Abstractive Screenplay Summarization with Attention

Erick Martinez and Marc Semonick

April 9, 2022

Abstract

While there is often work being done in text summarization broadly across the field of NLP, few groups are working on the specific problem of summarizing screenplays. Screenplays have a different structure than most documents and provide a unique challenge in summarization. A combination of techniques used across various summarization fields may improve outcomes in making automated summaries useful and human-readable. Recent advances in attention can help models to better understand the context of scenes so that useful information can be extracted from a large body of text. Combining extraction with abstraction makes human readability a priority so that the results can be more useful for proposed end users. Experimentation in this paper shows that further research needs to be done on these assumptions to make any leaps in summarizing scripts.

1 Introduction

In the realm of NLP, a common challenge is that of text summarization: The process of condensing a larger corpus of text into a concise representation of the original text. Common tasks include the summarization of news articles and social media posts towards various ends[Hartl and Kruschwitz, 2022]. These summarization tasks often focus on smaller bodies of text that are typically easier for neural networks to understand and represent. Leading summarization tools such as T5 [Raffel et al., 2019] or Pegasus [Zhang et al., 2019] limit input length to 512 tokens. Summarizing larger bodies of text becomes a challenge computationally. In addition to size as a constraint, there are some more unique document types such as scientific papers [Tretyak and Stepanov, 2020] that may not perform well with a generalized summarization tool. One such category of documents is screenplays for television and film, which follow their own unique structure and style. Combining these unique features and the length of typical screenplays, the automatic summarizing of screenplays is a challenge in our field worth exploring.

A niche opportunity, the effective summarization of screenplays could be of specific use in the film industry, where script screeners sift through piles of submitted offerings. Having a reliable summary would help in knowing whether the full script is worth their attention. It could also be helpful to aspiring writers and hobbyists for the same reason as they seek to read scripts that meet their specific interests.

The length of screenplays pose the first challenge for this summarization task. There is too much text to put through normal summarization tools. Some work has been done in this area, to attempt to reduce a screenplay to its key scenes. The authors

[Papalampidi et al., 2020a] propose that a screenplay can be isolated to five key turning point scenes that contain the most crucial data about the work. Using a bi-directional LSTM network, they extract key scenes for further summarization. While not focused on movie scripts, other tactics tackle the summarization of another long-form type of text (Research Papers) by combining both extractive and abstractive techniques [Tretyak and Stepanov, 2020]. New research leverages the attention mechanism of Transformers to better identify key scenes in a screenplay for summarization [Lee et al., 2021].

Identifying turning points appears to be a valid strategy to extract key scenes from a film and minimize the work of summarizing the entire film, however abstractive summarization will help to make the result more human-readable and relevant to the proposed audience (film industry professionals and film hobbyists). We propose that a combination of the following techniques will lead to improved screenplay summaries:

1. Film Turning Point identification.
2. Leveraging Transformers for attention across scenes in a film.
3. Abstractive summary of identified Turning Point scenes.

2 Method

2.1 Dataset

We use the TRIPOD dataset, which contains 122 films and human-annotated turning point scenes [Papalampidi et al., 2020b]. The TRIPOD dataset is quite imbalanced as for each 180-scene movie, there are only 5 turning points. In building the feed-forward network, we encountered a bug in Keras that prevented us from using the class weight feature. This was severely needed as we had a very unbalanced dataset: 1 of each turning point for every film of approximately 200 scenes. To overcome this imbalalnce, we oversample the positive classes by a factor of 256 to achieve a workable model.

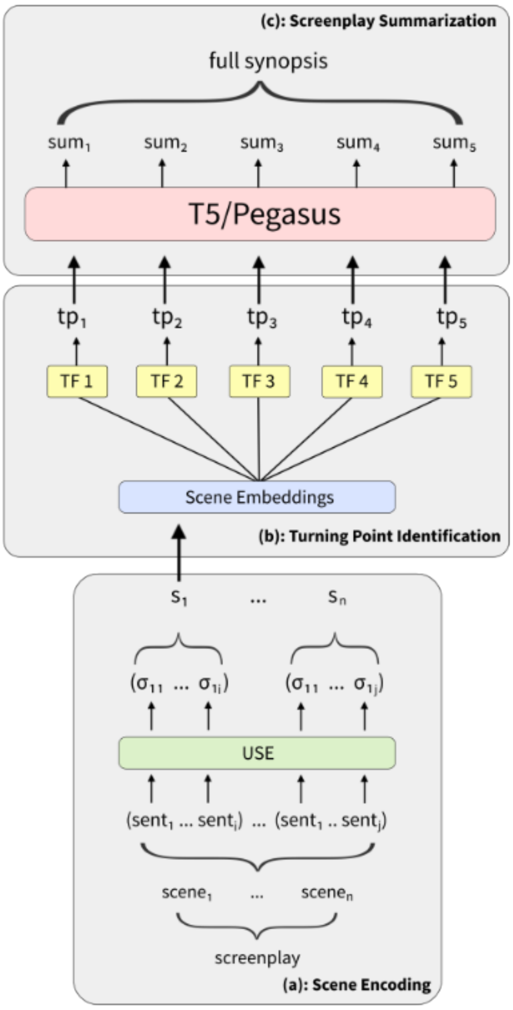
(c): Screenplay Summarization
full synopsis
sum_1 sum_2 sum_3 sum_4 sum_5
T5/Pegasus
tp_1 tp_2 tp_3 tp_4 tp_5
TF 1 TF 2 TF 3 TF 4 TF 5
Scene Embeddings
(b): Turning Point Identification
s_1 ... s_n
$(\sigma_{11} \dots \sigma_{1i})$ $(\sigma_{11} \dots \sigma_{1j})$
USE
$(sent_1 \dots sent_i)$ ... $(sent_1 .. sent_j)$
$scene_1$ ... $scene_n$
screenplay
(a): Scene Encoding

