Semester project

Using movie plots for unsupervised Movies/TV Series summarization

Author: Nicolas BOINAY
February 9, 2022

Contents

Abstract

Text summarization is one of the most challenging problem in NLP. As most of the works have been done on summarizing structured texts like encyclopedia articles or news reports, the purpose of this project is to generate summaries of movies or TV series based on their plots. The project was then based on aligning movie plots and movie scripts in order to extract the most important scenes to summarize. In this paper, we will focus on a particular strategy based on movie dialogues summarization in order to be able to align the summary scene by scene generated with movie plots and extract the important scenes. To do so, we used a Sequence to Sequence (Seq2Seq) method based on the different views of a conversation. The code of the whole project can be found at `https://github.com/NicolasBOINAY/ScriptSummarization`

1 Introduction

An important aspect of language understanding is the ability to produce a concise and fluent summary of stories, dialogues and other textual contents. Automatic text summarization is an important topic in NLP with many approaches for different types of text but mostly news articles or scholarly publications. This project focuses on less narrative texts, movie scripts. The main idea is to focus on alignment of scenes from movie script with sentences from the plot summaries inspired by a work made in proceedings ACL 2021 [10].

As movie scripts contain mostly dialogues which are not narrative text when plot summaries are. The strategy of this project is to try a method of generating summaries for each scene from movie scripts. The purpose of this is then to work on the alignment of two narrative texts. Therefore we might obtain a better alignment.

For this script summarization, I worked on dialogue summarization based on a Sequence-to-Sequence (Seq2Seq) Method which is called MultiViewSeq2Seq [2]. It differs from most existing research based on single-speaker documents as said before. The key challenges with conversation summarization are that they are often informal, verbose and repetitive, sprinkled with false-starts, back channeling, reconfirmations, hesitations, speaker interruptions [12] and the salient information is scattered in the whole chat. In this project, we will focus on the specific conversational structures, in other words how are utterances always organized in order to make the conversation meaningful and understandable [12].This is what differs dialogues from structured texts: conversations have their own dynamic structures and one dialogue can be seen from different perspective. For instance, a conversation can be divided into the topics discussed and we call this perspective topic view, see Figure 1 where we can segment into *greetings*, *today's plan*, *plan for tomorrow*, *plan for Saturday* and *pick up time*. But it can also be seen from a progressive perspective that we call stage view, the same dialogue in Figure 1 can be then segmented into *openings*, *intention*, *discussion* and *conclusion*. Then, there are two more perspectives, global view where a discussion is seen as whole and discrete view where each utterance is seen as one segment.

It is important to combine all these different views. For example, models only focused on topic view may fail to capture the comprehensive and nuanced conversational structures which without a doubt would lead to errors in the decoding stage. Which is why, this method of dialogue summarization combines those diverse and multiple views in order to get more precise summaries. For the summarization, a multi view sequence to sequence model is designed and it consists in a conversation encoder which encodes different views and a multi-view decoder to generate dialogue summaries. The code of the whole project can be found on this repository `https://github.com/NicolasBOINAY/ScriptSummarization`

Conversation		Topic View	Stage View
James:	Hey! I have been thinking about you :)	Greetings	Openings
Hannah:	Oh, that's nice ;)		
James:	What are you up to?	Today's plan	Intention
Hannah:	I'm about to sleep		
James:	I miss u. I was hoping to see you		
Hannah:	Have to get up early for work tomorrow	Plan for tomorrow	Discussion
James:	What about tomorrow?		
Hannah:	To be honest I have plans for tomorrow evening		
James:	Oh ok. What about Sat then?	Plan for Saturday	
Hannah:	Yeah. Sure I am available on Sat		
James:	I'll pick you up at 8?	Pick up time	
Hannah:	Sounds good. See you then.		Conclusion

Figure 1: Example conversation from SAMSum (Gliwa et al., 2019) with its topic view and stage view

2 Data description

The used data comes from the ScriptBase corpus [6] (Corpus) which contains data for 1276 movies. This dataset contains different kind of data for each movie such as script, keywords, summary, synopsis... For my project, I will use mainly the script, the wiki plots and the summaries which are written by fans on IMDB.

I also use data from the TurningPoint dataset that gives me the screenplays segmented into scenes for 99 movies but also a python script to segment screenplays from other movies not in this dataset into scenes.

3 Method

Conversations can be interpreted from different views and every single view enables the model to focus a specific aspect of the conversation. To take advantages of those rich conversation views, I will base my method on the project of Multi-view Sequence-to-Sequence which first extracts different views of conversations and then encodes them to generate summaries. I will then explain this method.

