AMRTVSumm: AMR-augmented Hierarchical Network for TV Transcript Summarization

Yilun Hua[*]
Columbia University
yh3228@columbia.edu

Zhaoyuan Deng[*]
Columbia University
zd2286@columbia.edu

Zhijie Xu
Carnegie Mellon University
zhijiex@andrew.cmu.edu

Abstract

This paper describes our AMRTVSumm system for the SummScreen datasets in the Automatic Summarization for Creative Writing shared task (Creative-Summ 2022). In order to capture the complicated entity interactions and dialogue structures in transcripts of TV series, we introduce a new Abstract Meaning Representation (AMR) (Banarescu et al., 2013), particularly designed to represent individual scenes in an episode. We also propose a new cross-level cross-attention mechanism to incorporate these scene AMRs into a hierarchical encoder-decoder baseline. On both the ForeverDreaming and TVMegaSite datasets of SummScreen, our system consistently outperforms the hierarchical transformer baseline. Compared with the state-of-the-art DialogLM (Zhong et al., 2021), our system still has a lower performance primarily because it is pretrained only on out-of-domain news data, unlike DialogLM, which uses extensive in-domain pretraining on dialogue and TV show data. Overall, our work suggests a promising direction to capture complicated long dialogue structures through graph representations and the need to combine graph representations with powerful pretrained language models.

1 Introduction

Abstractive summarization of TV show episodes aims to produce a summary from their transcripts or screenplays, capturing important plot development and character relations. For this shared task, we participated in the two SummScreen categories, which involve abstractively summarizing prime-time TV series (ForeverDreaming) and daytime soap operas (TVMegaSite) (Chen et al., 2022). This task presents several new challenges compared with other abstractive summarization tasks. First, transformer-based language models that perform well on shorter texts become computationally expensive when their self-attention is applied to long inputs (Vaswani et al., 2017). Also, consecutive scenes often describe parallel or different subplots, making it difficult to integrate information and present a correct narrative (Chen et al., 2022). Finally, like other dialogue texts such as meetings and media interviews, TV transcripts contain complicated character and entity interactions as well as more varied structures.

Works on long-document summarization have explored transformers with sparse or window-based attention (Beltagy et al., 2020; Wang et al., 2020), hierarchical models (Zhu et al., 2020), and the "retrieve-then-summarize" approach (Chen et al., 2022; Zhang et al., 2021). Large pretrained language models such as BART-large also give strong results by taking longer inputs at the cost of larger embeddings and increased computational complexity (Lewis et al., 2019; Zhong et al., 2021). However, despite the many works addressing the long transcript problem, few have studied novel approaches to model the complicated interactions and structures in TV transcripts. Even the state-of-the-art on SummScreen, DialogLM, relies on dialogue-specific denoising pretraining on TV data. The model architecture itself does not address TV transcripts' conversational structures and takes the input transcripts as plain texts (Zhong et al., 2021).

Therefore, we propose a novel Abstract Meaning Representation (AMR) to capture the diverse entity interactions and complex structures of TV transcripts. AMR, as a graph representation, captures the most salient semantic knowledge using its concept nodes and preserves inter-concept relations with its labeled edges. It is thus believed to convey information orthogonal to the text input (Song et al., 2019). Our work generalizes the sentence-level AMR introduced by Banarescu et al. (2013), adding new features to make them suitable for individual scenes of TV shows. We use these scene-level AMRs to augment a hierarchical

[*] equal contribution

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 36–43
July 5–10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.26615/978-954-452-056-4_006

encoder-decoder baseline. To this end, we also propose a cross-level cross-attention to scene AMRs, such that the encoder of local tokens and utterance embeddings can benefit from the structural information and higher-level semantics from the entire corresponding scene, without being interfered by an adjacent scene, which may focus on a parallel or different subplot.

To sum up, the major contributions of our work are presented as follows:

- We propose the steps to construct scene AMRs and introduce 1) Speaker/Utterance nodes and 2) Coreference/Pronoun edges, both of which connect the standard sentence AMRs to capture and extract core semantic and structural information of a scene.
- We design a cross-level cross-attention mechanism so that the encoding of local tokens and utterance embeddings can benefit from the structural and higher-level information from the scene.
- We demonstrate the effectiveness of our AMR augmentation on SummScreen and discuss the need to combine it with dialogue-specific pretraining.

2 Datasets

We participated in the two SummScreen categories of the CreativeSumm 2022 shared task: summarization of primetime television transcripts (ForeverDreaming) and summarization of daytime "soap opera" transcripts (TVMegaSite) (Chen et al., 2022).

