Analyzing Movie Scripts as Unstructured Text

Conference Paper · April 2017

DOI: 10.1109/BigDataService.2017.43

3 authors, including:

hye yeon yu
Yeonsung
9 PUBLICATIONS **14** CITATIONS

SEE PROFILE

Analyzing Movie Scripts as Unstructured Text

Seong-Ho Lee, Hye-Yeon Yu, and Yun-Gyung Cheong
College of Software
Sungkyunkwan University
Suwon-si, South Korea
sk1528@skku.edu, yu0529@skku.edu, aimecca@skku.edu

***Abstract*—This paper describes our initial attempt to analyze movie scripts for understanding patterns and narrative flow that can be present in storytelling. We collected movie scripts, processed the data using simple natural language processing and machine learning techniques. The result suggests that analyzing big movie script may reveal patterns related to narrative structure.**

Keywords—movie, script, scenario, data analysis, narrative

I. INTRODUCTION

As movies and TV series contents have been created extensively and easily accessible by content providers such as Netflix and Amazon, retrieving contents appropriate for the user preference has received much attention recently. A basic approach to solve this problem is to use the movie metadata (i.e., the genre, the director, actors/actresses, camera angle, lighting, sound, music, etc.) [1] to recommend similar movies that share the meta information. On the other hand, with the availability of the user's movie ratings and behavior data such as Movielens datasests [2] and the dataset used for Netflix Prize [3], collaborative filtering and machine learning techniques have been actively used to predict the user preference about movies [20]. Finally, Natural Language Processing of text reviews and tweets on SNSs (Social Network Services) have been recently utilized for recommendation and box-office prediction [4, 17, 19]. However, these types of data-driven approaches have a drawback which requires crowd-sourcing lots of users' ratings and reviews enough to predict the target user's preference. Imagine, for example, a cold-start problem case where a new movie is just released and few people have seen the movie. The movie would not be recommended to the target user even if it fits her/his preference.

To fill in the gap, we hypothesize that analyzing movie scripts may The goal of this paper is to examine text-based scenarios from the statistical and machine learning perspectives to unveil interesting facts underlying stories and movies. An American novelist, Kurt Vonnegut, noted that in one lecture, a graph of the luck and misery experienced by a hero can be used to find a particular trend along with a curve [10]. This assertion has been experimented in the recently published Reagan study [11], where the curves the emotions of over 2,000 novels were analyzed and categorized into six storytelling patterns. This article presents a work-in-progress paper that describes the data collection process and statistical analysis process focusing on emotional expressions found in the scenarios.

II. DATA COLLECTION AND PREPROCESSING

This section discusses the data collection method used in this study to obtain 978 movie scenarios. The scenarios used in this study were collected movie scenarios provided by IMSDB (The Internet Movie Script Database) site [5]. The site contains the largest number of movie scenarios along with various metadata such as the opening date of each movie, genres, writers, and average ratings. IMSDB does not provide APIs that allows us to access to the scenarios stored in the database. Each movie script is in the HTML format which contains scenario-unrelated contents such as online advertisements, web user interfaces, menus, and logos.

