Long Input Dialogue Summarization with Sketch Supervision for Summarization of Primetime Television Transcripts

Nataliia Kees
inovex GmbH
nataliia.kees@inovex.de

Thien Quang Nguyen
Technical University of Munich
thien.nguyen@tum.de

Tobias Eder
Technical University of Munich
tobias.eder@in.tum.de

Georg Groh
Technical University of Munich
grohg@in.tum.de

Abstract

This paper presents our entry to the CreativeSumm 2022 shared task, tackling the problem of prime-time television screenplay summarization based on the SummScreen Forever Dreaming dataset. Our approach utilizes extended Longformers combined with sketch supervision including categories specifically for scene descriptions. Our system was able to produce the shortest summaries out of all submissions. While some problems with factual consistency still remain, the system was scoring highest among competitors in the ROUGE and BERTScore evaluation categories.

1 Introduction

This paper represents our submission to the CreativeSumm 2022 shared task, which itself was subdivided into four distinct summarization tasks and consists of summarization of (i) chapters from novels, (ii) movie scripts, (iii) prime time television scrips and (iv) day-time television scripts. Our system focused on summarizing prime-time television show transcripts for task (iii) and aims at producing a brief description for a single episode of its main developments based on the underlying episode script.

Our system has been trained on the SummScreen Forever Dreaming (FD) dataset (Chen et al., 2022) that was released as part of the shared task to produce automatic abstractive summaries of TV show screenplays. The structure of the data and its strong stylistic reliance on dialogues allows us to construct the problem of TV screenplay summarization as a dialogue summarization problem. To solve this, we use a dialogue summarization architecture of Abstractive Dialogue Summarization with Sketch Supervision introduced by Wu et al. (2021) as a base architecture and adjust it to process large inputs of up to 16384 tokens and handle scene descriptions, which is one important characteristic of this data and makes it different from typical dialogue data.

With our architecture, we achieve results which strongly outperform the baseline end-to-end models and result in better performance than our competitors on word- and context-based metrics on the CreativeSumm 2022 shared task.

2 Related Work

2.1 Dialogue Summarization

Existing research in dialogue summarization is of high relevance for our task, since we approach the task of summarizing TV screenplay transcripts as dialogue summarization and are interested in state-of-the-art methods which would be suitable for the given dataset. In this chapter, we present a few approaches we considered for using to solve the task of television screenplay summarization.

There have been several works achieving some progress in producing dialogue summaries (e.g. Chen and Yang, 2020, Liu et al., 2021, Wu et al., 2021, Liu and Chen, 2021, Zou et al., 2021, Park and Lee, 2022). Because dialogues, especially in the context of television show scripts, can be analyzed from different perspectives (e.g. topics they cover or order of the utterances, or stages of the discussion they represent), an interesting approach in this regard has been presented by Chen and Yang (2020). They model conversations via these different standpoints, which they call *views*, by incorporating different structures a conversation can consist of, and use those for summarization. Chen and Yang (2020) distinguish between different views, which focus on a specific aspect of each speaker's intent. For instance, one view would structure the conversation around topics, whereas another view would focus on the order of the utterances. The authors segment conversations according to the views into single blocks of utterances. That way, they extract relevant information for different contexts and intents in order to generate summaries. For the summary generation they apply a conversation encoder

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 29–35
July 5–10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.26615/978-954-452-056-4_005

consisting of a combination of BART (Lewis et al., 2019) and LSTM layers and a multi-view decoder which is built with transformer layers (Vaswani et al., 2017), combining the views.

A specific characteristic of dialogues is the fact that all relevant information is scattered across utterances and it might be hard to connect the pieces of the discourse in an automated way. To tackle the scatteredness of information across all utterances, Liu et al. (2021) use the notion of *coreferences*, which they use to gather relevant parts of information across multiple segments in a conversation. They introduce a dataset with annotations of the coreferences, which rely on coreference resolution models, and use graph convolutional neural networks on graph representations of the conversations and their coreferences. To obtain a contextualized representation of the nodes, the authors introduce coreference-guided self-attention to the coreference information.

