Movie-Spin: Automatic Extraction of Structured Information from Movie Plots

Rosangel García
Le Moyne College
Syracuse, NY 13214

Vicente Ordóñez
University of Virginia
Charlottesville, VA 22904

Abstract

Automatically browsing data collections containing thousands of records with unstructured information such as textual data is challenging for humans. In this project we propose to automatically analyze thousands of texts corresponding to movie plots in order to extract structured information in the form of the main characters for each movie, and their sentiment polarity in each movie. We leverage existing research in natural language processing such as named entity recognition, and sentiment analysis to obtain this information. Additionally we present Movie-Spin a browsing platform that incorporates this information in addition to other movie statistics, therefore potentially allowing for easy browsing for thousands of movies in a single platform. We present a proof of concept for this platform and conduct basic statistic analysis on a large movie plot dataset consisting of 5000 movies.

Introduction

Current research in movie analysis mostly focuses on movie recommendation systems where users are suggested new content based on statistical machine learning models or data mining techniques. There has been less research devoted to browsing movies, where users are allowed to freely browse a large movie collection at will. Browsing large collections of information however can be difficult. We propose Movie-Spin a browsing system that presents users with listing of movies that include its movie poster, and a word-cloud based on textual plots extracted from Wikipedia. Additionally we present a method that leverages natural language processing (NLP) to automatically extract the main characters in a movie, and a sentiment analysis of the main characters' actions in the movie. We incorporate this information as well in Movie-Spin.

In this project we specifically worked with unstructured data from a data set of movies, aiming to to find a way to use natural language processing techniques to structure the data. We used the Natural Language Processing Toolkit (NLTK) Library (Bird and Loper 2004) to extract information from text files using an array of techniques such as tokenization, part of speech tagging, and name entity recognition.

Final Project Report for the CRA-W DREU program. Summer 2018 at the University of Virginia. Submitted: Oct 5, 2018.

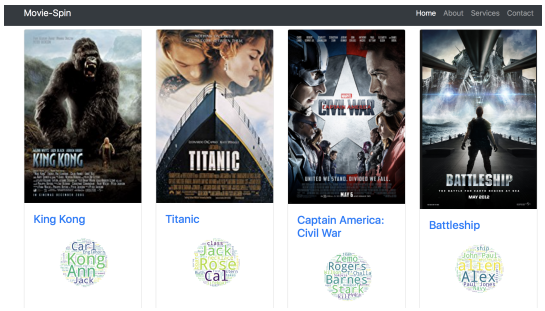

Figure 1: Screenshot of the Movie-Spin browsing tool built for this project, showing both images (movie posters) and a word cloud summary of the movie plots extracted from Wikipedia. We used 5,000 movies containing movie plots and metadata in our experiments.

Related Work

The methodology used in this project was inspired by current research in movie recommendation and processing systems such as MovieLens (Miller et al. 2003) or others (Wei et al. 2016), which help a user find movies that they would like to watch. You are able to rate movies so that you can build a custom taste profile. After, MovieLens will be able to recommend other movies for the user to watch. The user is able to learn more about the movies recommended with the rich data of images,and trailers. In our project, instead of focusing on movie recommendation, we focus on enhanced movie browsing through the use of information extraction techniques to enhance the user experience. The user is able to browse movies and quickly glimpse at the plot of the movies through word clouds, and read about the important characters in the movie through automatic extraction of main characters for each movie.

Background

The Natural Language Processing ToolKit (NLTK) Library is a text processing library that allows you to work with linguistics and natural language to analyze linguistic structures and corpora. From this library Name Entity Recognition was also used. Name Entity recognition(NER) is also known as entity identification, entity chunking or entity ex-

Figure 2: Common words for all movies

traction. The goal of NER systems is to identify named entities in freely structured text. Named Entities refer to definite noun phrases that refer to specific types such as these predefined categories like name of persons, organizations, locations, expressions of time, quantity, monetary values, percentages, etc. In this project we also worked with a concept called Sentiment Analysis which is also included in the Natural Language Processing library. It can categorize into categories such as positive, negative, and neutral. In order to categorize, there is a need to classify data and to do that, the data has to be trained. The classifier uses the training data in order to make predictions. We used the Vader Sentiment Analysis pre-trained model, which is included in NLTK. It is a tool that automatically generates positive, negative, and neutral sentiment scores for a given input. This tool analyzes a piece of text and depending what words are used then it will give these sentiment scores to that piece of text. We included in this paper figures showing the results obtained after using these libraries. Figure 1 shows how the website looks with results for the first 100 movies. Figure 2 shows a word cloud showing the most frequent words in all movie plots in all the 5,000 movies in the data set all together. Figure 3 shows the poster of the movie Titanic which I discuss in this paper. Figure 4 shows the word cloud for this movie. Figure 5 shows also the poster of another movie I discuss in this paper being the Dark Knight. Figure 6 shows the word cloud for this movie. Figure 7 shows the top characters for the movie Titanic. It gives an example of how the result of NER (Named Entity Recognition) looks like. Both figures 8 and 9 show the sensitivity analysis and top two characters for both movies. Figure 10 shows the frequency of movie characters (a person in a movie). Figure 11 shows them in a graph. Figure 11 shows a plot with the common words for all movies in the data set.

