Wentworth Institute of Technology
COMP 7800 – Natural Language Processing

Automated Text Summarization on Movie Dialogue Using Gensim

Sterling Pilkington
Faculty Advisor: Yetunde Folajimi

ABSTRACT

This study focuses on text summarization and its usefulness in producing more concise versions of documents. Previous studies have investigated patterns in movie character dialogue [2] as well as automated summarization of news articles, films, and documentaries [3]. Using a sentence ranking algorithm [4] and a dataset of over 220,000 lines of character dialogue in 617 movies [2], it was possible to create a 100-word summary of several movies in a few seconds. The algorithm performed well overall and showed most of the themes and plot points in several movies. There were only a few exceptions, such as minor word choice errors in the summaries.

Keywords: Natural language processing, movie, text summarization, Gensim.

I. INTRODUCTION

Text summarization, which was first researched in the 1950s [1], involves producing more concise versions of text while retaining all key points. It has many uses such as compressing sentences, search indexing (for example, Windows Search or Spotlight Search), creating abstracts of documents, and more. Although it is useful, this process is very time-consuming for humans to do since one has to read the entire document to develop a complete understanding [1]. On the other hand, it is difficult for computers because they lack knowledge about human language. However, they can be aided by natural language processing (NLP), which can allow computers to understand the world's languages.

Movie scripts are an example of documents that often need to be summarized. Screenwriters have to write a script synopsis to show to movie production companies and convince them that people will like it. Also, movie critics write summaries of new movies for magazines, newspapers, websites, and more. This helps people decide whether they want to watch the movie. Since summarizing scripts can be long, time-consuming, and biased, this study creates an automated process for generating summaries of them using Python and the Gensim summarize module [4].

II. RELEVANT WORK

A study performed by Danescu-Niculescu-Mizil *et al.* [2] focuses on how people adapt to each other's language styles during a conversation in order to gain approval or emphasize a different social status. This concept is compared to fictional dialogue in movies, specifically looking at how authors can create the conversations, but the benefits of language style adaptation are received by the characters. In addition, they also look at the effects of gender and other character traits. To do this, the researchers created a large dataset of movie scripts using a web crawler. The dataset includes lines, characters, and other movie metadata from the IMDb website.

Another relevant study looked at summarizing films using their subtitles and scripts [3]. The researchers compared the summaries generated by various algorithms (such as LSA, GRASSH., and LexRank) to news abstracts, documentary summaries, film plots written by humans. They found that news abstracts performed the best out of the three datasets.

These works inspired this study since it would be interesting to see how specifically character dialogue would perform when being summarized using a sentence ranking algorithm in Gensim [4].

III. IMPLEMENTATION

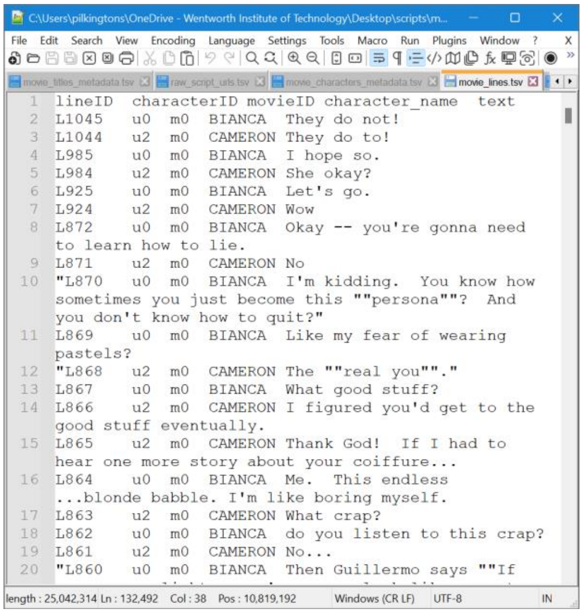
A. Dataset

The dataset being used is the one produced by [2] which includes over 220,000 conversational exchanges from 617 movies. This data is contained in several tab-separated value files (TSV). For example, movie_titles_metadata.tsv contains the movieID (to identify the movie in the other files) title, release year, the rating on IMDb, and the genres. Another file, movie_lines.tsv, holds data for the lines themselves. It has each character's name, the movieID to link each line to its movie, and the utterance.