Another approach, which aims at tackling both the scatteredness of information and the distinctions between single logical blocks of the discourse flow in a dialogue, is the *Controllable Abstractive Dialogue Summarization (CODS)* architecture by Wu et al. (2021). In addition to that, it aims at controlling the length of the output summary, which is particularly interesting in the context of long dialogue summarization, where the possibilities with regard to the output summary length are broad. The approach envisions the generation of a summary sketch of the dialogue and using a segmentation model to control the amount of sentences the model generates for the summary (Wu et al., 2021). The sketch contains only the most relevant information, therefore excluding non-factual sentences. It outlines the intents of each speaker and contains the key phrases representing these intents. It also functions as a weakly supervised signal for the model. The length control takes place via predicting the text span cutoffs, which lead to multiple segments as an output. The more sentences the model is supposed to generate, the more segments will be extracted. Of all three architectures, Wu et al. (2021) report for their CODS model the highest performance on SAMSum data (Gliwa et al., 2019)[1]. This, along with the convenient use of the intent model, which can be expanded easily, is the reason why we make use of this model as a basis for our experiments, adjusting it to our data as described in Chapter 4. Though, we do not use the segmentation model in our experiments, as we want to keep the architecture simple and reduce the environmental footprint of the training process. As the authors report, removing the segmentation influences the performance on the SAMSum dataset only marginally (ROUGE-1 F1 of 51.79 for sketch supervision vs. 52.65 for the full CODS approach (Wu et al., 2021)), which can be seen as negligible.

2.2 Summarization of Television Show Transcripts

The SummScreen-FD dataset is a summarization dataset which provides pairs of TV series transcripts and human-written summaries (Chen et al., 2022). One of the challenges it poses is its long input size (see Chapter 3 for details), which requires special treatment due to high computation complexity of typical transformer models.

Zhang et al. (2021) compare several methods on the SummScreen-FD dataset, such as BART (Lewis et al., 2019) with input length of 1024 tokens, HMNet (Zhu et al., 2020), which is a hierarchical model for dialogue summarization, with input size of 8192 tokens, as well as Longformer Encoder-Decoder (Beltagy et al., 2020) with input size of 4096 tokens. They also compare these to a retrieve-then-summarize pipeline based on TF-IDF, BM25 or Locator (Zhong et al., 2021) retriever and arrive at the conclusion that a BART-large model (Lewis et al., 2019) pretrained on CNN/DM dataset yields the best performance on SummScreen-FD, achieving a ROUGE-1 F1 score of 28.86.

Zhang et al. (2022) approach summarization of long-input dialogues by using a greedy segment-then-combine method for compressing the inputs and use two summarizers based on BART (Lewis et al., 2019): one produces coarse summaries and another one (with different parameters) finegrains the coarse summarizes to produce the final outputs. They report reaching 32.48 in terms of ROUGE-1 score on SummScreen-FD data.

Because of the potentially different data splits that the authors of the above models have used for their evaluation, which most probably do not correspond to the blind test set the CreativeSumm workshop participants received for the shared task, these numbers are not directly comparable with our evaluation results reported in Chapter 5. However, they are useful as an orientation of the minimal

[1] Reported ROUGE-1 F1 score of 52.65 for CODS (Wu et al., 2021) compared to 50.9 for Coref (Liu et al., 2021) and 49.3 for MultiView (Chen and Yang, 2020)

expected performance for our selected approach.

3 Dataset

The SummScreen dataset contains television show screenplays – one screenplay per episode – and human-written summaries which provide a short description of the story line of the given episode (Chen et al., 2022). It consists of two parts: SummScreen Forever Dreaming (FD), which contains prime TV shows screenplays, and SummScreen TV Megasite (TMS), consisting of daytime TV show screenplays.

In this work, we focus on SummScreen-FD as our data source for model training and evaluation. Our training data contains 4008 instances of TV series episodes with corresponding hand-written summaries as gold labels, for validation we use additional 337 instances and for testing 459 examples. An example snippet from the dataset can be found in Table 1.

An important characteristic of this data is that the input length of the screenplays much exceeds the typical input length that standard Transformer-based language models are designed to tackle: in terms of word count our train data, for example, has on average 7587 words, with a smallest screenplay having 1934 words and the longest one 21435 words. Outputs in the training data are much shorter, with average of 111 words, where the longest summary has 822 words and the shortest summary being only 8 words long. This requires an approach which would be capable of processing large inputs, grasping temporal relations and references on long distances, and squeezing them into concise outputs.

Such large input size can in part be attributed to another characteristic of SummScreen-FD, which makes it different from SummScreen-TMS, which is that it also contains descriptions about environments or characters and their feelings, similar to scene setting descriptions. We exploit this characteristic in our method, as we describe in more detail in Chapter 4.

4 Method

To tackle the task of summarizing TV show screenplays, we construct it as a dialogue summarization problem and use a dialogue summarization model proposed by Wu et al. (2021). Before processing the dialogues, a preprocessing pipeline also provided by the authors is applied, which first cleans up the text and labels the utterances based on whether they have meaningful overlap with tokens from the gold labels or not, based on the intersection of their stemmed tokens. Meaningful overlap here means at least one hit which is not an English stopword.