We primarily experimented with our system on ForeverDreaming, since it includes more genres and covers 66 TV shows in the train set. We used its entire train set of 3673 episodes to train our model. For TVMegaSite, we only used 6000 of its 18915 training episodes due to the time constraint. The 6000-episode subset was sampled from the original train set to include approximately the same number of episodes from each TV show. We still use the original dev and test sets for both ForeverDreaming and TVMegaSite.

3 Constructing AMR Representation For TV Series

3.1 Scene AMRs

A scene in a TV show episode is a consecutive sequence of closely related lines and actions. For TV transcripts, in particular, we define a scene as a sequence of character utterances and stage directions contributing to a subplot. Here, an utterance is an uninterrupted line by a character, which can contain one or more sentences. Within a scene, speakers may respond to each other, request and perform actions, and refer to entities mentioned by others. All of these interactions give rise to complex dialogue structures. Therefore, we use the AMR graph representation to explicitly capture these important relations and core semantics, which can be difficult to discern for conventional transformers operating on text input.

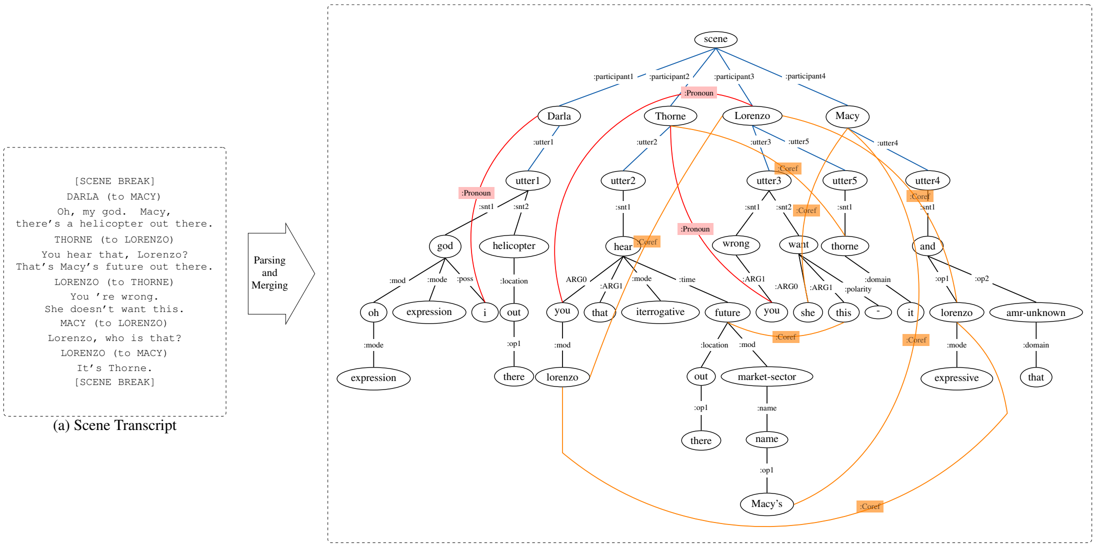
To construct scene AMRs, we adapt the steps in Bai et al. (2021), which build dialogue AMRs, and additionally introduce speaker nodes, utterance nodes, and a new procedure to represent coreferences. As illustrated by Figure 1, given a scene consisting of multiple utterances, we use the AMR parser by Cai and Lam (2020) to obtain an AMR graph for each utterance and then construct the scene AMR by connecting utterance AMRs. We then add utterance nodes, speaker nodes, and a dummy scene node (the root node), as well as the edges that capture node relations.

Utterance Node/ Utterance Edge. Given an utterance containing one or more sentences, we parse each sentence into its AMR graphs, and connect them with an utterance node tagged `utter` through sentence edges (tagged as `snt1`, `snt2`, etc.). We then connect the utterance node to the corresponding speaker node with an utterance edge (tagged as `utter1`, `utter2`, etc.).

Speaker Node/ Participant Edge. For each speaker in the scene, we add a speaker node tagged with the corresponding speaker name and connect it with the scene node using a participant edge (tagged as `participant1`, `participant2`, etc.). The scene node therefore acts as a root of the entire scene AMR.

Compared with Bai et al. (2021), we want our proposed speaker nodes and utterance nodes to encode the fine-grained hierarchical information from different levels of the scene AMR (synthesizing multiple utterances by one speaker, multiple sentences within one utterance, and etc.). This is made possible by our graph encoder, which exploits their abundant and unique interactions with other AMR concepts (see Section 4.1).

