Text Mining Project Report

"Hospital Playlist Sentiment Analysis"

Stefano de Saraca

1. Introduction

The "Hospital Playlist Sentiment Analysis" project aims to research, understand and store the sentiment of the lines acted by each character in the first three episodes of the South Korean TV series "Hospital Playlist".

The ultimate purpose of the analysis is to identify the five main characters for each episode, to understand how the sentiment of each one changes throughout the episodes and whether this can be found in the actual footage.

The English subtitles (the language chosen for the analysis) were obtained from www.opensubtitles.org at the URL:

https://www.opensubtitles.org/en/ssearch/sublanguageid-all/idmovie-920439

To transform the files from SRT format (basic subtitles) to CSV, the converter made available free of charge by www.gotranscript.com at the URL was used:

https://gotranscript.com/subtitle-converter

The chosen programming language for the analysis is Python and multiple libraries have been used of which the most important are wordcloud, re (Regular Expressions), NLTK and Transformers.

The project is structured in phases:

- Data Cleaning (described in the data card)
- Word Frequency Analysis and WordCloud Generation
- Sentiment Analysis witch VADER and roBERTa
- Descriptive Analysis and Plots
- Data Export (includes everything, both analyzed and result scores)

Data Cleaning

Reading the data and cleaning it using various methods. The main cleaning steps are:

- Filling NaNs
- Data type casting where needed
- Assigning the character's name for each sentence
- Removing emojis and converting every character to its closest ASCII one
- Removing dashes (-) in the sentences
- Removing stopwords

Word Frequency Analysis and WordCloud Generation

Using the re (RegEx) and wordcloud libraries we'll find the top 10 most frequent terms for every episode and generate a WordCloud of the top 50 (most frequent).

Stopwords have been removed before the word frequency analysis.

Sentiment Analysis with VADER and roBERTa

Is this section of the project we'll apply two different techniques for sentiment analysis and gather the result in a final dataframe which we'll use later for further investigations and analyses.

The first method is based on the "Bag of Words" method, while the second one uses a Neural Network model developed by Google and improved by CardiffNLP training it on around 58 million tweets and finetuning it for sentiment analysis.

Descriptive Analysis and Plots

A full descriptive analysis conducted on the dataframe containing both text and related data (like speaker, sentence number, etc.) and sentiment data.

Multiple plots are generated during this phase.

Both the numbers and plots obtained from the analysis are saved respectively in a custom folder, one for the data and one for the plots.

The results are written both in CSV and JSON.

NOTE: only the results of the analysis and the plots are exported during this phase, NOT the dataframe (the complete one) which was used for the analysis.

Data Export

During this phase the dataframe which contains both text, related data and sentiment one is exported in both CSV and JSON.

3. Algorithms and Approaches

Three analyses have been carried out on the project:

- Word Frequency Analysis and WordCloud Generation
- Sentiment Analysis
- Descriptive Analysis

For each of them specific algorithms and approaches have been used respectively to calculate statistics and insights on the data itself and to minimize the computational power required (applying efficient algorithms and data structures, simple code syntax, etc.).

3.1.1 Word Frequency Analysis

This specific analysis is carried out in steps:

1. Loading the text and joining all sentences
2. Loading stopwords into a list
3. Finding patters
4. Creating a dataframe with words and frequencies as columns
5. Returning the data
6. Showing the results

1. Loading the text and joining all sentences

Loading the text is quite easy and requires just to create a subset with only the "Text" column of the dataset (which contains all the sentences from the episode).

2. Loading stopwords into a list

Loading the stopwords just requires to read a text file which contains the words themselves.

Using a list comprehension to generate the list of stopwords the code will run faster.

3. Finding patterns

Using the re library we'll first define the pattern we're looking for and then find and returning it.

4. Creating a dataframe with words and frequencies as columns

We'll create a dictionary which has the words frequency as the keys and the word themselves as the values.

Using a specific method for the Pandas library we're able to create a dataframe from the dictionary generated before.

Furthermore, the values are sorted in descending order through a method from the same library as before.

5. Returning the data

We'll just return the dataframe with the "return" statement from the function.

6. Showing the data

The data is shown just after the data cleaning to have an overview on the most frequent terms which will be combined with other insights later on.

3.1.2 WordCloud Generation

WordClouds are often used to show a cool overview of the most recurrent words by combining a specific number of them (the more recurrent ones) into an image in which the bigger the word is written, the more frequent it is.