B. Data Preprocessing

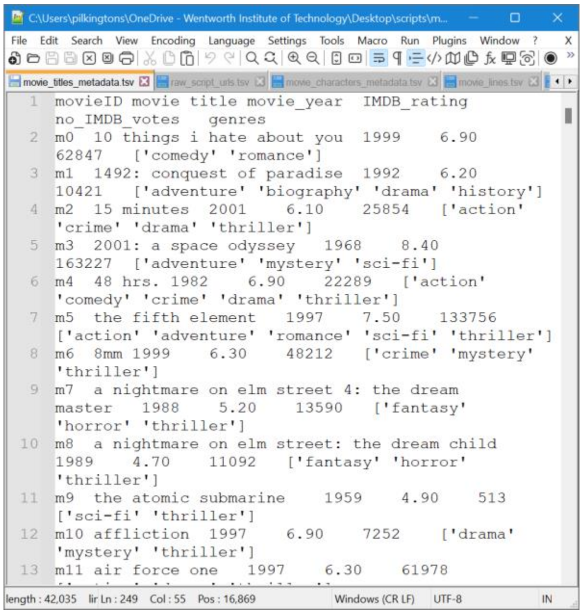
The original dataset did not have column headers in each TSV file. Instead, the columns were described in README.txt. To get the data ready for the Python code and make it slightly easier to manipulate, column labels were added back into the files (Figures 1 and 2 below). Also, since the character dialogue itself was of interest, only movie_lines.tsv was imported in the code.

When reading in the data using Pandas DataFrame, there were a few errors due to incomplete lines. Therefore, the on_bad_lines='skip' argument was used, which slightly reduced the number of available lines. Besides that, the dataset was ready for processing.


C:\Users\pilkingtons\OneDrive - Wentworth Institute of Technology\Desktop\scripts\m...
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X
movie_titles_metadata.tsv
raw_script_urls.tsv
movie_characters_metadata.tsv
movie_lines.tsv
length : 25,042,314 Ln : 132,492 Col : 38 Pos : 10,819,192 Windows (CR LF) UTF-8 IN

```
lineID  characterID movieID character_name  text
L1045   u0  m0  BIANCA  They do not!
L1044   u2  m0  CAMERON They do to!
L985    u0  m0  BIANCA  I hope so.
L984    u2  m0  CAMERON She okay?
L925    u0  m0  BIANCA  Let's go.
L924    u2  m0  CAMERON Wow
L872    u0  m0  BIANCA  Okay -- you're gonna need
to learn how to lie.
L871    u2  m0  CAMERON No
"L870   u0  m0  BIANCA  I'm kidding.  You know how
sometimes you just become this ""persona""?  And
you don't know how to quit?"
L869    u0  m0  BIANCA  Like my fear of wearing
pastels?
"L868   u2  m0  CAMERON The ""real you""."
L867    u0  m0  BIANCA  What good stuff?
L866    u2  m0  CAMERON I figured you'd get to the
good stuff eventually.
L865    u2  m0  CAMERON Thank God!  If I had to
hear one more story about your coiffure...
L864    u0  m0  BIANCA  Me.  This endless
...blonde babble. I'm like boring myself.
L863    u2  m0  CAMERON What crap?
L862    u0  m0  BIANCA  do you listen to this crap?
L861    u2  m0  CAMERON No...
"L860   u0  m0  BIANCA  Then Guillermo says ""If
```