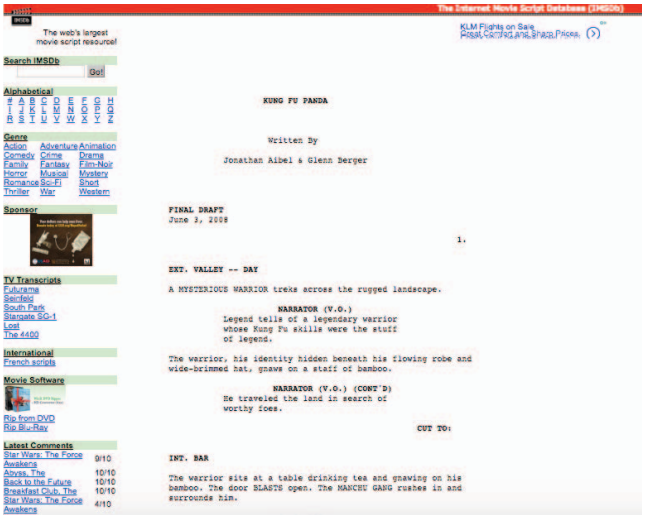

Fig. 1. IMSDB webpage showing the Kung-Pu Panda script

In order to collect the scenarios, we used the *BeautifulSoup* 4.4.1 library—a Python HTML Parsing Tool [6]. The library provides the ability to specify the desired tag and class in the HTML document of the desired Web site, parse the value, and display it as a Python dataset. Figure 2 illustrates the collection process. In advance, we prepared a list that contains movies that are found in the IMSDB. The first step requests the movie scenarios on the list using the PATH information in which the scenarios of each movie are present. Second, the received

978-1-5090-6318-5/17 $31.00 © 2017 IEEE
DOI 10.1109/BigDataService.2017.43

HTML page is parsed using the *BeautifulSoup* library to eliminate HTML tags and useless information such as advertisements as the information can contaminate our data, which can cause a large error in extracting the emotion degree data of the scenario. Finally, the parsed data is refined using regular expressions and stored as a file. The refinement process exclusively removes unnecessary sentences using regular expressions.

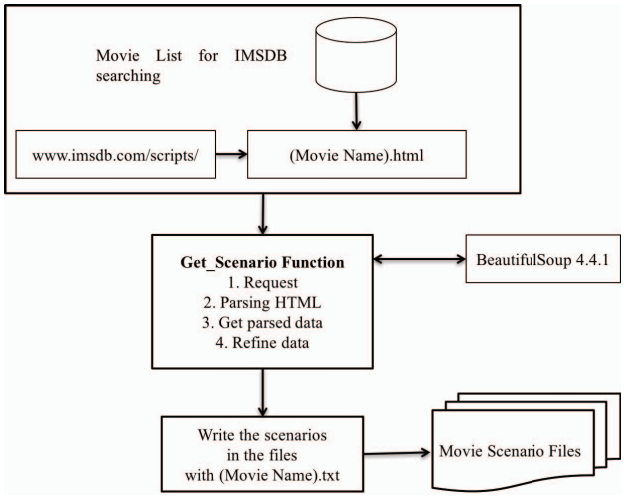

Fig. 2. The data collection process

In addition, the characters generated in error in the HTML parsing process or the special characters included in the IMSDB scenario itself may cause an encoding problem in the operation of the program, and the method of removing the characters in this step is taken in advance. A movie script details the characters' dialogues, actions, and behavior directions as well as camera directions. Individual screenwriter has his/her own style to denote the change of the scene, the change of the place, the name of the person, and the psychological change of the person. The scenario data we obtained exhibit no clear rules to indicate such information (see Figure 3). The scenario text was unstructured, which made it difficult to extract the exact part of individual scene and character.

III. Natural Language Processing

For natural language processing, we utilized NLTK (Natural Language Toolkit) open source Python toolkit [7]. NLTK provides corpus for different languages and text processing functionalities such as tokenization, parsing, and classification. Emotional vocabulary dictionaries were also used for sentiment analysis; SentiWordNet [8] and LIWC (Linguistic Inquiry and Word Count) [9] are widely used to extract the polarity of words (positive, negative, neutral). SentiWordNet is a vocabulary dictionary annotate a word with numerical values that represent the degrees of positive, negative, and neutral words. The SentiWordNet dictionary was constructed by applying semi-supervised learning algorithms and random walk to the data of WordNet [6].

1.

EXT. VALLEY -- DAY

A MYSTERIOUS WARRIOR treks across the rugged landscape.

NARRATOR (V.O.)
Legend tells of a legendary warrior
whose Kung Fu skills were the stuff
of legend.

The warrior, his identity hidden beneath his flowing robe and wide-brimmed hat, gnaws on a staff of bamboo.

NARRATOR (V.O.) (CONT'D)
He traveled the land in search of
worthy foes.

CUT TO:

INT. BAR

The warrior sits at a table drinking tea and gnawing on his bamboo. The door BLASTS open. The MANCHU GANG rushes in and
surrounds him.