To accomplish this task the wordcloud library has been used.

Using two specific methods from the library itself the WordCloud gets generated from the dataframe which was passed as an input.

In particular, only the 50 most frequent words have been plotted.

The WordCloud is then exported as a .png image in a custom folder.

3.2 Sentiment Analysis

The sentiment analysis phase is structured in three main parts:

1. VADER approach
2. roBERTa neural network approach
3. Plots generation and export
4. Sentiment statistics data export

Each method is applied directly to each sentence individually to better understand the sentiment of the character in the specific instant in which he or she's talking.

3.2.1 VADER Approach

VADER, which stands for Valence Aware Dictionary for Sentiment Reasoning, is a tool of the NLTK library for the Python programming language which **uses the "Bag of Words" approach** for sentiment analysis.

This means that the algorithm won't take into consideration the context in which the words are located, thus potentially ignoring specific meanings that are related to the context rather than to the words only.

Original VADER documentation available on the NLTK official website at: https://www.nltk.org/api/nltk.sentiment.vader.html

Using this kind of approach means less complexity for the analysis, thus reducing the computational power and time required.

The analysis takes place running the algorithm on the "Text" field of each row (which contains the sentence acted by the characters).

The algorithm returns four different scores: neg, pos, neu, compound, each of them go from 0 to 1 and the sum of neg, neu and pos is 1.

Each of them represents a "quote" of the sentiment of the text.

Example: a text can have a positive sentiment (pos) of 0.73, a negative one of 0.05 and a neutral one of 0.22. This means that the text will probably be very positive, slightly neutral and almost non negative.

At the end of the analysis, we'll have a full dictionary that includes sentence indexes as keys and dictionaries that include the sentiment scores as values.

After collecting the results we'll create a dataframe using the Pandas library which gathers the sentence index and all the scores organized in columns.

This dataframe will then be merged with the original one containing other data and the one with the sentiment scores collected from the other sentiment analysis using the roBERTa neural network.

3.2.2 roBERTa (Neural Network) Approach

roBERTa is a neural network model based on the BERT neural network developed by Google. It has been improved by training it on around 58 million tweets, finetuned for sentiment analysis and maintained by CardiffNLP and published on HuggingFace at:

https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

The BERT neural network: https://it.wikipedia.org/wiki/BERT

CardiffNLP GitHub: https://cardiffnlp.github.io/

This model uses a "String of Words" approach, which means the context in which the words are located matters to the algorithm and so it has an influence on the sentiment scores of the text.

In fact, with this approach the meaning of the words isn't determined only by themselves, but from the context too.

roBERTa has been implemented in the code using the Transformers library for the Python programming language.

The code structure for the analysis is made of two functions:

Parent function

Here both the tokenizer and the model are instanced, the sentiment scores are gathered and organized in a dataframe.

Nested function

Here the text gets encoded and the model analyzes it returning a PyTorch tensor object from which the scores are collected by transforming it into a Numpy array and applying the SoftMax function.

To obtain the sentiment scores for the "Text" field of each row the nested function is applied with a for cycle to every one.

The roBERTa model returns three parameters: neg, neu and pos. These have the same characteristics as the ones returned from the VADER algorithm.

The scores example is the same too.

3.2.3 DataFrames Merge

Finally, all the dataframes created since the start of the program are merged into one complete dataframe that includes both the original data which was used for the analysis and the sentiment scores obtained from the VADER algorithm and the roBERTa neural network.

Before continuing with the analysis, there's an important step which is necessary to understand the true sentiment of the sentence.

Two more columns are created, these will define the major sentiment.

Since during the analysis we'll compare the results from both VADER and roBERTa. Knowing the approaches which are used by each one, we'll take for granted that the roBERTa model will give a more precise evaluation of the sentence's sentiment and for this reason we'll use its scores to elaborate plots and other potential statistics based on the sentence's major sentiment.

The two new columns will have either "Positive", "Neutral" or "Negative" as their possible values.

Finally, from this new and complete dataframe a full descriptive analysis will be carried out to understand the results.

3.3 Descriptive Analysis

In this last analytic phase, we'll explore the data and process some statistics on it.

For everything related to numbers, Pandas, Numpy and standard Python functions have been used while on the other side Matplotlib and Plotly have been used for the plots creation.