Coreference Edge. Like Bai et al. (2021), we



[SCENE BREAK]
DARLA (to MACY)
Oh, my god. Macy,
there’s a helicopter out there.
THORNE (to LORENZO)
You hear that, Lorenzo?
That’s Macy’s future out there.
LORENZO (to THORNE)
You ’re wrong.
She doesn’t want this.
MACY (to LORENZO)
Lorenzo, who is that?
LORENZO (to MACY)
It’s Thorne.
[SCENE BREAK]

(a) Scene Transcript

(b) Scene AMR Graph

Figure 1: Generating Scene AMR

use NeuralCoref[1] to obtain coreference relations between words, and JAMR[2] to obtain alignment between concepts and words. Yet unlike Bai et al. (2021), if an utterance mentions a speaker (often a third character), we also connect their concept in utterance AMR to the corresponding speaker node.

Pronoun Edge. Because the off-the-shelf NeuralCoref does not guarantee finding all coreference relationships, we also add rule-based pronoun edges. We connect first-person pronoun concepts (e.g., ‘I’, ‘We’) to the current speaker node with pronoun edges. For some TV series, there is information that indicates which character the current speaker is talking to (e.g., Alice (to Bob):). When this information is available, we also connect the second-person pronouns (e.g., ‘You’) to the corresponding speaker node (e.g., Bob).

3.2 Scene Segmentation

Transcripts in the SummScreen dataset often have accurate [SCENE_BREAK] tokens suggesting the beginning of a new scene. These [SCENE_BREAK]s segment the transcripts into texts of reasonable length, for which we can concisely construct scene AMRs and encode them with a graph transformer. However, SummScreen is based on community-contributed transcripts and some of the transcribers may have a different understanding of scene breaks. We found that some episodes contain much fewer [SCENE_BREAK]s than others or no [SCENE_BREAK]s at all.

Thus, we adopt an existing strategy (Chen and Yang, 2020) that combines the classic topic segment algorithm C99 (Choi, 2000) with SentenceBERT (Reimers and Gurevych, 2019), to re-segment scenes into reasonable lengths. We still primarily use the [SCENE_BREAK]s from the transcripts and only apply this algorithm on long scenes that exceed our threshold of 600 tokens.

4 System Overview

Our AMR-augmented hierarchical summarization network consists of a hierarchical text encoder and a scene-level AMR encoder. The system is illustrated in Figure 2.

4.1 Scene AMR Encoder

The scene-level AMR graphs contain rich structural information and entity interactions in their AMR concepts (nodes) and edges. To exploit this graph information, we apply Zhu et al. (2019)’s structure-aware graph transformer to encode the scene AMRs. Depth-first traversal is used to linearize the AMR into a sequence of concepts. The relationship r_{ij} between a concept pair x_i, x_j is encoded using convolutional network, which convolves the shortest sequence of edges between the pair. As in Zhu et al. (2019), every concept node attends to every other concept node with a modified attention mechanism informed by the relationships between them. In the end, the output of the AMR encoder is

scene-encoder $(scene\_AMR) = \{x^c_0, .., x^c_m\}$, for a scene with m AMR concepts. Note that the

[1] https://github.com/huggingface/neuralcoref

[2] https://github.com/jflanigan/jamr

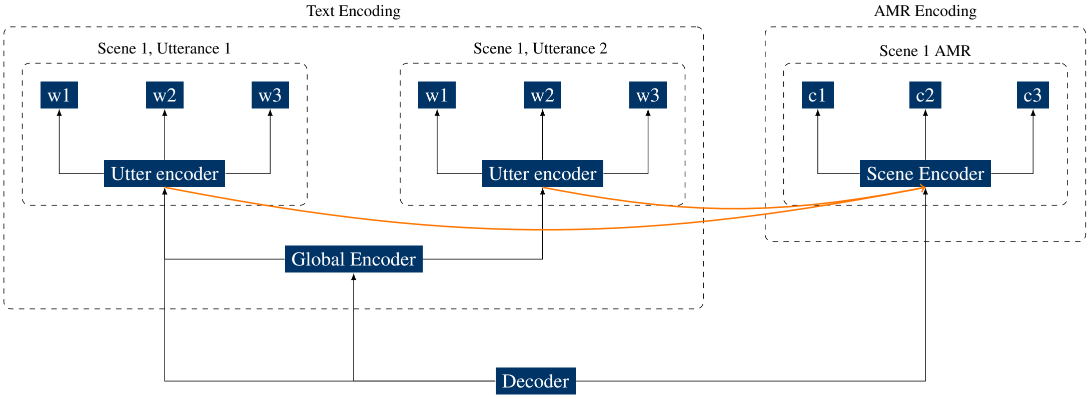


Figure 2: AMR-augmented hierarchical summarization network. The orange arrows indicate the cross-level cross-attention from the utter-encoder's [BOU] outputs to the scene encoder's concept outputs.

graph encoder processes each scene AMR independently, so the encoding of local concepts is not interfered by concepts in neighboring scenes that develop their distinct subplots.

