Content-based Movie Recommender with Salient Sentences Extracted by Bert

Xiaolin Deng, Data Science and AI for the Creative Industries

1. Introduction and Motivation

The emergence of short-form videos significantly impacted human lives in several dimensions. People nowadays live fast-paced lives with the explosion of data. Therefore, the demands of text-summarization have notably arisen. Although human power is the most reliable resource for text-summarization, the needs are far beyond supply. Computers have long been considered unable to perform complex and creative tasks. Text generation is an example of an exciting new wave of deep learning-based methods which threaten to challenge this belief. It's a field that has an easy-to-interpret output with straightforward applications. With the ongoing rise of streaming services, consumers have a much larger pool of movies to select from. In order to identify which movies are worth watching, it is important to have quality movie trailers. However, trailer creation requires substantial resources. In this project, we propose a method to assist movie trailer creation by automatically recommending the most salient sentences from the original film.

2. Problem Definition

We define our problem as a text summarization task, a subset of the common NLP task of text generation. The inputs are the movie subtitles, and the outputs are sentences selected from the inputs that can summarize the entire movie.

Our training input X consists of {s1, s2, …, sn} where si is a line in the film's screenplay and the ground-truth trailer for that film. For testing, our input was just X. Our output is Y, where Y X consists of {a1, a2, …, ak} where ai is one of the most informative lines from the X. For example, the training input for the movie Ant-Man will consist of the entire screenplay and the ground-truth trailer. The output will consist of the trailer script generated by the model, which we will then evaluate against the ground truth.

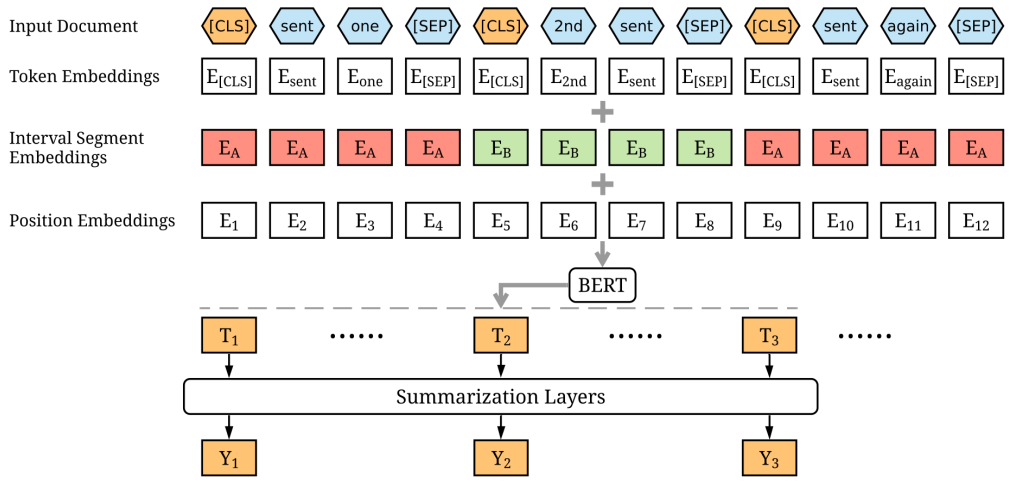
3. Solution

Our task is an example of **extractive** summarization, wherein the goal is to extract exact excerpts from the source text in the output text. In this case, the source text is the movie screenplay in its entirety, and the output text is the trailer (which is nothing more than a type of summary). At a high level, machine learning is necessary to determine the most prominent segments of the movie screenplay in an automated fashion. we will then select these segments for the output trailer. Some of the latest transformer-based NLP methods are utilized to accomplish this goal, which we will discuss in further detail in successive sections.

4. Machine Learning Design and Model Architecture

Figure 1 shows the architecture of the model we utilize, which is known as **BERTSUM** (Liu, 2022). BERTSUM is a variant of BERT that allows for extractive summarization. More specifically, we use **DistilBERT** (Sanh et al., 2020) as our base model, as it retains 95% of BERT's performance at only 45% of the computational cost.

Figure 1: BERTSUM model architecture.



The model architecture consists of a pre-trained BERT-base encoder from the masked language modeling task. It also contains a summarization classifier (a small Transformer model with three layers) that is used to assign each sentence a binary label to determine whether or not it should be included in the output trailer.

5. Data

Our datasets comprise a total of 49 movies from Marvel and DC, along with their corresponding trailers. We initially attempted to collect data using regular expressions. However, due to the complexity of variable download resources and anti-crawler protections on the subtitles websites, we ended up manually collecting the movie and trailer subtitles from the internet.

There are 73892 sentences from the movie and a total of 7927 sentences from trailers. At the beginning of each sentence, we add an integer to annotate how many seconds elapsed from the movie's start. For example, suppose the sentence "This is the Asgardian" (from Avengers: Infinity War) appears right at the movie's beginning. After preprocessing, it would become "0. This is the Asgardian".

6. Language/Library/Tools Used

We developed exclusively in Python in a Conda environment, heavily utilising third-party libraries. These libraries include PyTorch and PyTorch-Transformers for model building and Pandas for data preprocessing. We also use a GitHub repository that focuses on extractive summarization of newspaper data and applies substantial modifications to the codebase to make the preprocessing and training pipelines compatible with our movie script/trailer data.

