUGE, BERTScore (Zhang et al., 2019), LitePyramid (Zhang and Bansal, 2021), and SummaC (Laban et al., 2021)) computed on the test dataset. The baseline model is LED (Beltagy et al., 2020), with variations of input size of 1024, 4096, and 16384.

The results indicate that our approach outperforms the baseline model in terms of the ROUGE and BERTScore metrics. BERTScore use BERT to calculate similarity score between candidate sentence and reference sentence in each token. We obtained 0.4144 for ROUGE-1, 0.0823 for ROUGE-2, and 0.3963 for ROUGE-3, achieving three times better results than the baseline model. Our ap-

Table 2: The evaluation metrics of the experiment on the test set. The subscript denotes input size. The best performance is shown in bold.

	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore-P	BERTScore-R	BERTScore-F1
LED$_{1024}$	0.1492	0.0146	0.1373	0.4298	0.4238	0.4258
LED$_{4096}$	0.1416	0.0130	0.1299	0.4245	0.4137	0.4179
LED$_{16384}$	0.1368	0.0125	0.1277	0.4322	0.3924	0.4099
Our approach	**0.4144**	**0.0823**	**0.3963**	**0.5163**	**0.5233**	**0.5194**

	LitePyramid-p^{2c}	LitePyramid-l^{2c}	LitePyramid-p^{3c}	LitePyramid-l^{3c}	SummaCZS
LED$_{1024}$	**0.3436**	0.3546	0.2517	0.2833	0.0000
LED$_{4096}$	0.3604	**0.3763**	**0.2674**	**0.3042**	0.0000
LED$_{16384}$	0.3009	0.3082	0.2312	0.2602	0.0000
Our approach	0.0370	0.0063	0.0356	0.0072	**0.0476**

	Length	Density	Coverage	Novel 1-grams	Novel 2-grams
LED$_{1024}$	903	1.1809	0.7021	**0.3357**	**0.7485**
LED$_{4096}$	877	1.3432	0.7311	0.3092	0.7273
LED$_{16384}$	551	1.4103	0.7266	0.3024	0.7211
Our approach	729	**3.4398**	**0.8759**	0.1621	0.4827

proach also outperforms the baseline models in terms of the density and coverage metrics. The density score denotes the average length of the extractive fragment in the summary, and the coverage score denotes how many words in the document are included in the summary (Grusky et al., 2018). Since we use an extractive summarization model to create the final summary, the Novel n-grams scores tend to be low (see Table 2).

Our approach underperforms for various LitePyramid metrics, which compare the reference with system summary using a natural language inference (NLI) model. But we get good score at SummaCZS, which compute NLI score between pairs of sentences from segmented document. Length denotes the average length of the summaries that the model produces.

4 Conclusion

In this paper, we present a two-stage approach for the shared task of Creative-Summ 2022. We first segment the script into scenes and create their summaries using an abstractive summarization model. Second, we apply topic modeling to eliminate less important scenes. Then, a BERT-based extractive summarization model generates the final summary of the movie. The result of evaluation indicates that our approach outperforms baseline models in several metrics. We got 0.4144 for ROUGE-1, 0.0823 for ROUGE-2, and 0.3963 for ROUGE-3 which are better than baseline models. We also got better BERTScore such as 0.5163 for precision, 0.5233 for recall and 0.5194 for F1 score.

Acknowledgments

This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2019R1A2C1006316), Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2019-0-00421, Artificial Intelligence Graduate School Program(Sungkyunkwan University)), and NCSoft.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2021a. Summscreen: A dataset for abstractive screenplay summarization. *CoRR*, abs/2104.07091.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021b. Dialogsum: A real-life scenario dialogue summarization dataset. *arXiv preprint arXiv:2105.06762*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. A survey on dialogue summarization: Recent advances and new frontiers. *arXiv preprint arXiv:2107.03175*.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2022. Msamsum: Towards benchmarking multi-lingual dialogue summarization. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 1–12.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Philip Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076.

Quentin Grail, Julien Perez, and Eric Gaussier. 2021. Globalizing bert-based transformer architectures for long document summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1792–1810.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2021. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *arXiv preprint arXiv:2111.09525*.

Junpeng Liu, Yanyan Zou, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Caixia Yuan, and Xiaojie Wang. 2021. Topic-aware contrastive learning for abstractive dialogue summarization. *arXiv preprint arXiv:2109.04994*.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Paramita Mirza, Mostafa Abouhamra, and Gerhard Weikum. 2021. Alignarr: Aligning narratives on movies. In *The 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 427–433. ACL.

Blake Snyder. 2005. Save the cat! the last book on screenwriting you'll ever need.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2021. Primer: Pyramid-based masked sentence pre-training for multi-document summarization. *arXiv preprint arXiv:2110.08499*.

Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. *arXiv preprint arXiv:2109.11503*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lulu Zhao, Weiran Xu, and Jun Guo. 2020. Improving abstractive dialogue summarization with graph structures and topic words. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 437–449.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11765–11773.