3.1 Transformers

First, before explaining concretely the method, we will describe the different transformers that are used in the method.

First, we will focus on **BERT** [4], it is a transformer bidirectional encoder only, mapping a sequence of tokens to a sequence of d-dimensional vectors. It is pre-trained on unsupervised tasks including prediction of masked tokens and next sentence. It can take a number of sentences as input, where a sentence is an arbitrary span of contiguous text. This first explanation allows to introduce **Sentence-BERT** [11] which will be the transformer used to segment and encode conversations. It takes a single sentence as input and is trained by metric learning objectives, e.g. in a siamese or triplet structure, facilitating efficient sentence similarity search. It is learned by fine-tuning a pre-trained BERT model on supervised semantic textual similarity.

Then, we also use the **C99** algorithm [3]. This algorithm is also used to segment conversations. It takes a list of tokenized sentences as input then it measures the similarity between sentences using the cosine measure. Then it does a ranking of the sentences inside their local region (this is the number of neighboring elements with a lower similarity value). Finally, the last step is clustering which determines the location of the topic boundaries and so allows us to segment the conversation.

And finally, **BART** [9] is the transformer that allows us to decode and summarize conversations. It combines a bidirectional encoder and an auto-regressive decoder. It is pre-trained as an unsupervised denoising autoencoder, in other words, corrupting input text and learning to reconstruct the original.

3.2 Conversation view extraction

The issue with a conversation summarization model is that it can easily get lost among all different information coming from various speakers and utterances and even more when conversation becomes long what happens often in a movie script even if we divide it into scenes. Which is why, if it is structured in small blocks that represent the different informative structures, the model would then be able to understand them better. This is why we first have to extract different views of structures from conversations.

Topic View: Even if a conversation can be seen as unstructured compared to a formal document (news reports, encyclopedia articles), it is always organized around topics in a coarse-gained structure [8]. For instance, a telephone chat could possess a pattern of "*greetings* $\rightarrow$ *invitation* $\rightarrow$ *party details* $\rightarrow$ *rejection*" from a topical perspective. This segmentation in topics could allow a model to interpret conversations more precisely and then generate more precise summaries that can focus on more important topics. Here we combine the classic topic segment algorithm, C99 [3] that segments conversations based on inter-sentence similarities, with recent advanced sentence representations Sentence-BERT [11], to extract the topic view. Specifically, each utterance u_i in a conversation C $= u_1, u_2, ..., u_m$ is first encoded into hidden vectors via Sentence-BERT. Then the conversation C is divided into blocks $C_{topic} = b_1, ..., b_n$ through C99, where b_i is one block that contains several consecutive utterances, such as the topic view described in Table 1.

Stage View: As already said conversation is always kind of structured and the utterances are organized in a certain way. A first one is the topic segmentation, but another observation showed that conversations are found to follow a common pattern of "*introductions* $\rightarrow$ *problem exploration* $\rightarrow$ *problem solving* $\rightarrow$ *wrap up*" . This is called conversation stage view and it provides high-level sketches about the functions or goals of different parts in conversations, which could help models focus on the stages with key information. In order to extract stages from the dialogues, we follow Althoff et al (2016) [1] which consists in using a Hidden Markov Model (HMM). The observations in the HMM are the encoded representation h_i from Sentence-BERT. Then the number of hidden stages is set as 4 and as for topic view extraction, the conversation is segmented into blocks that contain several utterances. Figure 1 shows also a stage view from a dialogue.

Global and Discrete View: In addition to the two previously described structured

views, conversations can also be viewed from a relatively coarse perspective, i.e., a global view that joins all utterances into one giant block, and a discrete view that separates each utterance into a distinct block.

3.3 Multi-view Sequence-to-Sequence Model

To summarize conversations, we use the pre-trained model from the Multi-View-Seq2Seq project which extends generic sequence to sequence models to encode and combine the different conversation views. They implement their base encoders and decoders with a transformer based pre-trained model, BART [9]. The model was trained on the data contained in the SamSUM corpus [5]. We will now explain the way their model works.