7. Results

We evaluated our models on a test set of ten movies. We used the F1(Scikit-learn.org, 2019) value in the rouge metric as our evaluation metric using the definitions below (where beta is 1):

$$P = \frac{Nnumber\ of\ overlaping\ words}{Total\ words\ in\ candidate\ summary} \qquad R = \frac{Nnumber\ of\ overlaping\ words}{Total\ words\ in\ reference\ summary}$$

$$F - measure = \frac{\left(1 + \beta^2\right) R * P}{R + \beta^2 * P}$$

The results are shown in Table 1 and Table 2. They vary quite a bit from movie to movie, with Batman Begins being the most accurate movie according to the evaluation and Constantine doing the worst.

Table 1: The F1 rouge evaluation of the pre-trained model on the movie trailers.

movie_name	rouge-1	rouge-2	rouge-4	rouge-l
Avengers Age of Ultron	4.494E-02	7.547E-03	0.000E+00	4.494E-02
X-Men Days of Future Past	7.930E-02	1.778E-02	0.000E+00	7.930E-02
Thor Ragnarok	1.045E-01	7.519E-03	0.000E+00	1.045E-01
Constantine	4.286E-02	0.000E+00	0.000E+00	4.286E-02
Batman Begins	1.615E-01	1.085E-01	9.449E-02	1.538E-01

Table 2: The F1 rouge evaluation of the baseline model on the movie trailers.

movie_name	rouge-1	rouge-2	rouge-4	rouge-l
Avengers Age of Ultron	4.396E-02	7.380E-03	0.000E+00	4.396E-02
X-Men Days of Future Past	3.883E-02	0.000E+00	0.000E+00	3.236E-02
Thor Ragnarok	8.485E-02	6.098E-03	0.000E+00	8.485E-02
Constantine	2.548E-02	0.000E+00	0.000E+00	2.548E-02
Batman Begins	8.750E-02	2.516E-02	0.000E+00	8.750E-02

Although the Rouge metric was used as our primary evaluation, for the time being, it is an insufficient metric for trailer generation. Rouge implies that the quality of the generated trailer has to do with its direct similarity with the ground truth trailer. This may not be a reasonable assumption, as there could be several effective trailers generated by a model that has little overlap with each other. Therefore, the best evaluation would be performed by humans.

8. Related Works

Related works focusing on this problem with an NLP approach are hard to come by. There were many works (Ross, M Grauer and Bernd Freisleben, 2009, p.146; Zhou, Liu and Huang, 2018; Smith et al., 2017; Wang et al., 2020) that used audio and video input to generate trailers automatically, but these either did not use NLP or used it as a subtask.

Smart trailer(Hesham et al., 2018) attempted a much more similar task to ours. The authors used subtitles and a deep learning model to generate a movie trailer. Using this task for extractive summarisation, they were able to achieve an F1 between 0.32 and 0.47. The dataset, however, was not open source.

9. Conclusion

We successfully performed text recommendation tasks using a pre-trained BERT model for the task of movie trailer generation. By implementing the DistilBERT on our dataset, we were able to get the rouge-4 score from an average of 0.0062, although a better measure of the movie trailer's quality would be human testers.

Further improvements are needed in order to make this project industry-standard. In particular, we should greatly expand the movie trailer and screenplay from the current total of 49 examples. However, we believe that our project serves as a proof-of-concept as tool to assist in movie summarising.

10. Thinking Process

The goal of this project is to recommend sentences that people "like". Therefore, we need a standard to measure the importance of each sentence. Our initial thoughts include asking viewers to select a few random sentences or images, and the program will return the sentences that the user may like. However, these measurements are insufficient to support our predictions' reliability. Given that movie trailers are made by professional editors, we realised that trailers themselves could be the "standard" that is lined with general aesthetics. Hence we manually collected trailer scripts for evaluation.

Reference

Hesham, M., Hani, B., Fouad, N. and Amer, E. (2018). Smart trailer: Automatic generation of movie trailer using only subtitles. *2018 First International Workshop on Deep and Representation Learning (IWDRL)*. doi:10.1109/iwdrl.2018.8358211.

Liu, Y. (2022). *BertSum*. [online] GitHub. Available at: https://github.com/nlpyang/BertSum [Accessed 28 Jun. 2022].

Ross, M., M Grauer and Bernd Freisleben (2009). *Digital tools in media studies : analysis and research : an overview*. Bielefeld: Transcript ; New Brunswick [N.J, p.146.

Sanh, V., Debut, L., Chaumond, J. and Wolf, T. (2020). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. [online] Available at: https://arxiv.org/pdf/1910.01108.pdf.

Scikit-learn.org. (2019). *sklearn.metrics.f1_score — scikit-learn 0.21.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html.

Smith, J.R., Joshi, D., Huet, B., Hsu, W. and Cota, J. (2017). Harnessing A.I. for Augmenting Creativity. *Proceedings of the 2017 ACM on Multimedia Conference - MM '17*. doi:10.1145/3123266.3127906.

Wang, L., Liu, D., Puri, R. and Metaxas, D.N. (2020). Learning Trailer Moments in Full-Length Movies with Co-Contrastive Attention. *Computer Vision – ECCV 2020*, pp.300–316. doi:10.1007/978-3-030-58523-5_18.

Zhou, Z., Liu, S. and Huang, K. (2018). Research on airport trailer emergency scheduling model based on genetic simulation annealing algorithm. *IOP Conference Series: Materials Science and Engineering*, 383, p.012044. doi:10.1088/1757-899x/383/1/012044.