3.3.1 Basic Statistics

First thing first, we'll check the columns and some basic statistics of them as already showed in the data card.

This is made very easy by the Pandas describe() method which saves a lot of time that would have been otherwise spent by processing everything by ourselves.

3.3.2.1 roBERTa Sentiment Percentages

We'll then calculate the percentages of sentiment based on the scores obtained from the roBERTa model for the episode taken into consideration, meaning we'll divide the number of positive, neutral and negative sentences by the total number of sentences.

$$roBERTa\ Positive\ Sentences\ (\%) = \frac{Total\ roBERTa\ Positive\ Sentences}{Total\ Sentences}$$

$$roBERTa\ Neutral\ Sentences\ (\%) = \frac{Total\ roBERTa\ Neutral\ Sentences}{Total\ Sentences}$$

$$roBERTa\ Negative\ Sentences\ (\%) = \frac{Total\ roBERTa\ Negative\ Sentences}{Total\ Sentences}$$

3.3.2.2 VADER Sentiment Percentages

Here the same principle from above is applied, but using the major sentiment calculated from the VADER algorithm scores, which for the nature of its approach will be less precise.

$$VADER\ Positive\ Sentences\ (\%) = \frac{Total\ VADER\ Positive\ Sentences}{Total\ Sentences}$$

$$VADER\ Neutral\ Sentences\ (\%) = \frac{Total\ VADER\ Neutral\ Sentences}{Total\ Sentences}$$

$$VADER\ Negative\ Sentences\ (\%) = \frac{Total\ VADER\ Negative\ Sentences}{Total\ Sentences}$$

3.3.3 Sentences Count by Character

One important key factor for the analysis is determine the number of sentences by character, in fact since not every character acts the same number of phrases the results of the analysis can be influenced by this.

3.3.4 Various Functions on Sentences with a Specific Sentiment by Character

This phase is dedicated to understanding some aspects of the data we obtained from the two algorithms applied for the sentiment analysis.

Six groupby operations get executed, respectively to gather positive, neutral and negative sentences by character for both VADER and roBERTa.

Four functions will be then applied to the groupbies:

- Count
- Mean
- Median
- Standard Deviation

Doing this we'll better understand how the sentiment changes between characters and major sentiment of the sentences acted.

In fact, the goal of this section of the analysis is to gather all these statistics in one dataframe which can be then saved and examined afterwards.

From these data we can determine the variation of the sentiment trends, so creating a history between the episodes.

Example: having median, standard deviation and count of sentences with a specific major sentiment (calculated either from VADER or roBERTa scores) for a character, we can see how the trends change throughout the episodes.

For instance, it could be: "Character XYZ has an average positive sentiment for positive sentences of 0.73 (out of 1) with a standard deviation of 0.18 (out of 1) in the first episode. In the second one the average drops of 0.10 and the std drops of 0.08.

Logically we'll see an increment on either neutral or negative scores or both at the same time.

We can deduct then that the character after some specific events changed his behaviour, having a less positive attitude and communication.

Finally, with a short dictionary containing useful information about the episode itself the data gets saved in JSON format in a custom folder.

3.3.5 Plots

This section of the analysis is completely dedicated to plots graphically representing the data and the statistics calculated before.

This is especially useful to both see and compare the episodes between each other and see the differences.

All the images will be shown in the "Results" paragraph.

Six plots have been created using different libraries:

- roBERTa Major Sentiment Plot – Pie Chart (Matplotlib)
- Sentiments Median by Character – Grouped Bar Chart (Matplotlib)
- Sentiments Standard Deviation by Character – Grouped Bar Chart (Matplotlib)
- Median Positive Sentiment by Character – Bar Chart (Plotly)
- Median Neutral Sentiment by Character – Bar Chart (Plotly)
- Median Negative Sentiment by Character – Bar Chart (Plotly)

roBERTa Major Sentiment Plot

Represents in a bar chart the percentages calculated before and described in the 3.3.2 paragraph.

Sentiments Median by Character

Shows the median positive, neutral and negative sentiment (calculated respectively only on positive, neutral and negative sentences) on a grouped bar chart.

Sentiments Standard Deviation by Character

Based on the same approach of the "Sentiments Median" one, but using standard deviation instead.

This one is a grouped bar chart too.

Median Positive, Neutral and Negative Sentiment by Character

These three charts represent different data in the same way.