Leo: Piper?
Piper: Hm?
Leo: What are you doing?
(Leo sits up. Piper walks out of the nursery carrying a packet of diapers.)
Piper: I'm putting the diapers back where they belong, that is what I'm doing.
(She puts the diapers on a shelf.)

Table 1: Example of a screenplay snippet from SummScreen-FD with environment or character descriptions (in brackets)

As mentioned in Section 2.1, this method introduces the idea of constructing a summary sketch as a step prior to predicting the summary itself, which is one of the main features of this architecture. The summary sketch consists of the keywords extracted by applying syntax-driven sentence compression method (Xu and Durrett, 2019) combined with constituency parsing with a self-attentive encoder (Kitaev and Klein, 2018), as implemented in the Berkeley Neural Parser[2]. The relations between the keywords are modelled according to a predefined *utterance intent classification* model (Wu et al., 2021). This idea makes this method suitable to our long inputs and intricate conversation structures with scene breaks and important information spread over several not necessarily adjacent utterances, because it helps capture only the core information, excluding the character or plot development turns, irrelevant to the summaries.

This dialogue processing pipeline is also easily adjustable due to the flexibility of the predefined utterance intent model. In the original model, the summary sketches are built after the FIVE Ws principle, classifying the utterances as to their intent of „why“, „what“, „where“, „when“ or „confirm“. Utterances which do not fall under any of these categories are marked as „abstain“. We extend this approach by incorporating the scene setting information by introducing the additional intent of „scene“. Subsequently, in our pipeline we create summary sketches based on the intent information

[2] https://spacy.io/universe/project/self-attentive-parser

for each line from the transcripts and the keywords which could be identified, removing noise from the data (see Tables 2 and 3 for examples). We limit the amount of utterances which are considered as a part of the sketch to 20 in order to stay below the maximum output limit of 1024 tokens.

intent	line	keywords
abstain	*Leo:* Piper?	['piper']
abstain	*Piper:* Hm?	[]
what	*Leo:* What are you doing?	[]
scene	(Leo sits up. Piper walks out of the nursery carrying a packet of diapers.)	['leo', 'piper']
abstain	*Piper:* I'm putting the diapers back where they belong, that is what I'm doing.	[]
scene	(She puts the diapers on a shelf.)	[]
abstain	*Leo:* But it's 2:00 in the morning.	[]
what	*Piper:* Yeah, well, apparently our little ghosts and goblins are not sleeping, so how can I? I wish they would just attack us rather than move stuff around.	['are not sleeping']
scene	(She goes back in the nursery and picks up a pile of diapers from under the crib. She takes them into the bedroom and places them on the shelf.)	['takes them into the bedroom and places them on the shelf']

Table 2: Example of processed dialogues with the classified intent and extracted keywords from SummScreen-FD

The gold label is then concatenated to the sketch separated by the token **TLDR**, which together serve as a label which the generator is trained to predict (for an example, see Table 3). Our generator architecture is based on the Longformer Encoder Decoder (LED) (Beltagy et al., 2020), of which we used the large version with input size of 16384 tokens. The model has been retrieved via the HuggingFace transformers model hub[3]. A graphic representation of our pipeline is in Figure 1.

The training has been performed on a single NVIDIA A40-48C GPU with 48 GB RAM with training batch size of 2 and gradient checkpointing. We performed a few experiments, tuning the learning rate and adjusting the maximum epoch size. We could achieve the fastest and most reliable training process training at initial learning rate of 5e-5 (with Adam optimizer), maximum epoch size of 40. Early stopping ended the training after reaching the best model after 19 training epochs and model not improving for 5 subsequent epochs.

The training script and instructions can be found on Github[4].

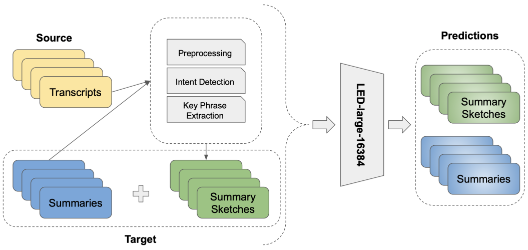

Figure 1: Workflow of the used framework to summarize TV show transcripts. It processes transcripts (source) and gold label summaries, modelling intent of the source and extracting key phrases based on the overlap with the gold summaries, and creates summary sketches (e.g. Table 3) and concatenates them with the summaries, which together serve as model targets. A Longformer Encoder Decoder is then used to predict both summary sketches and summaries from the source.

