Comedy Shows in Artificial Intelligence Evaluation

Anjaney Singh / 710071538

1. Abstract

The aim of the project is to use artificial intelligence elements to evaluate the relationships of comedy shows by using their scripts to create a network among the comedians. To achieve the task we are going to use several AI tools such as data scraping, word cloud, word frequency, cosine similarity, sentiment analysis, and network analysis. At the end of the project, we will have a recommender system that suggests people watch another related show if they enjoy a particular comedian, as well.

2. Introduction

Nowadays, AI implementations solve a lot of problems in different sectors. This project intends to evaluate the content of various comedy shows to understand their way to make people enjoy. Moreover, we will check whether there is a network among comedians which will give us a chance to suggest a particular comedy show to people. To clearly state the triggering question of this task is "why do I enjoy watching Frankie Boyle?" which directed me to compare other comedians to find patterns. In the introduction section, you will discover the details of the methodology of the project and the tools that are used to answer the question.

There are three main components of the project which are gathering and structuring data, natural language processing, and finally, network analysis among the comedians respectively. Each of these steps has substeps within itself which I will try to introduce briefly below. However here I would like to touch on each component to give a crystal-clear understanding.

The first step is collecting data which will be the scripts of a certain comedy show of each of the comedians. There are several web pages to use for this data collection process, so I will use one of them to get the raw form of the data. Cleaning and structuring this data for algorithms that I am going to use in the next stages will be another task in this component. This is a vital part of the project since if the data is not clean enough or not well structured, the result will be wrong and failure. Therefore, we do our best in this stage.

The following step will be NLP to understand the characteristics of each comedian via the words that they use. NLP will give us a chance to evaluate each show in a variety of aspects such as word counts, word clouds, sentiment analysis, speed of speech, etc. This stage gives us insight into data and takes us one step further to answer our questions. Here we will decompose the whole content to find the details or patterns in the haystack.

Finally, we will combine all of our findings to create a connection among comedians via the words that they used in their shows. Thus we will be able to group comedians who have strong connections so we can suggest people watch related shows if someone enjoys with any other within that group. This will be a recommender system as well.

3. Technical Aspect

a. *Data Pre-Processing*[i]

This part goes through a necessary step of any data science project - data preparation. Data preparation is a time-consuming and unenjoyable task, yet it's a very important one. Keep in mind, "garbage in, garbage out". Feeding dirty data into a model will give us results that are meaningless. In this process, specifically, we'll be walking through:

Getting the data - in this case, we'll be scraping data from a website

Cleaning the data - we will walk through popular text preprocessing techniques

Organizing the data - we will organize the cleaned data in a way that is easy to input into other algorithms

The output of this part of the notebook will be clean, organized data in two standard text formats:

- **Corpus** - a collection of text
- **Document-Term Matrix** - word counts in matrix format[ii]

I am going to use the Scrapes From the Loft website to scrape transcripts of selected shows of the comedians. Web scraping is a handy tool that data scientists use to get data in various formats over various websites. There are several Python libraries to get data from the web, however, in this project, I am going to use the Beautiful Soup and the Request libraries in combination to get data from the website that I mentioned above.

To decide which comedians to look into, I went on IMDB and looked specifically at comedy specials that were released in the past 10 years. To narrow it down further, I looked only at those with greater than a 7.5/10 rating and more than 2000 votes. If a comedian had multiple specials that fit those requirements, I would pick the most highly rated one. I ended up with ten comedy specials.

Cleaning text data is a little different from numerical data. There are many things to go on when handling the cleansing process of data text data however here I will follow a minimum viable approach to keep it simple and convenient for the sake of optimization. Here is the process I will follow up on common data cleaning steps on all text such as lower case characters, removing punctuation, removing numerical values, removing common non-sensical text (/n), tokenizing text, and removing stop words. This process can go further however for the sake of simplicity I will stop here. If I am not satisfied with the results I will come back to make some more adjustments in this step.

The result of this data pre-processing step should be a ready format for algorithms so we will finalize this step in the following formats:

Corpus - a collection of text which is the raw transcript of each comedy show in this case.

Document-Term Matrix (DTM) - word counts in a matrix format in which each row of matrices are the frequency of words in the corresponding column.

For many of the techniques we'll be using in future tasks, the text must be tokenized, meaning broken down into smaller pieces. The best choice of tokenization technique for our purpose is to break down the text into words. We can do this using sci-kit-learn's CountVectorizer, where every row will represent a different document and every column will represent a different word. Additionally, with CountVectorizer, we can remove stop words. Stop words are common words that add no additional meaning to text such as 'a', 'the', etc.

b. *Natural Language Processing*[iii]

NLP includes a wide range of evaluation techniques to learn from text data. In this stage, we will use some of these approaches from simplest implementations to advanced data algorithms to find some insights. The findings of this stage will be discussed in the outcomes section of the report as a whole.