Figure 1: Screenshot of movie_lines.tsv file

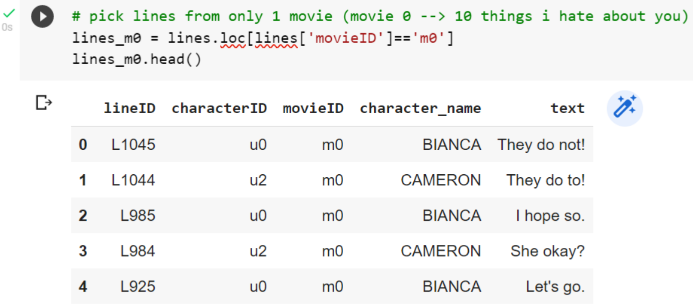

C:\Users\pilkingtons\OneDrive - Wentworth Institute of Technology\Desktop\scripts\m...
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X
movie_titles_metadata.tsv
raw_script_urls.tsv
movie_characters_metadata.tsv
movie_lines.tsv
length : 42,035 lir Ln : 249 Col : 55 Pos : 16,869 Windows (CR LF) UTF-8 IN

```
movieID movie title movie_year  IMDB_rating
no_IMDB_votes   genres
m0  10 things i hate about you  1999    6.90
62847   ['comedy' 'romance']
m1  1492: conquest of paradise  1992    6.20
10421   ['adventure' 'biography' 'drama' 'history']
m2  15 minutes  2001    6.10    25854   ['action'
'crime' 'drama' 'thriller']
m3  2001: a space odyssey   1968    8.40
163227  ['adventure' 'mystery' 'sci-fi']
m4  48 hrs. 1982    6.90    22289   ['action'
'comedy' 'crime' 'drama' 'thriller']
m5  the fifth element   1997    7.50    133756
['action' 'adventure' 'romance' 'sci-fi' 'thriller']
m6  8mm 1999    6.30    48212   ['crime' 'mystery'
'thriller']
m7  a nightmare on elm street 4: the dream
master  1988    5.20    13590   ['fantasy'
'horror' 'thriller']
m8  a nightmare on elm street: the dream child
1989    4.70    11092   ['fantasy' 'horror'
'thriller']
m9  the atomic submarine    1959    4.90    513
['sci-fi' 'thriller']
m10 affliction  1997    6.90    7252    ['drama'
'mystery' 'thriller']
m11 air force one   1997    6.30    61978
```

Figure 2: Screenshot of movie_titles_metadata.tsv file

C. Algorithm

After reading in the TSV file, the first step was to pick a movie script to be summarized. This was done by checking the movie_titles_metadata.tsv file and picking a movieID. Then, that was cross-referenced with the lines in movie_lines.tsv. The algorithm then gathered all lines from that movie and put them in a DataFrame, shown in Figure 3 below.



```
# pick lines from only 1 movie (movie 0 --> 10 things i hate about you)
lines_m0 = lines.loc[lines['movieID']=='m0']
lines_m0.head()
```

	lineID	characterID	movieID	character_name	text
0	L1045	u0	m0	BIANCA	They do not!
1	L1044	u2	m0	CAMERON	They do to!
2	L985	u0	m0	BIANCA	I hope so.
3	L984	u2	m0	CAMERON	She okay?
4	L925	u0	m0	BIANCA	Let's go.

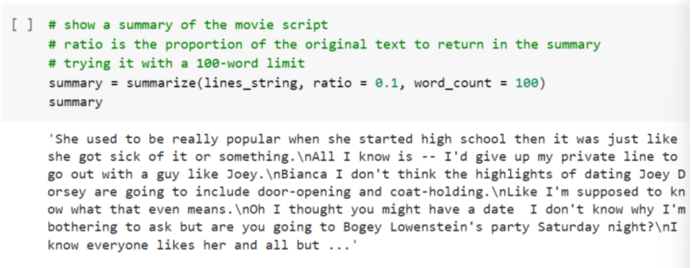
Figure 3: Screenshot of lines from 10 Things I Hate About You

Next, each line from the text column was extracted and concatenated into one long string, which was then fed into Gensim summarize. Gensim can take arguments such as ratio (the proportion of the original text to return in the summary) and the word count (length limit of the summary) [4]. For the purposes of this study, a ratio of 0.1 and a word count of 100 were used, to be able to fit examples into this paper.