They're three bar charts which are created with Plotly and they show the median positive, neutral and negative sentiment score (calculated respectively only on positive, neutral and negative sentences) for each character.

4. Results

In this section of the report, we'll briefly describe the results of the analyses, structuring the most important ones and arriving to a conclusion.

There will be also a great focus on the motivations of some results which can be very interesting to compare to the footage of the TV series.

The main objects of this paragraph will be:

- Numerical and categorical results of the analyses
- Plots
- Examples

Since many results of the algorithms which have been used in the analysis are already reported on the data card we'll go over only the most important ones in this paragraph.

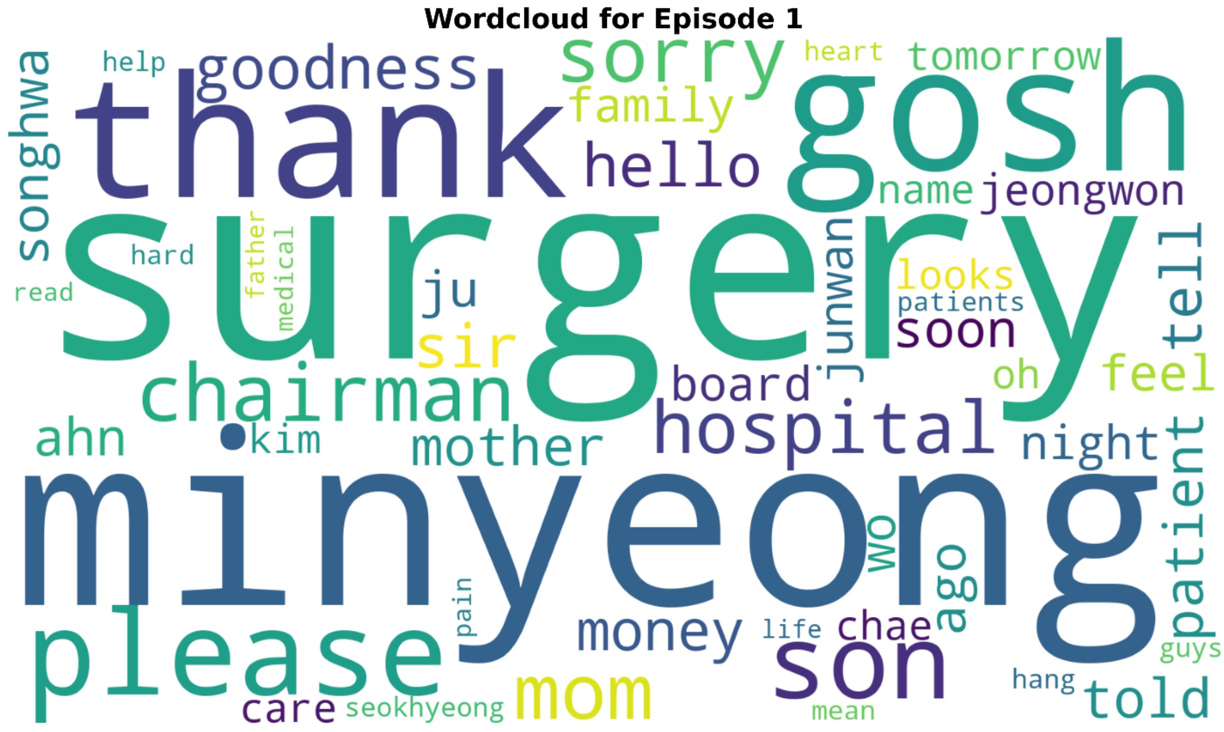
4.1 Word Frequency Analysis

TOP 10 Most Frequent Words for Each Episode

Episode 1	Words Frequency	Episode 2	Words Frequency	Episode 3	Words Frequency
surgery	39	surgery	70	surgery	56
minyeong	32	patient	39	gosh	28
gosh	27	gosh	34	sorry	23
thank	25	mom	31	hello	22
please	24	sorry	29	thank	22
son	18	thank	27	heart	21
chairman	18	please	24	chanhyeong	20
sorry	17	jang	23	day	19
mom	17	tell	23	patient	19
hospital	15	told	21	guys	18