Conversation Encoder: Given a conversation under a specific view k with n blocks: $\mathbf{C}_k = \mathbf{b}_1^k, ..., \mathbf{b}_n^k$ each token $x_{i,j}^k$ in a block $\mathbf{b}_j^k = x_{0,j}^k, x_{1,j}^k, ..., x_{m,j}^k$ is first encoded through the conversation encoder E , e.g., BART encoder as shown in Figure 2(a). Note that we add special tokens $x_{0,j}^k$ at the beginning of each block and use these tokens' representations to describe each block. To depict different views using hidden vectors, we aggregate the information from all blocks in one conversation through LSTM layers [7].

Multi-view Decoder: Thanks to the different views, the model had different types of conversational aspects to learn and after had to determine which set of utterances were the most important in order to generate better dialogue summaries. So the key for the decoder is to strategically combine different views. That is why a transformed based multi-view decoder is used as we can see in Figure 2(b). The input in the decoder is the previously generated tokens that will be predicted thanks to the multi-view decoder. But the difference with generic transformer decoder is that here a multi view attention layer is introduced in each transformer block and this layer decides the importance of each view. Then the multi-head attention is performed over conversation tokens $h_{i,j}^k$ from different views k and form A_k separately. The attended results are further combined and this is how summaries are generated.

A global illustration of the model is described in Figure 2(c).

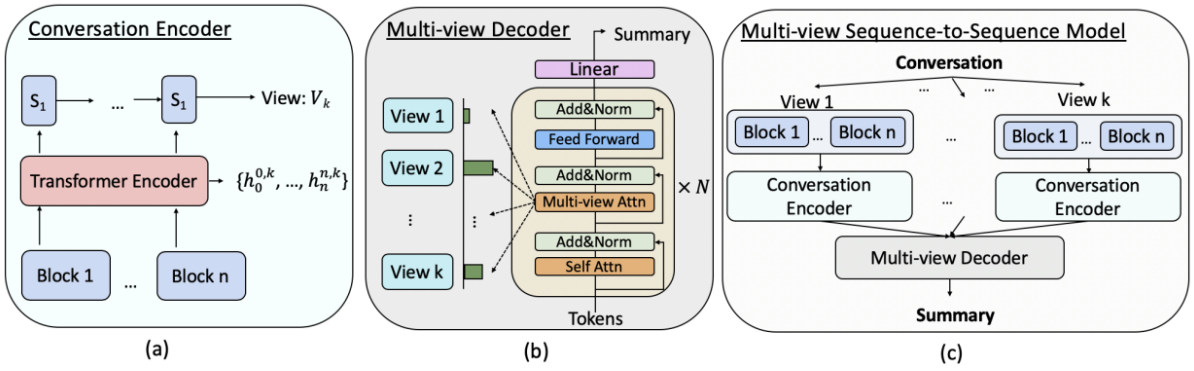


Figure 2: : Model architecture. Different views of conversations are first extracted automatically, and then encoded through the conversation encoder (a) and combined in the multi-view decoder to generate summaries (b).

4 Experiments and results

4.1 Experiment

In order to test this method of generating summaries based on movie dialogues, I used a movie from the ScriptBase corpus to run the experiment. I chose the movie Gladiator. In order to use the model some processing steps were needed. The purpose of this experiment is to generate summaries scene by scene in oas the alignment is made specifically between the scenes of the movie script and the sentences of the plot summaries. The first preprocessing step is then to segment our movie script into scenes using the TRIPOD method. Next, as the dataset permits us to retrieve movie scripts entirely only, we needed to retrieve only dialogues from the screenplays. In order to do so we had to write a Python script that is based on the pattern in each movie from Scriptbase which is before each utterance the name of the speaker is written in the middle of the line and is followed by what the speaker says again with a specific number of blank spaces from the left of the line. Then with the specific indentation of these movie scripts I could get a list containing all the spoken utterances from characters of the movie. An example of the result of these preprocessing steps is shown in figure 4. We can observe the list of string obtained for a specific scene, containing only the character names followed by their utterance. This format will be useful for the next step. Which is to segment the movie dialogue following topic, stage, general and discrete view. But before that it is needed to do the embedding of my utterances.

In order to do the embedding of the sentences and store all the embedding of the utterances in a pickle file. We used sentence-transformer. After that, the objective was to segment my conversations into topic views and stage views that were the two different segmentation that needed coding as global and discrete view are easy to get (it is just changing a list layout). For topic segmentation, it was mostly in two steps, first a step for taking out all the names of the utterances as topic segmentation only focuses on what is said. Then the second step was concretely segmenting into topics and to do so the C99 algorithm described earlier was used. This allowed to generate another pickle file of the utterances segmented into topics. For stage segmentation, it was different, we used the first pickle file generated with the embedding of the sentences and then a Hidden Markov Model, especially a Gaussian Hidden Markov Model with the hmm package from python. Then the last preprocessing operation generates the final preprocessed files for each view by changing the format for each different view to make it the same. And finally, in order to generate summaries, I had a final step of decoding the conversation summary using a BART model that was pre-trained and I took from the MultiViewSeq2Seq project.