GANG BOSS
(to warrior)
I see you like to CHEW!
(beat)
Maybe you should chew on my FIST!!

The Boss punches the table.

NARRATOR (V.O.)
The warrior said nothing for his
mouth was full. Then, he swallowed.

He swallows.

NARRATOR (V.O.) (CONT'D)
And then, he spoke.

WARRIOR
(dubbed hero voice)
Enough talk. Let's FIGHT!
SHASHABOOEY!

WHAM! The warrior delivers a punch and the whole gang goes flying.

NARRATOR (V.O.)
He was so deadly in fact that his
enemies would go blind from
overexposure to pure awesomeness.

Fig. 3. The processed movie script of Kung-Pu Panda, directed by Mark Osborne and written by Jonathan Aibel,Glenn Berger

The SentiWordNet 3.0.0 contains about 117,000 lines, each of which contains a set of words that denote similar meanings, positive score, and negative score between 0 and 1. For instance, the word 'deadly in the script text of Figure 3 is annotate with 0.0 positive score and 0.5 negative score.

For analysis, our preprocessing step eliminated unnecessary whitespace in the text. Then, the scenario was divided into units called blocks in this study to compute the positive value of each block. Ideally, one scene should constitute the basic unit. However, since distinct movie scripts have different ways of exhibiting the scene information, we determined that one block contains 20 lines via experimentations. One block was again parsed into individual words using the NLTK toolkit. Then, the word set of a block is compared with the SentiWordNet word set, and the sentiment value of each emotion block is derived. The emotion score of each word is derived based on the positive score among emotion values specified in SentiWordNet. When a word contains multiple meanings and emotional values, the affect value of that word is assigned via ranking. The ranking process considers the weight of the score according to the formula provided by SentiWordNet. Finally, we define the positive value of a block as the sum of the positive values of the words contained the block divided by the total number of sentiment words in the block.

IV. Data Visualization

We illustrated the sentiment values of movie scripts as graphs to find their narrative structures, and the graphs were spiky and all pointed. To smooth the graph to a continuous shape, we applied a moving method that regards several blocks as one window unit. For example, when 4 lines constitute one block and one window is defined as five blocks window 1 contains scene 1 to scene 5, and window 2 contains scene 2 to scene 6. As the number of blocks constituting one window increases, the graph becomes more gentle. Fig. 4 shows an example when four lines constitute one block, 20 lines (4 lines * 5 blocks = 20) form one window.

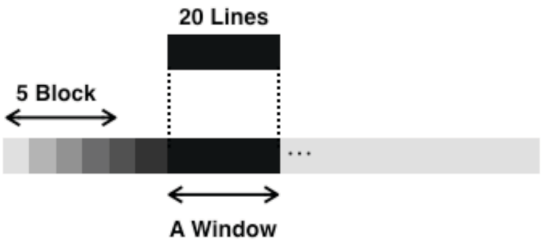

Fig. 4. Window sliding of 5 blocks for data visualization

Figure 5 exhibits the positive sentiment graph of the film 'V for Vendetta.' The x axis represents the scene progression, and the y axis represents the average value of positive score of each window. The closer to 0 on the y axis, the lower the positivity of the window which contains 5 blocks. The maximum positive value can be 1, but impossible in practice because it means that all the words in the corresponding 5 blocks (100 lines in our study) should have the maximum positive score. In fact, the maximum positive score in the SentiWordNet 3.0.0 is 0.875, of which category include the words nice, kind, lucky, legendary, intellectual, and formidable.

Figure 6 illustrates the level of positive sentiment score of the film '127 hours', of which story is about a mountaineer's struggle with a hand stuck in a rock. This graph shows frequent changes of sentiment score.