4.2 Hierarchical Text Encoder

We adopt a hierarchical model for the text input because the conventional transformers face huge limitations when encoding long transcripts from TV series. They apply full self-attention to the entire input sequence, despite the computational complexity being quadratic in the input length. Our model adapts the more efficient hierarchical architecture from HMNet (Zhu et al., 2020), which has shown promising results on meetings and other long-dialogue summarization tasks.

Utterance-level Encoder. Following Zhu et al. (2020), our hierarchical structure starts with an utterance-level transformer (utter-encoder), which encodes a sequence of tokens from an utterance u_i using self-attention. We initialize a trainable token embedding matrix D using the pre-trained weights from Zhu et al. (2020). Following their approach, we enrich the token representations by training two other embedding matrices for part-of-speech (POS) and entity tags. The token embedding, POS embedding, and entity embedding are concatenated into the overall token input $x_{i,j}$ (for the j-th token in the i-th utterance). A special token $w_{i,0}$=[BOU] (beginning-of-utterance) is added before every utterance, which is essential for the later utterance representation and cross-attention. We denote the utterance-level encoding operation in every transformer layer as follows:

layer-k$(\{\hat{x}_{i,0,k}, .., \hat{x}_{i,L_i,k}\})$
= $\{\hat{x}_{i,0,k+1}, .., \hat{x}_{i,L_i,k+1}\}$, for the i-th utterance that has length L_i.

Cross-level Cross-attention to AMR outputs. The [BOU] token we added above is analogous to the [BOS] (beginning-of-sentence) token in document encoders. Conventionally, the hidden state output for the [BOS] token can be trained to directly model sentence-level information. For dialogues, however, an utterance has many diverse and complicated structures, making it more difficult to derive reliable patterns through self-attention. Therefore, we enrich the [BOU] embedding $\hat{x}_{i,0,k+1}$ with the scene AMR. In an utterance-level encoder layer, the [BOU] embedding will first have full attention to the tokens in the same utterance. It then cross-attends to the hidden states of all AMR concepts from the entire scene where this utterance locates. Specifically, we derive the Key and Value matrices for the cross-attention from the AMR hidden states and the Query matrix from the [BOU] embeddings.

We call this mechanism "cross-level" cross-attention because it allows the upper-level, more global information (scene-level) to guide the encoding of lower-level information (utterance-level). First, it improves the utterance representation by providing access to the entire scene. Each [BOU] embedding can attend to all the concepts in the scene. Also, the root node, utterance nodes, and important entity nodes, would likely have aggregated information to different extents in graph hidden states so the cross-attention can easily utilize. This scene-level information improves the [BOU] embedding and can guide the extraction of local token features after the improved [BOU] embedding is sent to the next utter-encoder layer. Second, this cross-attention captures the relational information from AMRs, allowing for a better grasp of dialogues' structural features.

This cross-level cross-attention is more efficient than directly attending to all the tokens in the scene. AMR extracts salient features and complex interactions while compressing the input sequence for the attention mechanism. For a typical scene in SummScreen, the number of AMR nodes ranges from half to two-thirds of the token number, leading to significantly lower cross-attention complexity than attending to all the tokens in the scene.

Finally, the overall utter-encoder with cross-level cross-attention has the following operations:

layer-1($(\{x_{i,0}, ..., x_{i,L_i}\})$=$\{\hat{x}_{i,0,1}, ..., \hat{x}_{i,L_i,1}\}$,
layer-k ($\{\hat{x}_{i,0,k}, .., \hat{x}_{i,L_i,k}\}$)
=$\{\hat{x}_{i,0,k+1}, .., \hat{x}_{i,L_i,k+1}\}$.