4.2 WordCloud Generation

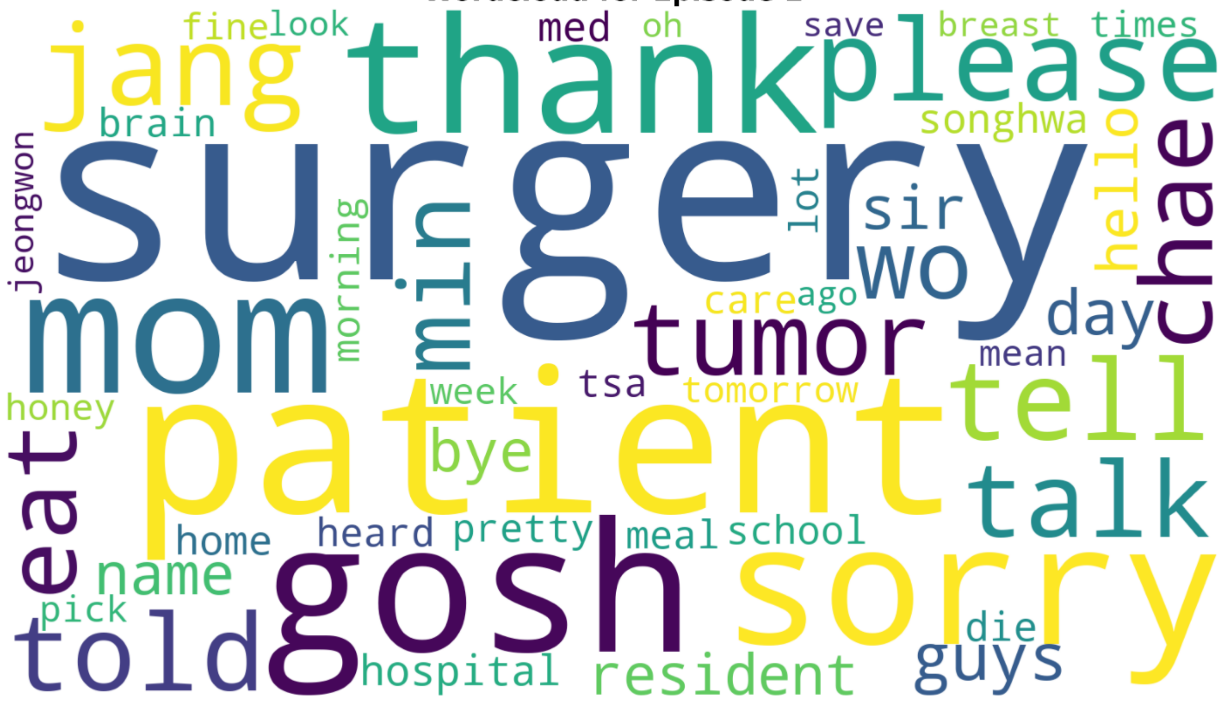
WordClouds generated for each episode:



Description: there are multiple details which we can trace back to the footage, here are some of them:

- **Minyeong**: the little girl who unfortunately died after complications, she was very young and her mother suffered very much from her loss. The doctors did their best to save her.
- **Surgery**: as per usual doctors have to operate on their patients and so the main characters (doctors) have to.
- **Chariman**: the owner of the hospital in which the main characters work. During the first episode, after his death, it turns out that he was the father of Ahn Jeong-won. His friends get a bit upset not knowing he was his son.
- **Money**: a very debated subject which the characters talk about after the chairman death and the inheritance of his assets by his family (so Ahn Jeong-won too).

Wordcloud for Episode 2



Description: other multiple interesting words relatable to the footage. Some of them are:

- **TSA**: one of the procedures which the doctors execute during the episode.
- **Meal**: Lee Ik-joon disguises himself as a canteen operator and gives more food to new interns of the hospital.
- **Honey**: when cancer gets diagnosed to a patient, she and her husband cry a lot, but choose to treat it.
- **Med**: as the new interns explore the hospital from the doctor's point of view, they tell their story and why they have chosen med school.

Wordcloud for Episode 3

Description: as in the other two examples we can see some great references to the episode's footage. Here are some of them:

- **Dad**: as time passes Lee Ik-joon's son grows and his father, after his wife asked for divorce, doesn't want to let down his son.
- **Gyeoul**: one doctor (Gyeoul) likes Ahn Jeong-won a lot and starts to collect information about him through her friendships and colleagues, but she's still very shy with him.
- **Surgery**: at the start of the episode Kim Jun-wan finds out a dangerous mass which threatens a man's life. When telling his diagnosis to the man and his daughter he's very unemotional and detached from his patient's feelings and this causes afterwards a debate between him and another doctor which was present at that moment too.
- **Heart**: when attending as student-spectators to a heart operation two med-school students (interns too) get the chance to touch the beating heart for the first time and feel the true power of the beat.