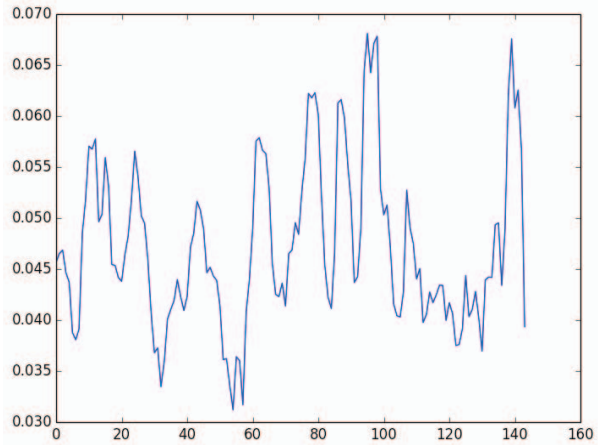

Fig. 5. Graph of the film 'V for Vendetta', directed by James McTeigue

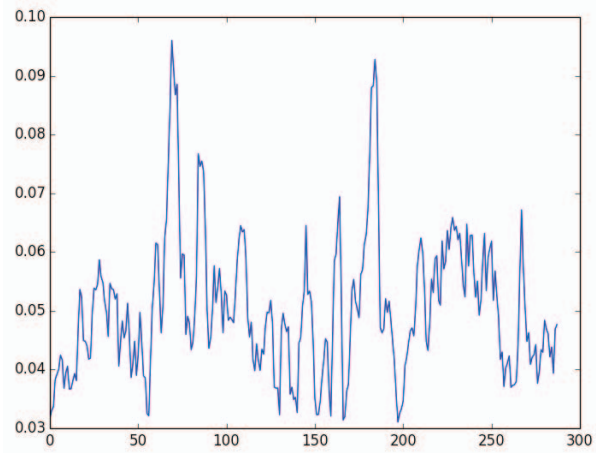

Fig. 6. Graph of the film '127 hours', directed by Danny Boyle

V. Data Smoothing

The visualization of the data shows that a scenario sentiment data (which is smoothed with window sliding) is still spiky, which makes us difficult to see patterns. In order to determine an efficient smoothing method for a scenario representation, we applied a set of signal smoothing algorithms to the scenario sentiment raw data where each value represents a scene's average of positive sentiment scores.

A. *Spline Interpolation*

Spline interpolation is a technique of dividing a whole section into small sections and making a low-order polynomial into a smooth curve, which is also referred to as a piecewise polynomial interpolation. Spline interpolation methods are linear spline, quadratic spline, and cubic spline. In this study, smoothing works were performed using B-Spline Curve. The spline interpolation method provides an excellent approximation to the data showing the behavior of a sudden change locally. In this study, B-Spline interpolation filtering is implemented through interpolate library which is sub-package of interpolation existing in the Scipy module [13].

B. *LOWESS(Locally Weighted Scatterplot Smoothing)*

LOWESS (Locally Weighted Scatterplot Smoothing) adds a local weight to the data and uses it to create a linear or quadratic curve that best fits the scattered data [15]. The LOWESS smoothing in our study was performed using the Python Statsmodels library [18].

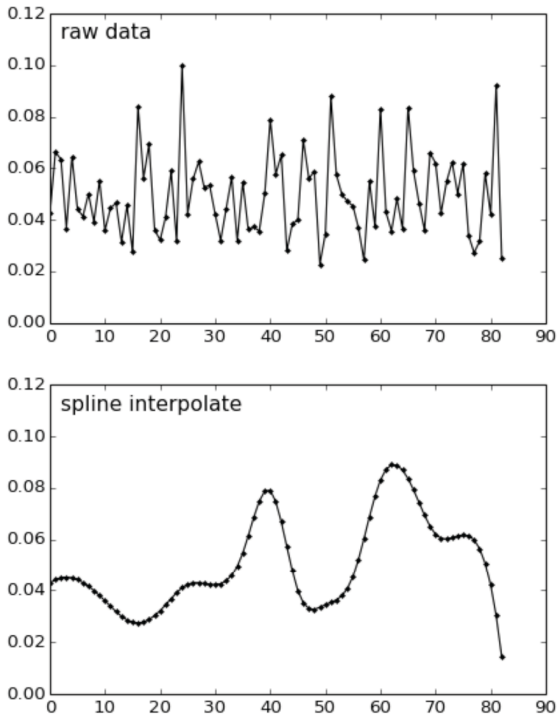

Fig. 7. Graphs of the film 'Kung-Fu Panda', with a Spline interpolation filtering applied.