IV. Results and Evaluation

To prove the code was working, four movies were summarized. Three of them were familiar: *Antz*, *Back to the Future*, and *Indiana Jones and the Temple of Doom*. Then, one was summarized that had not been seen before: *10 Things I Hate About You*. This way, it could be proven whether the generated output gave the user a good summary of the movie. The movie scripts had 320 lines, 324 lines, 364 lines, and 669 lines, respectively. Each movie took less than five seconds to be summarized using a Google Colab notebook.

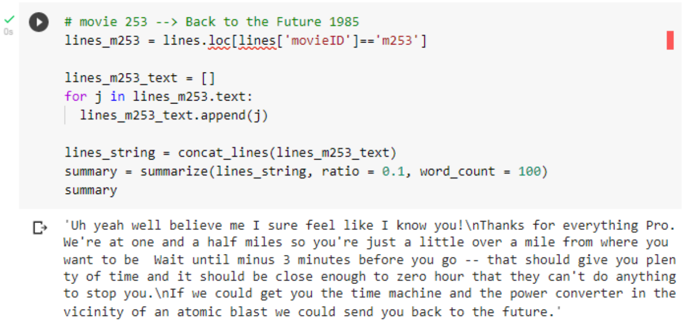
Figure 4 below shows what the summary output looks like for *10 Things I Hate About You*. Although the output has some grammatical errors, it gives the user a general idea of what the movie is, which in this case is a romantic comedy involving high school students.



```
[ ]  # show a summary of the movie script
     # ratio is the proportion of the original text to return in the summary
     # trying it with a 100-word limit
     summary = summarize(lines_string, ratio = 0.1, word_count = 100)
     summary

     'She used to be really popular when she started high school then it was just like
     she got sick of it or something.\nAll I know is -- I'd give up my private line to
     go out with a guy like Joey.\nBianca I don't think the highlights of dating Joey D
     orsey are going to include door-opening and coat-holding.\nLike I'm supposed to kn
     ow what that even means.\nOh I thought you might have a date  I don't know why I'm
     bothering to ask but are you going to Bogey Lowenstein's party Saturday night?\nI
     know everyone likes her and all but ...'
```

Figure 4: Generated summary of 10 Things I Hate About You (not seen before)

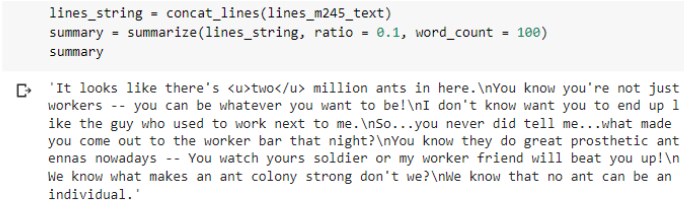
The algorithm worked well for each movie except *Back to the Future*, which it seemed to struggle with (shown in Figure 5 below). The summary referenced major plot points in the movie, but they lacked context and would be confusing to someone who had not watched it. For example, the sentence "Uh yeah well believe me I sure feel like I know you!" referenced how the main character traveled back in time and met his parents, who seemed to recognize him. At the time of this study, it was also unclear whether the quality or accuracy of the script in the dataset could have played a factor. Once again, not having this prior knowledge about the movie would make this summary confusing.



```
# movie 253 --> Back to the Future 1985
lines_m253 = lines.loc[lines['movieID']=='m253']

lines_m253_text = []
for j in lines_m253.text:
  lines_m253_text.append(j)

lines_string = concat_lines(lines_m253_text)
summary = summarize(lines_string, ratio = 0.1, word_count = 100)
summary

'Uh yeah well believe me I sure feel like I know you!\nThanks for everything Pro.
We're at one and a half miles so you're just a little over a mile from where you
want to be  Wait until minus 3 minutes before you go -- that should give you plen
ty of time and it should be close enough to zero hour that they can't do anything
to stop you.\nIf we could get you the time machine and the power converter in the
vicinity of an atomic blast we could send you back to the future.'
```