5 Results

The winning model that we then used for producing summaries for the blind test data was determined based on the validation set performance employing early stopping. We include evaluations of the validation ROUGE-1 and ROUGE-L scores across all training epochs in Figure 2. The best performing model reached a ROUGE-1 performance of 34.45 and ROUGE-L of 28.05 respectively in Epoch 19. Based on this we chose this model for evaluation on the test set.

Test set evaluation shows generally lower, but similarly promising results. Table 4 shows some of the shared task results for reference. Our model was able to beat all LED baselines by a large margin, while simultaneously achieving the highest

[3] https://huggingface.co/allenai/led-large-16384
[4] https://github.com/nkees/creative-abs-summ-sketch

target (sketch + gold label)	**prediction**
0 abstain lily marshall waits at the airport 1 abstain during the second year of college , when lily is back from vacation marshall 2 none 3 abstain lily marshall waits at the airport 4 where ted 5 none 6 none 7 abstain ted and marshall are in mcclaren's 8 none 9 none 10 abstain lily want me to take a taxi to the airport just to get a taxi in the opposite direction 11 none 12 none 13 abstain that's all i want in the world 14 abstain runs out of the bar ted barney 15 abstain wanted to see what looks like a license in arizona 16 none 17 what 's good 18 none 19 none **TLDR** Lily comes back from a trip to Seattle only to run into a blizzard, which may threaten an airport ritual of meeting Marshall. Meanwhile, Ted & Barney offer to keep watch over the bar at MacLaren's while awaiting their dates, since Carl wants to close up early.	0 none 1 when when lily is back from vacation 2 none 3 none 4 where ted 5 none 6 none 7 abstain are in mcclaren's 8 none 9 none 10 none 11 none 12 abstain that's all i want in the world 13 abstain ted and barney 14 abstain wanted to see what looks like a license in arizona 15 none 16 what 's good 17 none 18 none 19 what barney leads ted to the bar entrance. **TLDR** When Lily is forced to spend the holiday in Seattle with her boyfriend, Ranjit, Ted and Barney try to find a bar.

Table 3: Example of target and prediction, containing concatenated together sketch and summary, for a transcript from SummScreen-FD (from validation split)

Model	ROUGE-1	ROUGE-L	BERTScore-F1	LitePyramid-p2c	SummaC_{ZS}	Length
LED 1024	14.28	12.36	40.52	13.71	05.59	330
LED 4096	16.94	15.01	46.00	03.37	10.52	188
LED 16384	15.14	13.34	44.89	03.37	16.44	192
InoTUM	28.60	25.29	57.50	06.73	02.72	86
Team UFAL	24.69	23.00	52.85	04.72	12.82	289
AMRTVSumm	23.07	21.06	51.08	01.16	02.40	256

Table 4: Abridged test set performance for different metrics across systems in the shared task

scores of the submitted systems in the ROUGE and BERTScore categories. Detailed evaluations compared to the various baselines can be found in Table 4. ROUGE-1 on the test set amounted to 28.60 and ROUGE-L to 25.29. At the same time the system was also able to produce the shortest summaries of all comparison systems, with only an average of 86 tokens per summary, making it less than half as long as any other comparison summary system. Because of the shorter length of the summaries, it is, therefore, not surprising that our system also achieves the highest BERTScore Precision out of all systems at 59.34. More surprising, however, is the great recall performance, which again is the highest out of all submissions at 56.09, demonstrating that conciseness does not necessarily sacrifice relevant information. The simultaneous good results on ROUGE and BERTScore, combined with the shorter length of summaries, makes us confident that the approach we tried warrants further exploration.

The LitePyramid evaluation shows good performance compared to other competitors, but here we are witnessing that our system is not able to outperform the LED_1024 baseline model. Lastly in the $SummaC_{ZS}$ scores our system, unfortunately, scored comparatively low, suggesting problems with factual consistency of the generated outputs. To investigate this problem of factual consistency, we compared outputs of the system manually with reference scripts and found this problem to be true. Thus, improving factual consistency would be a vital step for improving the overall model in the future.

We present an example of the generated summaries of our system in Table 5. The examples taken from the TV Show Breaking Bad are consistent enough to fit the theme of the show and incorporate certain elements of the episodes in question, but tend to mix in information that is not present in the actual script.