Methodology

We have a data set of movie plots which are text files of what each movie in the data set is about. These were originally crawled from Wikipedia and included in the CMU Movie Corpus (Bamman et al. 2013). For those files we used a word cloud library[1] to develop a word cloud with the most frequent words mentioned in every file for each movie. We generated programmatically a word cloud for each movie. After doing this, we used Name Entity Recognition (NER) in order to find all the people that were mentioned in each file. We wanted to know what characters (persons in the movie) were mentioned. We use the output of the NER subsystem and filter out any entities that do not belong to the category PERSON. This library also allowed us to see how many characters were mentioned in each plot file. For example one of the movie plot files was from Titanic. We generated a word cloud of all the words that were mostly mentioned in that plot file. The output is presented in Figure 3. The ouputs of these two sub-systems, word cloud generators, and movie character identification were integrated into the Movie-Spin browser.

Figure 3: Titanic Movie Poster

Figure 4: Titanic Word Cloud

As we can see from the example the most common words in that movie plot was ”Jack”, ”Rose”, and Cal. After, developing the word cloud the goal was to use Name Entity Recognition to extract the category of Persons to get the

[1] https://github.com/amueller/word_cloud

Figure 5: Dark Knight Movie

Figure 6: Dark Knight Word Cloud

characters and the number of times that they were mentioned in that specific plot file belonging to the corresponding movie, in this case being *Titanic*.

We show the characters and the number of times that they were mentioned for Titanic in Table 1. Although there were a list of characters, we decided to only include the top two characters that were mentioned in the plot file in Movie-Spin. So in this case the characters taken into consideration were ”Jack”and ”Rose”, which are indeed the main characters in this movie. As you can see there is a pattern between the word cloud and the top two characters for Titanic. Both seem to have the same names. Being the case that ”Jack”and ”Rose” were the top two characters. Then we applied sentiment analysis to see if the characters were negative, neutral, positive or compound as an additional source of information. We wanted to test only whether a character was deemed as strongly negative or strongly positive. We used the Vader sentiment analysis model included in NTLK to automatically handle sentiment analysis. We show the average positive, negative and neutral scores for Jack and Rose (the main characters) in Table 2. The way to determine if a character is more positive or more negative is by checking the scores and seeing which one is closer to 1. In this case ”Rose” is more positive. Her positive number is higher than her negative which means she is positive. If we have watched the movie we know that she is a good character and not a villain. Vader is able to know this because of the words that are used in the sentences in the movie plot to mention ”Rose”. As for ”Jack” he is also positive because his positive score is closer to 1 than his negative score is. From these scores we can see that ”Rose”and ”Jack”are both the top two characters of the movie but also have a positive score, which means they are good characters. Another observation is that most sentences used to refer to any of these two characters are actually neutral, so looking at any single sentence it is unlikely to yield to good predictions about the character's role in the movie but using an aggregation of all sentences proves more promising.

Now that we know how we used these libraries and how we determined all of our results, I can show an example of a movie that has negative characters. In the movie ”Dark Knight Rises”, the top two characters found are ”Bane” and ”Wayne” using NER. From what we can see in Table 3 both characters are negative characters because their sentiment score is closer to 1. This is consistent with the movie, arguably the character of Batman (Wayne) in this movie is thought to be negative (hence the Dark Knight) in its own way at the end of the movie. A more nuanced of the characters might be possible but not necessarily desired using automated tools as one does not necessarily provide too much information in Movie-Spin to the point that the plot of the movie is revealed in any way.

Table 1: Extracted named entities from the movie Titanic along with their frequencies. Repeated character entries are due to a character being referred in two different ways e.g. `Rose` and `Rose Dawson`

Named Entity	Frequency
`Jack`	22
`Rose`	14
`Cal`	12
`Lovett`	2
`Rose DeWitt Bukater`	1
`Jack Dawson`	1
`Keldysh`	1
`Akademik Mstislav Keldysh`	1
`Rose Dawson`	1
`Ruth`	1
`Brock Lovett`	1

Table 2: Average sentiment scores for the extracted main characters in the movie Titanic using the Vader Sentiment Analysis model.