After getting and restructuring data, now, it is time to discover some basic patterns among the words that the comedians use in their shows. This task is rather simple however vital to understand data before applying any fancy algorithms. This exploratory data analysis (EDA) part consists of;

Most common words - find these and create word clouds. In this part, I will try to find the most common words and unnecessary words that are used by comedians and visualize them to have an idea about their shows. I am not going to use advanced algorithms in this step but implement simple data evaluation techniques. The output of this section is the word clouds of each comedy show.

Size of vocabulary - look number of unique words and also how quickly someone speaks. I aim to find out more about the vocabulary that comedians use and their speed of speech which will be a piece of valuable information about the characteristics of the shows. I find the duration of each show in IMDB and stored them as a list. To find the speaking speed of each comedian, I will divide the total number of words over the time of the show so the result will be words per minute which is speech speed.

Amount of profanity - most common terms that comedians use. Here I will subset some bad words that I find out in word clouds and word counting of each program. This will help me to identify the characteristics of each show which will be a vital parameter to classify the comedians. The result will be a scatter plot that indicates the position of comedians' frequency of use of S-words vs F-words.

Sentiment analysis – to find out more about the vibe of the routine. We have gone through pretty generic analysis techniques such as counting, creating scatter plots, etc which could be used with numeric data as well. However, now I will apply more text-centric techniques such as sentiment analysis, topic modeling, etc. Before I start I would like to give brief information about the methods that I am going to use.[iv]

1. **TextBlob Module**: Linguistic researchers have labeled the sentiment of words based on their domain expertise. The sentiment of words can vary based on where it is in a sentence. The TextBlob module allows us to take advantage of these labels.
2. **Sentiment Labels**: Each word in a corpus is labeled in terms of polarity and subjectivity (there are more labels as well, but we're going to ignore them for now). A corpus' sentiment is the average of these.
   - Polarity: How positive or negative a word is. -1 is very negative. +1 is very positive.
   - Subjectivity: How subjective, or opinionated a word is. 0 is the fact. +1 is very much an opinion.

Topic modeling – to find the topics within the corpus. Topic modelling is another technique that is used in NLP. The focus of topic modelling is to find various topics that are within the content of the corpus. A document could be based on one single topic as well as multiple topics. Thus, we will have an idea of what comedians focus on in their shows. In this section, I will use Latent **Dirichlet Allocation (LDA)**, which is one of many topics that we studied in our course work, as modelling techniques. LDA is a method designed to evaluate text data, primarily.

In the technical aspect, we need to provide a DTM and a number of topics to the algorithm as an input. The optimum selection of a number of topic is important to receive reliable results. Once we got the words for the selected number of topics, we need to interpret what topic those words would belong to. If they are not meaningful we need to try again by changing the number of topics.

c. *Network Analysis*[v]

In this last section, we are going to look at the relationship between words and comedians via network analysis. There are lots of useful evaluation approaches in network analysis so I will go over and use some of them to figure out how words connect comedians to each other. To achieve this task, I am going to;

- draw a **network graph** to visualise if there is any hidden pattern
- check the **centrality** tendency of the network to find connection points
- calculate **cosine distance** to find out the similarity of comedians in the network

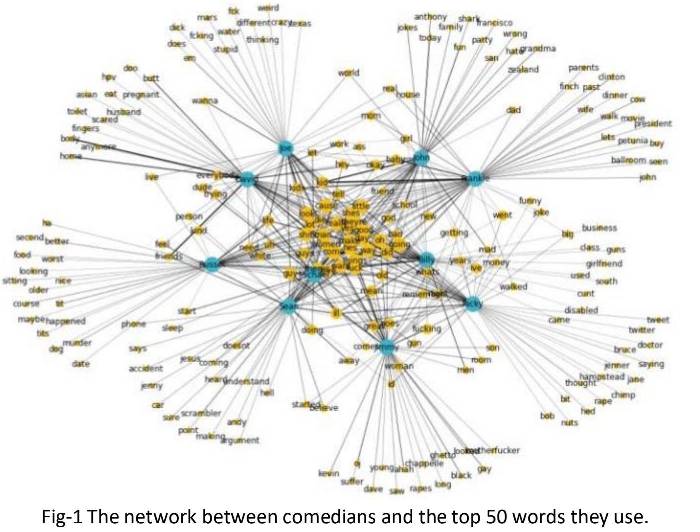

Fig-1 The network between comedians and the top 50 words they use.

The figure above is the connection map between comedians by words. To draw this network, I used the top fifty most frequently words by each comedian. Yellow nodes represent words and blue ones are for comedians. The size of each word node is proportional to its degree which is the number of total connections. So bigger yellow node means a more connected word. In addition, the thickness of the edge between nodes is thicker if the word is used more frequently by the comedian.

The centrality of a network can be measured for different purposes. In this part, I aim to figure out which nodes are most used by calculating degree centrality, what is the closest path between nodes by the calculation of closeness centrality and, finally, which nodes are the bridge between different parts of the network by using betweenness centrality. To do so, I used the Networkx's pre-built functions and stored the results in a data frame to evaluate them.