The cross-attention is applied to $\hat{x}_{i,0,k}$ after every layer's self-attention, where $\{x^c_0, ..., x^c_m\}$ is the graph hidden states of the scene:

$\hat{x}_{i,0,k}$ = cross_attn ($\hat{x}_{i,0,k}, \{x^c_0, ..., x^c_m\}$)

Overall, the output is:

utter-encoder ($utter_i, \{x^c_0, ..., x^c_m\}$)
= $\{x^u_{i,0}, .., x^u_{i,L_i}\}$.

Global Encoder. Like in Zhu et al. (2020), a global transformer aggregates the last utter-encoder hidden states of the [BOU] tokens. The output is denoted as

global-encoder ($\{x^u_{1,0}, .., x^u_{n,0}\}$) = $\{x^G_1, ..., x^G_n\}$
for an episode of n utterances.

4.3 Decoder

We use a transformer decoder to generate the summary sequence. At each decoding stage t, self-attention is applied to hidden states of the previous $t-1$ generated tokens. Then, the model synthesizes information across different levels by three cross-attention blocks, to the token embeddings from the utter-encoder, to the concept embeddings from the scene-encoder, and to the utterance embeddings from the global-encoder, respectively. In this way, AMR information not only benefits the token and utterance encoding but also directly contributes to the generation of summaries at the decoding stage.

5 Implementation Details

5.1 Initialization

We use the pre-trained weights from Zhu et al. (2020)'s HMNet to initialize our utter-encoder and global-encoder, including the token embedding matrix D. Their pre-training was done on news articles reformatted into conversation-like texts. We consider this an out-of-domain pretraining, which should be distinguished from the dialogue-specific pretraining of the current SOTA system DialogLM (Zhong et al., 2021).

We then copy and resize the matrix D to D_c as the embedding matrix for AMR concepts, expanding the matrix vocabulary with additional AMR concepts not present during pretraining. Since many AMR concepts are also common words and names, initializing with HMNet's pretrained embedding will help better extract relations between text tokens and AMR concepts. However, we do not tie the weights of AMR and text embedding matrices, as we expect them to emphasize different meanings when a token is treated as a word versus as an AMR concept.

5.2 Training

We use an effective batch size of 40 episodes and train our system for 2400 updates. The initial learning rate is set to 5e-6. Within 150 updates, it linearly increases to and remains at 5e-4. In addition, we use RAdam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$.

6 Results

At the time of our blind test submission to the shared task, we only trained our system on smaller subsets of the SummScreen datasets. The resulting checkpoints therefore did not achieve a high performance on the original test sets of SummScreen nor on the blind test sets provided by the shared task. We will analyze these results in Section 6.3. Here, we first present our more recent results from training on the expanded train sets after our blind test submission. Specifically, we eventually used the complete train set for ForeverDreaming and a re-sampled 6000-episode subset for TVMegaSite. All results were reported on the original test sets without re-sampling.

6.1 Results on Original SummScreen Datasets

We primarily compare our results with the hierarchical baseline HMNet from Zhu et al. (2020). After grid searching over key hyper-parameters, we trained HMNet using the same setup as our system, which produced a better result than the setup in the original paper. We also include results of other strong baselines reported by Zhong et al. (2021), including Longformer (Beltagy et al., 2020), BART-Large (Lewis et al., 2019), UNI-LM (Dong et al.,

Models	ForeverDreaming			TVMegaSite		
	R-1	R-2	R-L	R-1	R-2	R-L
Longformer*	25.90	4.20	23.80	42.90	11.90	41.60
BART-Large*	33.82	7.48	29.07	43.54	10.31	41.35
UNILM-base*	32.16	5.93	27.27	43.42	9.62	41.19
DialogLM-sparse*	35.75	8.27	30.76	45.58	10.75	43.31
HMNet	27.08	5.41	23.95	41.04	9.28	39.05
AMR_cross (our system)	31.45	7.39	27.14	43.08	10.77	41.53
- cross attention	31.07	6.15	27.30	-	-	-
- speaker/utter	31.10	6.09	27.26	-	-	-

Table 1: Comparison with baselines and ablation results. * indicates results reported by Zhong et al. (2021). "-" indicates we removed that feature for ablation study.

2019), and DialogLM (Zhong et al., 2021).

As shown in Table 1, our system consistently outperforms its hierarchical baseline HMNet on both ForeverDreaming and TVMegaSite. It also fully outperforms Longformer on ForeverDreaming and achieved comparable results on TVMegaSite, despite using a smaller TVMegaSite train set. In addition, our system sometimes outperformed BART-Large and UNILM-base in Rouge-2 or Rouge-L or both.