C. *Savitzky Golay Filter*

The Savitzky Golay filter is a type of data smoothing, a data filter that can be applied to increase the signal to noise ratio ensuring that significant distortion is not made to the signal. The Savitzky Golay filter finds a polynomial-regression expression Si with 2n + 1 adjacent points including smoothed data Xi and applies a smoothing effect to the data by replacing Xi with Si (Xi).

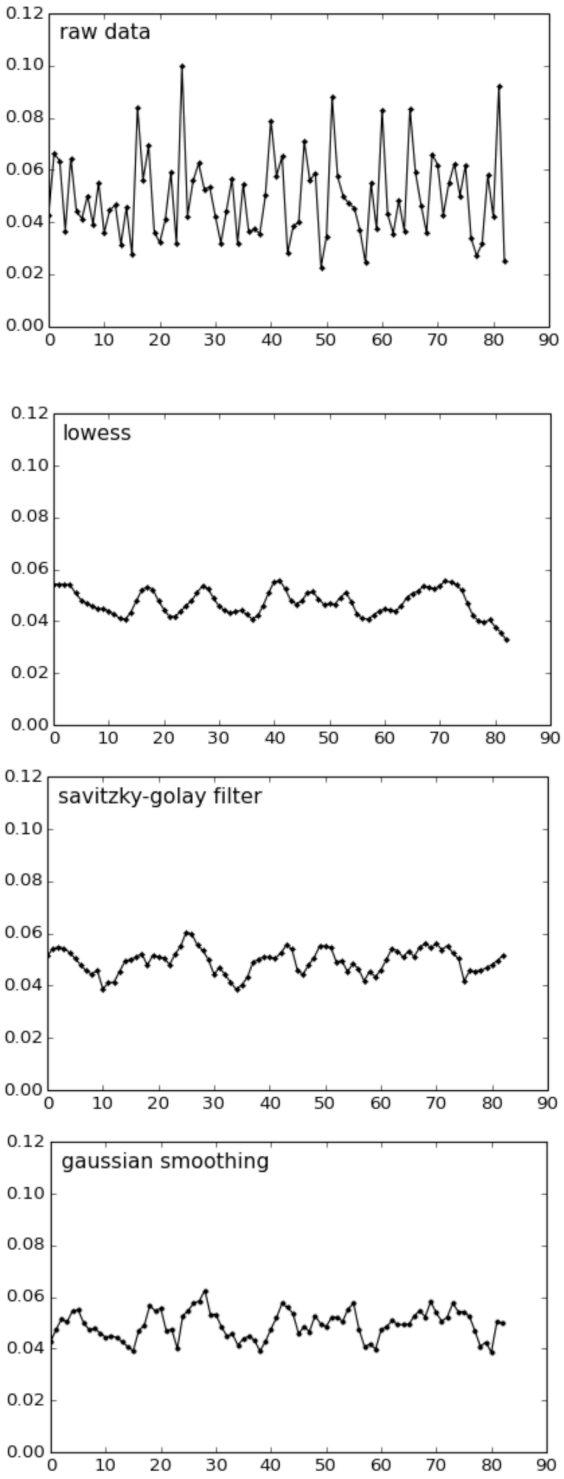

Fig. 8. Graph of the film 'Kung-Fu Panda' with , Savitzky-Golay filtering, LOWESS, and Gaussian smoothing filtering applied.

The calculation of the value of Si (Xi) takes the form of a convolution and can be easily obtained by reducing the number of calculations. In this study, we used the savgol_filter function provided by the scipy 0.18.1 version [14]. The size of the Savitzky Golay filter window was set to 13, and the polynomial applied to the sample is the third polynomial. The filter window size 13 was determined via experimentation. The value can only be odd values, and too large or too small window size led to similar or severe distortion to the original data.