I purposely kept cosine distance calculation to the end since it is the step I would generalize results. Cosine distance can be thought of as the angular distance between two vectors so it will give a clear perspective to relate the vectors. As we know each comedian has a column vector that comes from the words that they used in their shows, in our DTM. So the calculation of cosine distance between these vectors by iterating through each comedian will give us a matrix that contains values of cosine distances between each comedian. So we will be able to interpret how comedians are related to each other and be able to group them.

4. Outcomes

Up until here, we talked step by step about the technical process and methods of the project. In this part, we will dive into to findings and interpret the outcome of each step. I would like to point out that the project was built based on a data science life cycle process that starts with a question, collects data to solve it, cleanse the data and implement artificial intelligence to get results. Let's go through the natural language processing and network analysis steps of the project to see what we have already found.

Before AI implementation, I cleaned data by removing noise such as punctuation, numbers, newline characters, etc. Then, finally restructured data as a corpus and DTM to use for different purposes.

The first part of the NLP, wording, was a kind of deep dive into data to explore and understand it in detail. In this part, I found out that most comedians use profanity frequently in their shows. That's why I decided to evaluate the S and F words at the end of this section. Additionally, I noticed that Frankie talks about facts of life, kids etc. And he uses fewer S or F words that might attract my interest in his shows. You can also look at the word cloud in this section to interpret the characteristics of your favorite comedian.

The first part of the NLP, wording, was a kind of deep dive into data to explore and understand it in detail. In this part, I found out that most comedians use profanity frequently in their shows. That's why I decided to evaluate the S and F words at the end of this section. Additionally, I noticed that Frankie talks about facts of life, kids etc. And he uses fewer S or F words that might attract my interest in his shows. You can also look at the word cloud in this section to interpret the characteristics of your favourite comedian. Other evaluations in this section are the number of unique words, the total number of words used by a comedian and the speed of their speech. In terms of unique words, Michael and Ricky use diverse vocabulary in their shows. On the other hand, Ricky speaks farter than every other comedian. However, Russel speaks slower and uses fewer words which indicates, that perhaps, he uses mimics in his shows. These outcomes might be an important parameter to decide why a comedy show attracts your attention. Finally, the scatter plot, which shows how frequently comedians use profanity, groups comedians into three main classes. Interestingly, Frankie never used a bad word in his program and John and Ricky use relatively fewer F-words and S-word so this might be their common point. Since I don't like too much swearing, especially the f-word, which is probably why I've never heard of Michael, Jimmy and Billy. Therefore, It looks like the profanity parameter might be a good predictor of the type of comedy I like. Besides Frankie, I might like John and Ricky's comedy shows among others.

In sentiment analysis, I got two distinct graphs. The scatter plot indicates the Fact-Opinion level on the y-axis versus the Negative-Positive characteristics of the comedy routine on the x-axis. Frankie and David Mitchell relatively talk about the facts whereas John Cleese mostly talks about the opinion which is a significant distinction in the style of the comedian. Jimmy Carr and Michael are quite negative. On the other hand, Sean Lock, Jim Rogan and Frankie keep their speech relatively positive. Russel Hovard keeps the balance in all situations. He is in the middle of everything. Finally, I divided each speech into ten equal parts and evaluated each part to see how programs go through time. David, Sean, Frankie and Russel stay positive during their shows, however, John, Jimmy and Michael are quite negative in their shows. Jimmy goes very positive towards the end of his show.

The topic modelling and centrality didn't provide too much information so I just refer you to read the findings in the notebook so for the sake of saving space, I skip them in my report.

The network between words and comedians provides additional evidence to support our claims. In fig-1, you can see that The words in the outer ring of the graph are the unique words for the associated comedians. The words in the middle of the graph are the most frequently used words by most comedians. As you can see, Frankie and John talk about kids frequently and David uses body, and home words too much. The outputs of the graph support our previous findings.

In conclusion, the cosine similarity among comedians gives me a chance to group comedians as follows**.** **Group 1**: Russel, Michael, Ricky, Billy, David, Joe, **Group 2**: Jimmy, Michael, Billy, David **Group 3**: Ricky, Billy, Michael, John **Group 4**: Frankie, Sean.

Finally, I can use this model as a recommender system as well to recommend people a comedy show.

[i] García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing.

[ii] Anandarajan, Murugan, Chelsey Hill, and Thomas Nolan. "Term-document representation." *Practical text analytics*. Springer, Cham, 2019. 61-73.

[iii] Chowdhary, KR1442. "Natural language processing." *Fundamentals of artificial intelligence* (2020): 603-649.

[iv] Feldman, Ronen. "Techniques and applications for sentiment analysis." *Communications of the ACM* 56.4 (2013): 82-89.

[v] Brandes, Ulrik. *Network analysis: methodological foundations*. Vol. 3418. Springer Science & Business Media, 2005.