Here, HMNet, Longformer, UNILM, and BART-Large are all pretrained on out-of-domain data like our system. This suggests that scene AMR can effectively contribute to a summarization system through cross-attention. However, our system's performance is still lower than that of dialogLM_sparse, one of the best performing dialogLM variants, which uses extensive pretraining on TV data. Therefore, our future work will extend our proposed AMR and cross-attention approaches to combine with more powerful pretrained models.

6.2 Ablation Studies

We used ForeverDreaming to perform ablation studies because it has more TV show genres than TVMegaSite but fewer episodes overall. This allows us to conduct experiments efficiently and obtain more generalizable results. Our ablation includes removing the speaker and utterance nodes and omitting the cross-level cross attention to AMR concepts. Due to the time constraint, we did not perform a separate ablation for the coreference/pronoun edges. For each experiment, we report the ROUGE scores on the original test split of ForeverDreaming.

As shown in the last three lines of Table 1, removing cross attention and speaker/utter nodes both resulted in a lower overall performance than AMR_cross, though they are still better than the HMNet baseline that uses no AMR at all. Part of the performance decrease when speaker nodes are omitted may also come from the loss of coref/pronoun edges associated with these nodes. Therefore, we will conduct more thorough ablation experiments in the future, considering the case when only speaker-associated coref/pronoun edges are removed versus the case when all these edges are removed. Overall, these results suggest the effectiveness of our proposed approaches.

6.3 Blind Test Submission

At the time of blind test submission, we used a model checkpoint trained on a subset of 2000 episodes for ForeverDreaming. For TVMegaSite, we used a subset of 2500 episodes. Table 2 shows that blind test sets seem to be harder than the original test sets: the same model checkpoint achieved 28.84 Rouge-1 for ForeverDreaming's original test set while the Rouge-1 on the blind test was 23.07. The performance drop for TVMegaSite was even greater, from 41.16 Rouge-1 to 34.26 Rouge-1. Other Rouge scores were also lower for the blind test sets. This performance decline was much greater than what we observed between the original train, dev, and test sets in our experiments. This is likely a result of different TV show distributions or different transcript styles between the blind tests and the originally released train/dev/test sets. Using a smaller train set might have undermined our model's generalization, but it is likely not the main reason behind this discrepancy.

Instead, the effects of using smaller train sets are

Data Split	ForeverDreaming			TVMegaSite		
	R-1	R-2	R-L	R-1	R-2	R-L
Blind Test	23.07	3.03	21.06	34.26	7.17	32.80
Original Test	28.84	5.83	25.58	41.16	10.67	39.74

Table 2: Blind test scores vs. original test scores of the checkpoints we used for blind test submission. The original test scores here are lower than those in Table 1 since they come from checkpoints trained on smaller datasets.

most salient when comparing the results in Table 1 and those in Table 2. Our system achieved higher test Rouge scores for both ForeverDreaming and TVMegaSite in Table 1, using checkpoints trained on more data. It suggests that our system responds well to increased dataset sizes, and our future work should exploit all the data available.

7 Conclusion

We describe our AMRTVSumm system for the two SummScreen datasets in the CreativeSumm 2022 shared task. Based on our proposed scene AMR graph and hierarchical architecture with cross-level cross-attention, our system achieves substantial improvement over its hierarchical baseline under the same out-of-domain pretraining. However, it still does not outperform the state-of-the-art model that relies on extensive in-domain pretraining. Our work suggests that despite graph representation's power in modeling complicated dialogue structures, it does not replace the role of dialogue-specific and TV-specific pretraining. A promising future direction will be to leverage the advantages of both by augmenting a state-of-the-art pretrained language model with scene AMRs.

8 Acknowledgements

This work is supported in part by Columbia University under the Class of 1939, Columbia College Summer Research Fellowship. The authors would like to thank Daniel Bauer for his constructive feedback and comments on the use of AMR. The authors are also grateful to Samantha Mayers, Zheyu Liu, Duaa Tariq, António Câmara, and Zongmin Yin for their advice and assistance.

References

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic Representation for Dialogue Modeling. 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 4430–4445, Online. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

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

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4106–4118, Online. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *1st Meeting of the North American Chapter of the 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*. Curran Associates Inc., Red Hook, NY, USA.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768.

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, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Dialoglm: Pre-trained model for long dialogue understanding and summarization. *CoRR*, abs/2109.02492.

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.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better AMR-to-text generation. 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 5459–5468, Hong Kong, China. Association for Computational Linguistics.