EXT. HILL - TWILIGHT

The mighty catapults dwarf the humans. Soldiers from the elite Felix Regiment -- a legion of the Roman Army -- haul the monstrous machines up a hill.

The commanding General of the Felix Regiment, MAXIMUS, walks between two of the catapults. He is a striking and intense man in his 30's. Like all the soldiers who surround him, he is caked with mud and exhausted.

He trudges up the hill with his two lieutenants, TITUS and QUINTUS.

TITUS
You would do as well to read the mind of a rhinoceros.

QUINTUS
These barbarians would rather drown in blood than yield an inch. If I didn't hate them so much I would admire them.

They have reached the top of the hill. Stunning martial preparations are underway. The catapults join ten others. Archers are taking up position. Brutal "Scorpions" -- devices for firing multiple crossbow bolts -- are being loaded. Soldiers are also loading the catapults with enormous "Greek fire pots" -- large, round terra cotta pots.

Maximus and his lieutenants gaze down from the hilltop. Below them they can see a German encampment.

(a) First part of Scene 3

TITUS
They simply will not surrender.

A beat as Maximus gazes down at the German position.

MAXIMUS
(quietly)
A people should know when they are conquered.

A beat.

MAXIMUS
At the first signal release the catapults. We'll use the cavalry to cut off the retreat.

QUINTUS
General, I don't recommend that. Our cavalry might be caught in the flames.

MAXIMUS
I hope not, because I'm going to be leading them.

A beat as he gazes down at the enemy.

MAXIMUS
Why don't they know they're already dead?

(b) Second part of Scene 2

Figure 3: Scene 3 from movie Script

```
['       TITUS          You would do as well to read the          mind of a rhinoceros.',
"       QUINTUS          These barbarians would rather drown          in blood than yield
an inch.  If I          didn't hate them so much I would          admire them.", '
TITUS          They simply will not surrender.', '       MAXIMUS                (quietly)
A people should know when they are          conquered.', "       MAXIMUS          At the
first signal release the          catapults.  We'll use the cavalry to          cut off
the retreat.", "       QUINTUS          General, I don't recommend that.          Our
cavalry might be caught in the          flames.", "       MAXIMUS          I hope not,
because I'm going to be          leading them.", "       MAXIMUS          Why don't they
know they're already          dead?"]
```

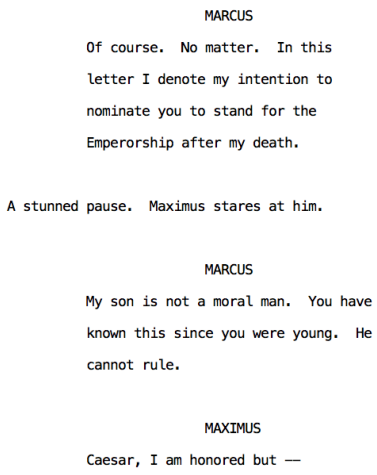
Figure 4: : Script of scene 3 after preprocessing. Only dialogues blocked into a list of strings.

4.2 Results

we conducted the experiment described and we managed to generate summary for each scene of the movie Gladiator. We obtained a narrative text of 74 sentences corresponding to the 74 scenes of the screenplay containing dialogue. Figure 8 shows the summary generated for the second scene of the movie whom raw script is in figure 7b. From the summary generated scene by scene from this movie and others, we could observe different results. First of all, the main issue is that when decoding and generating summary there is sometimes a mix between the names as in the dialogue speakers sometimes use name to address their selves to other characters or when in the script after the name it is said to whom the character is speaking. This conduct sometimes to summaries that invert the one talking and the one who they are talking to. For example, in Gladiator, in scene 18 Caesar tells Maximus he wants him to be the next emperor of Rome. But the summary inverts it and says Maximus wants Caesar to be the next emperor which does not make sense. See 56 Another thing is that, the length of the scene summary depends on the number of utterances inside the scene, which can lead to sense mistakes as in every movies, especially action ones, some scenes contain very few utterances but many things happen.

A good point for the summary scene by scene generated is that it is exhaustive and contains all the information contained in the plot summary such as locations, names and actions.