D. *Gaussian* Smoothing Filtering

We then applied the commonly used Gaussian function to identify important trends in repeated statistics. The one-dimensional Gaussian function is the probability density function of the normal distribution.

(Eq. 1) $$f(x) = ae^{-(x-b)^2/(2c^2)}$$

where $a = \frac{1}{\sigma\sqrt{2\pi}}$, $b = \mu$, $c = \sigma$

In the normalized Gaussian curve, μ is the expected value and the Gaussian probability distribution with the deviation is expressed by the following graph. $c = \sigma$ corresponds to the peak of the graph which determines the width of the Gaussian distribution. The larger the value, the gentler the curve becomes. We implemented the Gaussian smoothing using the the source at [16].

E. *Discussions*

The results of filtering through different smoothing algorithms show different shapes and characteristics (see Figure 7 and Figure 8). Therefore, it is important to consider what smoothing results can be said to be 'effective'. The goal of smoothing in this study was to foresee the patterns to determine if analyzing text movie scripts would be useful for recommendation and to uncover hidden patterns underneath storytelling. Therefore, the smoothing result which showed the tendency in the form of simplified graph in the line which does not deviate much from the original data seems appropriate.

From this point of view, the results of the spline interpolation were judged to be unfavorable due to the large difference from the original data. The results obtained through the other three algorithms show similar. With the application of LOWESS, the overall data tendency of the data appears as a smooth curve, but the noise of the original data is unnecessarily neglected compared to the results of the other two algorithms. On the other hand, the data applied with Savitzky-Golay and Gaussian smoothing method maintain the characteristics of the original data. Hence, we consider LOWESS, Savitzky-Golay, and Gaussian smoothing methods need further experimentation and comparative analysis.

VI. CLUSTERING THE DATA

Furthermore, we extracted two features of each scenario (the relative position of the lowest positive score of a window in a scenario and the relative position of the highest positive score of a window in a scenario) of all the 978 scenarios, and ran a SimpleKMeans algorithm using the WEKA toolkit setting the number of clusters as six, and obtained the graph shown in Figure 7. Since a few instances belong to cluster 1 (3% of the dataset, see TABLE 1), we regard these instances as in cluster 0. Then, the result shows that there are 5 clusters, and the most dense clusters are cluster 0 and 4. Looking further into the feature values of the scenarios in the cluster 0, the minimum positive point of story occurs within the latter half and the maximum positive point occurs in the beginning part. This may indicate that the story begins as happy and ends as unhappy. On the other hand, the scenarios in the cluster 4 featured that the maximum point occurs in the middle of the story and the minimum positive point occurs in the end. This may mean that the story was happy in the middle and unhappy in the end. Both clusters appear to represent tragedy, and yet, we have not confirm if individual or an exemplar story in those cluster fall in tragedy. The graph also shows that there are few instances that fall in cluster 5, suggesting that both the positive and the negative points occur in the end of the story.

TABLE I. NUMBER OF INSTANES FOR EACH CLUSTER

Cluster Id	0	1	2	3	4	5
Instances	206	27	183	188	222	152
Percentage	21%	3%	19%	19%	23%	16%

Fig. 9. The result of clustering of 978 scripts using the relative positions of the minimum and maximum postitive values of each scenario. The blue dots denote the scenarios in cluser 0, the red dots denote the scenarios that fall in cluster 1, the green dots denote the scenarios in cluster 2, the light blue dots denote cluster 3, the peach dots denote the scenarios in cluster 4, and blue dots the scenarios in cluster 5.

VII. CONCLUSIONS

To summarize, we collected 978 movie scripts in English, each of which was processed to extract positive sentiment scores for predefined text units—called block and window in our study. To find some patterns, we attempted to project each movie script's sentiment values as a form of a lined graph, and then applied a set of smoothing algorithms. The results seem to show some patterns, although not conclusive at this point. We will continue to make an effort to explore and expand the dataset.

In addition, we scattered the 978 scenarios on the two dimensional planes which represent relative position of the lowest positive value (x axis) and the highest positive value (y axis) respectively. When grouped into 6 clusters, it is suggested that there are two popular patterns of story progressions. We plan to look into example scenarios in these clusters to test whether the story progresses as anticipated—happy moments in the beginning or middle that ends unhappily.