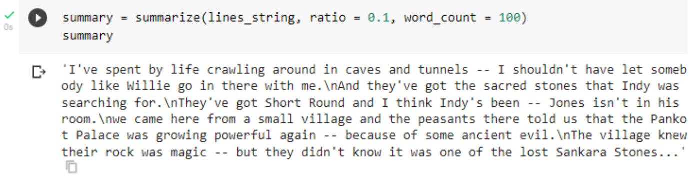
Figure 5: Generated summary of Back to the Future

Either way, the algorithm successfully provided a summary of each movie script tested with varying levels of usefulness. The output of *Antz* and *Indiana Jones and the Temple of Doom* can be seen below in Figures 6 and 7. Both of these summaries contain major plot points and give a clear idea of the stories.



```
lines_string = concat_lines(lines_m245_text)
summary = summarize(lines_string, ratio = 0.1, word_count = 100)
summary

'It looks like there's <u>two</u> million ants in here.\nYou know you're not just
workers -- you can be whatever you want to be!\nI don't know want you to end up l
ike the guy who used to work next to me.\nSo...you never did tell me...what made
you come out to the worker bar that night?\nYou know they do great prosthetic ant
ennas nowadays -- You watch yours soldier or my worker friend will beat you up!\n
We know what makes an ant colony strong don't we?\nWe know that no ant can be an
individual.'
```

Figure 6: Generated summary of Antz



```
summary = summarize(lines_string, ratio = 0.1, word_count = 100)
summary

'I've spent by life crawling around in caves and tunnels -- I shouldn't have let someb
ody like Willie go in there with me.\nAnd they've got the sacred stones that Indy was
searching for.\nThey've got Short Round and I think Indy's been -- Jones isn't in his
room.\nwe came here from a small village and the peasants there told us that the Panko
t Palace was growing powerful again -- because of some ancient evil.\nThe village knew
their rock was magic -- but they didn't know it was one of the lost Sankara Stones...'
```

Figure 7: Generated summary of Indiana Jones and the Temple of Doom

V. Future Work

There are many new directions that this concept of movie script summarization could be taken in. As seen above, one limitation of the methods in this study is that movie character metadata is not considered in the summary. A character name will only appear if it is uttered in the movie, however performance could be improved if an algorithm were to incorporate such metadata. The dataset also includes gender and character ranking based on their position in the end credits. This information would help determine who is a main character and may provide more meaningful summaries. One could also change argument values such as word count and ratio for Gensim to see how those affect the quality of the summaries.

Overall, these methods prove that movie script summarization is possible and useful. They can also be applied to other texts such as books, research papers, lecture notes, news articles, and more.

References

[1]
M. Allahyari *et al.*, "Text Summarization Techniques: A Brief Survey," vol. 8, no. 10, 2017, doi: 10.14569/IJACSA.2017.081052.

[2]

C. Danescu-Niculescu-Mizil and L. Lee, “Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs,” Jun. 2011 [Online]. Available: https://arxiv.org/abs/1106.3077

[3]
M. Aparício, P. Figueiredo, F. Raposo, D. Martins de Matos, R. Ribeiro, and L. Marujo, “Summarization of films and documentaries based on subtitles and scripts,” vol. 73, pp. 7–12, Apr. 2016, doi: 10.1016/j.patrec.2015.12.016. [Online]. Available: https://dx.doi.org/10.1016/j.patrec.2015.12.016

[4]
R. Řehůřek, “summarization.summarizer – TextRank Summariser,” 01-Nov-2019. [Online]. Available: https://radimrehurek.com/gensim_3.8.3/summarization/summariser.html