Fig.1: Data flow of summarization model

2.2 Scene Encoding with Attention

The first challenge in building our model is how to break down the full amount of data in each screenplay. Tokenizing each word would be too much information for our model to process efficiently. To address this obstacle, we leverage a technique from Papalampidi et al [2020a], the Universal Sentence Encoder (USE) [Cer et al., 2018]. USE allows us to encode each sentence of the scene into a 512-token array, rather than having to encode each word. However with up to 200

	Base-ext	T-ext	T-Pegasus	T-T5-Large
ROUGE-1	11.7	22.9	17.5	19.9
ROUGE -2	1.6	3.5	3.1	3.1
ROUGE-L	10.9	11.6	10.2	11.1
BLEURT	-0.896	-0.850	-1.00	-0.93

Table 1: ROUGE and BLEURT Scores

scenes in a film, and over 50 sentences per scene, this is still a lot of information to parse. To further handle the volume of data, we take the mean of the sentence encodings to represent each scene as the average meaning contained in its sentence vectors. With the mean of the USE encodings, we can now represent each film as a 256 x 512 vector. We then use multi-headed attention so that the scene embeddings contain the context of the entire film as a whole.

2.3 Turning Point Identification and Extraction

Gathering the scene embeddings, we pass them through a feed-forward neural network with softmax activation. The hyperparameters of the network are:

- Hidden Layer: 1 layer, 256 dimensions, Relu
- Output Layer: 1 x 256 Softmax
- Loss: Binary Cross-Entropy

We build a separate model tuned for each turning point, and run each screenplay's scene embeddings through each model to produce the 5 most likely turning points.

2.4 Abstractive Summarization

Having identified the 5 most likely turning point scenes, we gather the initial text from those scenes and run them through three different summarization options: BERT extractive summarization, T5-large, and Pegasus. We attempted to tune T5-Large to our specific dataset but were unable due to memory constraints. Turning point scenes are fed individually to generate abstract summaries, then the summaries are concatenated together to complete a full summary of the film.

3 Results

We run our results through two different scoring metrics, ROUGE (including ROUGE-1, ROUGE-2, and ROUGE-L) [Lin, 2004] and BLEURT [Sellam et al., 2020]. ROUGE has been used for summarization metrics across multiple papers, however BLEURT claims to more closely model human judgement on the task.

3.1 Baseline

In addition to our Transformer models, we also score and compare to a simple baseline model. Our baseline consists of a simple heuristic: We obtain the average location of turning points from the training set, and identify the average turning points from each screenplay to complete basic extractive summaries using BERT extractive summarization.

4 Discussion

From Table 1, we can see a clear jump in ROUGE-1 and ROUGE-2 scores from the Baseline to the Transformer-based Turning Point identification models. This indicates that we are able to identify important plot elements in the script compared to the baseline, whether that be characters or events. For ROUGE-L, we do not see a large difference between extractive and abstractive methods. Since this metric looks for specific words from the reference, abstractive text generation may not generate the words expected by the metric.

Furthermore, when looking at the BLEURT scores, the most successful models were both models using extractive summaries, while the abstractive summaries appear inferior. This is a surprising result, as we expected BLEURT to favor the more human-readable language of abstractive summarization. Pre-training the summarization models may have been more effective here, however we ran into resource challenges due to the amount of text being processed. As mentioned in the Method section, being able to set class weight may have also improved turning point selection.

Exploring the abstractive outputs, we see examples of repetitive text and stage directions that are not useful to the summary of a film. Much more time needs to be spent examining the causes of these outputs. Furthermore, the reference synopses are much longer than those generated by our model. This seems to indicate that far more than just five turning points are needed to make a thorough summary of a film.

Ultimately, we cannot claim that the attention mechanism in our Transformer architecture made a notable difference to either identifying turning points or summarizing text, nor did employing abstractive summarization techniques. Strategies that could assist in future attempts may include building a larger data set of annotated turning points, including more scenes that scored highly on turning point identification, fine-tuning the summarization on screenplay synopses, and potentially feeding the scene embeddings with attention alongside the text from the scene.

References

[Cer et al., 2018] Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., and Kurzweil, R. (2018). Universal Sentence Encoder. *arXiv.*

[Hartl and Kruschwitz, 2022] Hartl, P. and Kruschwitz, U. (2022). Applying automatic text summarization for fake news detection. *arXiv.*

[Lee et al., 2021] Lee, M., Kwon, H., Shin, J., Lee, W., Jung, B., and Lee, J.-H. (2021). Transformer-based Screenplay Summarization Using Augmented Learning Representation with Dialogue Information. *Proceedings of the Third Workshop on Narrative Understanding*, pages 56–61.

[Lin, 2004] Lin, C.-Y. (2004). [rouge]: A package for automatic evaluation of summaries. *Association for Computational Linguistics*, pages 74–81.

[Papalampidi et al., 2020a] Papalampidi, P., Keller, F., Frermann, L., and Lapata, M. (2020a). Screenplay Summarization Using Latent Narrative Structure. *arXiv.*

[Papalampidi et al., 2020b] Papalampidi, P., Keller, F., and Lapata, M. (2020b). Movie Summarization via Sparse Graph Construction. *arXiv.*

[Raffel et al., 2019] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv.*

[Sellam et al., 2020] Sellam, T., Das, D., and Parikh, A. P. (2020). BLEURT: Learning Robust Metrics for Text Generation. *arXiv.*

[Tretyak and Stepanov, 2020] Tretyak, V. and Stepanov, D. (2020). Combination of abstractive and extractive approaches for summarization of long scientific texts. *arXiv.*

[Zhang et al., 2019] Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv.*