4.3 Sentiment Analysis and Descriptive Analysis

In this section we're going to be showing the results obtained from the sentiment analysis executed on each episode's data and explaining the differences between the two algorithms' results.

We'll be focusing more on aggregated statistics and compare the episodes to see if there are any changes, and if so which ones.

NOTE: to clarify as much as possible the data which was first gathered from the algorithms and then analyzed during the descriptive analysis, this section will comprehend the results of both parts.

But first of all, let's give some context to the data we're going to see.

4.3.1 Total Sentences by Major Sentiment and Algorithm

Episode 1 – Major Sentiment		
	VADER	**roBERTa**
Total Positive Sentences	227	121
Total Neutral Sentences	892	929
Total Negative Sentences	99	168

Episode 2 – Major Sentiment		
	VADER	**roBERTa**
Total Positive Sentences	313	143
Total Neutral Sentences	1049	1164
Total Negative Sentences	109	164

Episode 3 – Major Sentiment		
	VADER	**roBERTa**
Total Positive Sentences	325	170
Total Neutral Sentences	1038	1165
Total Negative Sentences	101	129

As we can see in all of the three episodes there's a difference between the scores which are collected from the two algorithms, that sometimes (because of the different scores) give a different label to the sentence.

Example: the same sentence could have been assigned different sentiment scores between VADER and roBERTa, thus labeling the text as "Negative" for the first algorithm and "Neutral" for the second one.

4.3.2 Total Sentences by Character

	Character					
	Ahn Jeong-won	**Kim Jun-wan**	**Lee Ik-joon**	**Yang Seok-hyung**	**Chae Song-hwa**	**Secondary**
Episode 1	204	117	23	51	166	654
Episode 2	90	90	133	34	228	986
Episode 3	65	240	239	15	95	810

As we can clearly see setting apart the secondary characters, which summed always act more lines, there are some key characters for each episode.

Example 1 (referred to the table above): Ahn Jeong-won has an important role in the first episode because of his parental relationship with the chairman of the hospital, thus he talks more because there are more scenes in which he's an active speaker.

Example 2: in the third episode Lee Ik-joon plays a very important role too, because after meeting his wife after some time she asks for a divorce and there are many scenes in which he either talks with his wife or with his son, therefor he acts more lines.

Example 3: Yang Seok-hyung seems to be in all of the three episodes the least social person of the group of friends and this is confirmed when in the fourth episode he actively tries to avoid some attentions from other girls to take care of his mother instead.

<u>NOTE: the fourth episode is not taken in consideration in this analysis, but still is a confirmation of the data collected in the first three ones.</u>

4.3.3 Sentences Percentages

	roBERTa Sentiments		
	Positive	**Neutral**	**Negative**
Episode 1	0.099	0.762	0.137
Episode 2	0.097	0.791	0.111
Episode 3	0.088	0.795	0.116

	VADER Sentiments		
	Positive	**Neutral**	**Negative**
Episode 1	0.186	0.732	0.081
Episode 2	0.212	0.713	0.074
Episode 3	0.221	0.709	0.068

As we can see it's clear that the majority of the lines acted are neutral, that's because most of the sentences are either very short and not expressive, like "Okay" or "Thank you", or because many of them are just neutral.

On the other side we can see how the percentage of positive sentences varies a lot based on the algorithm.

This is obvious looking at the difference between VADER percentages and roBERTa ones.

The Hospital Playlist TV series is dramatic and many of the sentences in the real footage represent sadness, sometimes happiness too or just neutrality, but sadness is more present (of course).

Thus, we can deduce that not considering the context of the words seems to really make the difference as VADER returns more positive results, while roBERTa seems more balanced, but still slightly more oriented to the negative side.

What's curious too is that VADER seems to detect an increase of positive sentiment when going forward with the episodes, while the opposite can be said about roBERTa, which only decreases very little.

The total positivity sentiment percentage difference between the first and the third episode is:

- roBERTa: 0.011 (11.1% decrease in positivity)
- VADER: 0.035 (15.8% increase in positivity)

4.3.4 Various Statistics

To simplify this section of the report we'll report only the median values obtained from groupbies and standard deviations being the two most important information which isn't overlookable.