Named Entity	Sentiment	Score
`Rose`	positive	0.0896
`Rose`	neutral	0.8304
`Rose`	negative	0.0800
`Jack`	positive	0.1202
`Jack`	neutral	0.7974
`Jack`	negative	0.0816

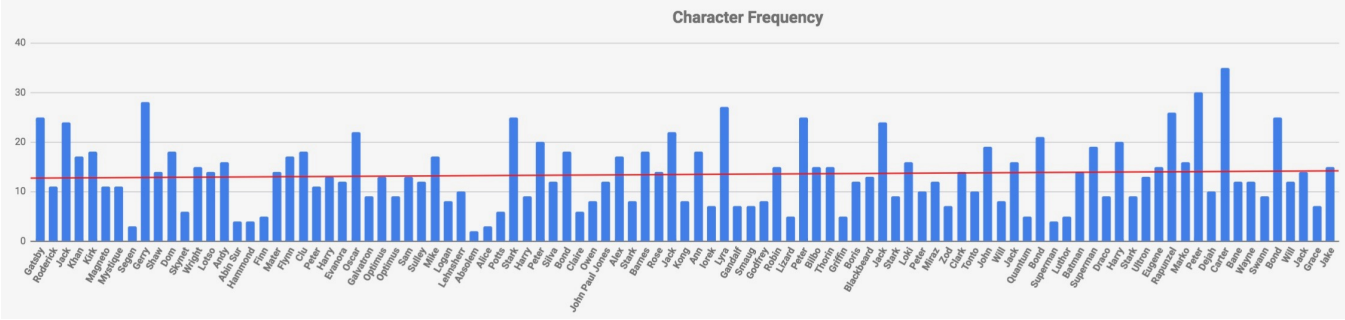

Figure 7: We show here the number of times characters are mentioned in their respective plots for each movie. Characters that are mentioned more times are likely to correspond to movies that have a strong main character or might be biographical. The red line shows the average number of times characters are mentioned. This can be used to divide the movies into movies with strong leading characters or not (e.g. Tony Stark in Iron Man, Mr Gatsby in The Great Gatsby, etc).

Table 3: Average sentiment scores for the extracted main characters in the movie The Dark Knight using the Vader Sentiment Analysis model.

Named Entity	Sentiment	Score
Bane	positive	0.0560
Bane	neutral	0.7798
Bane	negative	0.1641
(Bruce) Wayne	positive	0.0592
(Bruce) Wayne	neutral	0.8091
(Bruce) Wayne	negative	0.1316

Statistical Analysis

In the process of extracting all this data and information we were able to see some patterns. For example when looking at the top two characters for the first 100 movies, we can see that on average for each movie the top two characters are mentioned about 13 times. The maximum that a character is mentioned in this dataset is 37 times and the minimum is 1 time. This can characterize movies based on whether they contain a strong leading character or not. For instance, we can see in Figure 7 that movies with strong characters include Iron Man (Tony Stark), or The Great Gatsby (Mr. Gatsby), while other movies with a wide range of characters and worlds have less of a strong character such as Mystique in the X-Men series of movies which usually have a wide array of developed characters. The red line in Figure 7 marks the average of times characters tend to be mentioned in their movie plots. Any movie below that line can be considered in the second group of movies with a well developed cast of characters. This can be used in Movie-Spin to allow people to browse movies based on this characteristic.

Future Work

In the future we hope to be able to integrate the extraction of character and sentiment information from movie plots into a recommendation system. Based on data collected on characters and common words to determine what movie a person should watch based on the common words for each movie watched and how positive or negative are the top two characters for each movie. We also envision using more sophisticated natural language processing techniques such as semantic role labeling in order to identify more specific or fine-grained movie information such as *main character is a hero*, *main character is an anti-hero* in the extraction process. Further work is also needed to validate using human studies the efficacy of the browsing experience in Movie-Spin using a detailed breakdown of each type of automatically extracted structured information.

Acknowledgements

We would like to thank the UVA Dept. of Computer Science, in Partnership with DREU CRA-W, for hosting the internship NSF #CNS1246649, IAAMCS, and Access Computing for providing funding.

References

Bamman, D., OConnor, B., and Smith, N. A. (2013). Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 352–361.

Bird, S. and Loper, E. (2004). Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., and Riedl, J. (2003). Movielens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 263–266. ACM.

Wei, S., Zheng, X., Chen, D., and Chen, C. (2016). A hybrid approach for movie recommendation via tags and ratings. *Electronic Commerce Research and Applications*, 18:83–94.