However, the global meaning of each scene is contained in the summary which lets us believe that it could be interesting to use it to align with plot summaries in a future work.



MARCUS

Of course. No matter. In this letter I denote my intention to nominate you to stand for the Emperorship after my death.

A stunned pause. Maximus stares at him.

MARCUS

My son is not a moral man. You have known this since you were young. He cannot rule.

MAXIMUS

Caesar, I am honored but —

Figure 5: : Ceasar nominates Maximus in script

MAXIMUS wants to nominate Caesar to stand for the Emperorship after his death. MARCUS will nominate him.

Figure 6: : Maximus nominates Caesar in summary

INT. WAGON - DAY

Mist momentarily obscures a man's face. Frozen breath. The man is in his 20's, imperious and handsome. He is swathed in fur, only his face exposed. He is COMMODUS.

He glances up.

COMMODUS
Do you think he's really dying?

The woman across from him returns his gaze evenly. She is slightly older, beautiful and patrician. A formidable woman.

She is LUCILLA.

LUCILLA
He's been dying for ten years.

COMMODUS
I think he's really dying this time.

A beat. Their breath turns instantly to mist.

COMMODUS
He has to be bled every night now.

LUCILLA
How do you know that?

COMMODUS
I've been so informed.

(a) First part of Scene 1

She arches an eyebrow.

COMMODUS
If he weren't really dying he wouldn't have sent for us.

LUCILLA
(a smile)
Maybe he just misses us.

COMMODUS
And the Senators. He wouldn't have summoned them if --

LUCILLA
Peace, Commodus. After two weeks on the road your incessant scheming is hurting my head.

A beat.

COMMODUS
The first thing I shall do is honor him with games worthy of his majesty.

LUCILLA
The first thing I shall do is have a hot bath.

The wagon rumbles to a halt. Voices are heard outside.

Commodus leaps out...

(b) Second part of Scene 1

Figure 7: Scene 1 from movie Script

COMMODUS and LUCILLA think he's really dying. He's been dying for ten years. He has to be bled every night now. COMMODUS has been informed about it. He will honor him with games worthy of his majesty.

Figure 8: : Summary generated for scene 1.

5 Conclusion

This project was conducted with the goal of tackling a NLP challenge which is automatic text summarization, but more particularly movie summarization which is an even more difficult challenge as it is less narrative than news reports for instance. The main objective was to generate summaries of scenes of a movie in order to align them with plot summaries inspiring us on the project made in proceedings ACL 2021 [10]. The alignment would allow us to extract most important scenes and then generate precise movie summary.

For this, I first wrote a script which allowed us to turn raw screenplays into a processed text ready to be used by MultiView Seq2Seq model to transform each dialogue of scene into a summary. The model was run on different movies which allowed us to implement the alignment with plot summaries and especially to compare the results with the initial method of aligning plot summaries directly with the movie scripts.

Future work would be running this experiment for the whole set of movies used in the Alignarr method and conclude with regards to the alignment task. It would also be running the experiment for all the movies in the summarization task and conclude with regards to summarization.

The code of the whole project can be found at `https://github.com/NicolasBOINAY/ScriptSummarization`

References

[1] Tim Althoff, Kevin Clark, and Jure Leskovec. Large-scale analysis of counseling conversations: An application of natural language processing to mental health. *In Transactions of the Association for Computational Linguistics,*, 2016.

[2] Jiaao Chen and Diyi Yang. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. *In EMNLP*, 2020.

[3] Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. *In 1st Meeting of the North American Chapter of the Association for Computational Linguistics.*, 2000.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, , and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *In Proceedings of NAACL*, 2019.

[5] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *In Proceedings of the 2nd Workshop on New Frontiers in Summarization.*, 2019.

[6] Philip John Gorinski and Mirella Lapata. Movie script summarization as graph-based scene extraction. *In Proceedings of NAACL-HLT*, 2015.

[7] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *In Neureal Comput.*, 1997.

[8] Axel Honneth and Hans Joas. Social action and human nature. *In CUP Archive*, 1988.

[9] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *In Proceedings ACL*, 2020.

[10] Paramita Mirza, Mostafa Abouhamra, and Gerhard Weikum. Alignarr: Aligning narratives on movies. *In Proceedings of ACL*, 2021.

[11] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *In Proceedings of EMNLP-IJCNLP*, 2019.

[12] Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn taking for conversation. *In Studies in the organization of conversational interaction*, pages 7–55, 1978.