From this study, we learned that the movie scripts are unstructured texts, although they initially looked semi-structured. Therefore, a set of lines were used as a story unit in this study, we feel the need to implement an NLP algorithm to allow scene separation as scene is the basic unit of movie scripts. Second, we learned that movie scripts are temporal and spatial data. It is temporal because the story progresses as the reader reads it with some exceptions where story time flows back and forth as a narrative device to make the story enjoyable [12]. There can be treated as spatial data since the characters and text consume space when printed. Finally, although it is not conclusive in this study, there are possibilities that analyzing the movie scripts may reveal interesting patterns or narratological information in storytelling such as tragedy. We are particularly interested in the narrative structure.

Future work includes manual coding of movie information such as types of ending (sad or happy) and the climax moment. We will inspect LOWESS, Savitzky-Golay, and Gaussian smoothing methods for better representation of the data for narrative pattern prediction when using advanced machine learning techniques.

ACKNOWLEDGMENT

This research was supported in part by the MISP(Ministry of Science, ICT & Future Planning), Korea, under the National Program for Excellence in SW (R71151610060001002) supervised by the IITP(Institute for Information & communications Technology Promotion) and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2016933002).

REFERENCES

[1] Choi, S. M., Ko, S. K., & Han, Y. S. (2012). A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications*, *39*(9), 8079–8085. https://doi.org/10.1016/j.eswa.2012.01.132

[2] Movielense datasets. https://grouplens.org/datasets/movielens/

[3] Netflix Prize. http://www.netflixprize.com/

[4] Hur, M., Kang, P., & Cho, S. (2016). Box-office forecasting based on sentiments of movie reviews and Independent subspace method. *Information Sciences*, *372*, 608–624.

[5] The Internet Movie Script Database. http://www.imsdb.com/

[6] BeautifulSoup. https://www.crummy.com/software/BeautifulSoup/bs4

[7] NLTK: The Natural Language Toolkit (2002) by E. Loper, S. Bird

[8] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani, "SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining", LREC, 2010.

[9] Pennebaker, J. W., Booth, R. J., & Francis, M. E. (2007). Linguistic Inquiry and Word Count: LIWC [Computer software]. Austin, TX: LIWC.net.

[10] Kurt Vonnegut Jr., The Shapes of stories.

[11] Andrew J. Reagan, Lewis Mitchell, Dilan Kiley, Christopher M. Danforth, Peter Sheridan Dodds, "The emotional arcs of stories are dominated by six basic shapes", 2016.

[12] Bae, B.-C., Cheong, Y.-G., and Vella, D. H. Modeling Foreshadowing in Narrative Comprehension for Sentimental Readers. LNCS 8230: Interactive Storytelling, Koenitz et al. (Eds.), pp. 1-12, In the Proceedings of the 6th International Conference on Interactive Digital Storytelling (ICIDS), Istanbul, Turkey, November, 2013.

[13] Interpolation. https://docs.scipy.org/doc/scipy-0.18.1/reference/interpolate.html

[14] scipy.signal.savgol_filter. https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.signal.savgol_filter.html#scipy.signal.savgol_filter

[15] Cleveland, W. S. (1981) LOWESS: A program for smoothing scatterplots by robust locally weighted regression. *The American Statistician* 35, 54.

[16] Gaussian smoothing. http://www.swharden.com/wp/2008-11-17-linear-data-smoothing-in-python/

[17] Chiang, I. P., Wen, Y. F., Luo, Y. C., Li, M. C., & Hsu, C. Y. (2014). Using text mining techniques to analyze how movie forums affect the box office. *International Journal of Electronic Commerce Studies*, *5*(1), 91–96.

[18] Statsmodel library. http://statsmodels.sourceforge.net/

[19] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Empirical Methods in Natural Language Processing (EMNLP)*, *10*(July), 79–86.

[20] Schafer, J. Ben, Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. *International Journal of Electronic Business*, *2*(1), 77.