The number of utterances that go into the sketch for each script is another hyperparameter, that can be freely set and tuned by the developer. To further explore the effect of the sketches in the overall pipeline we conducted experiments varying the number of utterances for each individual sketch. Utilizing more utterances has however not lead to any significant changes in performance on the dev set. Due to the nature of the shared task, we could

Test set instance	Summary
Breaking_Bad_276	Dr. Bravenec helps Walter decide whether or not to have the lobectomy. After much consideration, Walter decides to go through with it. Meanwhile, after learning the truth from his doctor about the status of his cancer, Dr. Bravenec decides to take action for himself.
Breaking_Bad_290	Walter decides to continue treatment after he receives promising news about his cancer. Meanwhile, Jesse has problems with the new tenant and tries to get Jane to move into her apartment with him.

Table 5: Examples of generated summaries for the TV show Breaking Bad

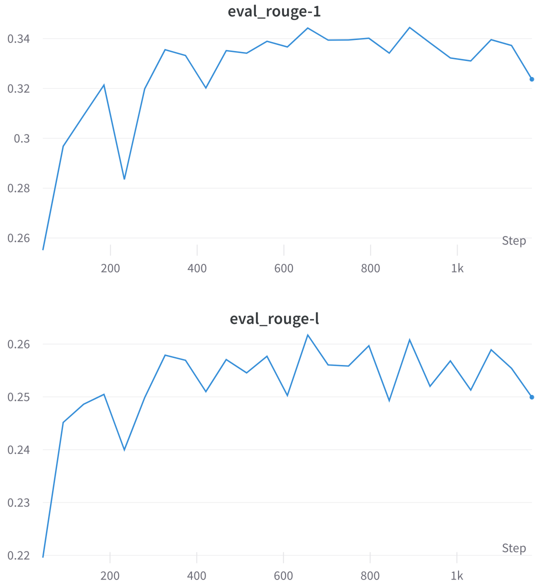

Figure 2: ROUGE-1 and ROUGE-L scores on the validation set across training epochs.

not provide evaluation on the test set, but believe these not to be a deciding factor in the success of the system.

6 Conclusion

In this paper, we presented our submission to the CreativeSumm 2022 shared task for tackling the problem of summarizing television script based on the SummScreen Forever Dreaming dataset. Our system is based on previous work by Wu et al. (2021) and extends the utterance and sketch categories particularly for the task of screenplay summarization. We have shown that this method can improve over baseline LED summarization significantly and have achieved good ROUGE and BERTScore performance, which were the highest among submitted systems. At the same time, this method was able to produce the most concise summaries out of the field.

The biggest issue of our system is factual consistency, often mixing up specific details and actors in the summarization part, thus creating seemingly good summaries but with factual errors, which would be hard to spot without actual knowledge of the underlying text. Improving factual consistency would, therefore, be an important follow-up step in further developing this approach.

Similarly, the preprocessing of sketches could be optimized by reevaluating utterance categories and trying to further specify relevant utterances for the task of script summarization with less of a focus on spoken dialogue alone to improve the base performance even more.

We are confident that with an improvement of the factual consistency our system will be able to also score higher in a human evaluation process, where such discrepancies are weighed much higher than in a pure word-level or representation-level approach.

References

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

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. *CoRR*, abs/2010.01672.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. Summscreen: A dataset for abstractive screenplay summarization. In *ACL*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. *CoRR*, abs/1805.01052.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Zhengyuan Liu and Nancy Chen. 2021. Controllable neural dialogue summarization with personal named entity planning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 92–106, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhengyuan Liu, Ke Shi, and Nancy F. Chen. 2021. Coreference-aware dialogue summarization. *CoRR*, abs/2106.08556.

Seongmin Park and Jihwa Lee. 2022. Unsupervised abstractive dialogue summarization with word graphs and POV conversion. In *Proceedings of the 2nd Workshop on Deriving Insights from User-Generated Text*, pages 1–9, (Hybrid) Dublin, Ireland, and Virtual. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. Controllable abstractive dialogue summarization with sketch supervision. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5108–5122, Online. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. *CoRR*, abs/1902.00863.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Awadallah, Dragomir Radev, and Rui Zhang. 2022. Summ^n: A multi-stage summarization framework for long input dialogues and documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1592–1604, Dublin, Ireland. Association for Computational Linguistics.

Yusen Zhang, Ansong Ni, Tao Yu, Rui Zhang, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz, Ahmed Hassan Awadallah, and Dragomir Radev. 2021. An exploratory study on long dialogue summarization: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4426–4433, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.

Yicheng Zou, Bolin Zhu, Xingwu Hu, Tao Gui, and Qi Zhang. 2021. Low-resource dialogue summarization with domain-agnostic multi-source pretraining. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 80–91, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.