-------------- EPISODE 1 ---------------

Episode 1 – Median Sentiment by Character (roBERTa)			
	Median Positive Sentiment (For Positive Sentences)	**Median Neutral Sentiment (For Neutral Sentences)**	**Median Negative Sentiment (For Negative Sentences)**
Ahn Jeong-won	0.846	0.716	0.682
Kim Jun-wan	0.697	0.715	0.605

Lee Ik-joon	0.616	0.681	0.843
Yang Seok-hyung	0.657	0.700	0.694
Chae Song-hwa	0.715	0.712	0.585
Secondary	0.719	0.705	0.650

Episode 1 – Sentiment Standard Deviation by Character (roBERTa)			
	Positive Sentiment STD (For Positive Sentences)	**Neutral Sentiment STD (For Neutral Sentences)**	**Negative Sentiment STD (For Negative Sentences)**
Ahn Jeong-won	0.158	0.108	0.115
Kim Jun-wan	0.106	0.107	0.119
Lee Ik-joon	0.129	0.097	0.129
Yang Seok-hyung	0.171	0.107	0.148
Chae Song-hwa	0.153	0.102	0.136
Secondary	0.131	0.112	0.123

Episode 1 – Median Sentiment by Character (VADER)			
	Median Positive Sentiment (For Positive Sentences)	**Median Neutral Sentiment (For Neutral Sentences)**	**Median Negative Sentiment (For Negative Sentences)**
Ahn Jeong-won	0.666	1.0	1.000
Kim Jun-wan	0.615	1.0	0.542
Lee Ik-joon	0.615	1.0	0.633
Yang Seok-hyung	0.615	1.0	0.610
Chae Song-hwa	0.691	1.0	0.706
Secondary	0.669	1.0	0.629

Episode 1 – Sentiment Standard Deviation by Character (VADER)			
	Positive Sentiment STD (For Positive Sentences)	**Neutral Sentiment STD (For Neutral Sentences)**	**Negative Sentiment STD (For Negative Sentences)**
Ahn Jeong-won	0.268	0.179	0.201
Kim Jun-wan	0.262	0.182	0.055
Lee Ik-joon	0.296	0.142	0.113

Yang Seok-hyung	0.308	0.160	0.161
Chae Song-hwa	0.221	0.190	0.150
Secondary	0.265	0.172	0.176

<u>Episode 1 Results Comment:</u> while all characters seem to have a very similar median sentiment for each polarity (positive, neutral, negative), there's remarkable evidence in the positive medians that VADER results are higher than the roBERTa ones.

On the other side neutral and negative sentiment medians tend to be very similar between roBERTa and VADER although sometimes one is a bit higher than the other.

-------------- <u>EPISODE 2</u> ---------------

Episode 2 – Median Sentiment by Character (roBERTa)			
	Median Positive Sentiment (For Positive Sentences)	**Median Neutral Sentiment (For Neutral Sentences)**	**Median Negative Sentiment (For Negative Sentences)**
Ahn Jeong-won	0.635	0.744	0.622
Kim Jun-wan	0.695	0.726	0.663
Lee Ik-joon	0.781	0.697	0.552
Yang Seok-hyung	0.629	0.689	0.610
Chae Song-hwa	0.700	0.692	0.700
Secondary	0.776	0.707	0.647

Episode 2 – Sentiment Standard Deviation by Character (roBERTa)			
	Positive Sentiment STD (For Positive Sentences)	**Neutral Sentiment STD (For Neutral Sentences)**	**Negative Sentiment STD (For Negative Sentences)**
Ahn Jeong-won	0.132	0.094	0.118
Kim Jun-wan	0.228	0.096	0.135
Lee Ik-joon	0.155	0.103	0.130
Yang Seok-hyung	0.147	0.106	0.165
Chae Song-hwa	0.142	0.111	0.100
Secondary	0.135	0.112	0.122

Episode 2 – Median Sentiment by Character (VADER)			
	Median Positive Sentiment (For Positive Sentences)	**Median Neutral Sentiment (For Neutral Sentences)**	**Median Negative Sentiment (For Negative Sentences)**
Ahn Jeong-won	0.572	1.0	0.654
Kim Jun-wan	0.590	1.0	0.608
Lee Ik-joon	0.655	1.0	0.655
Yang Seok-hyung	0.561	1.0	0.597
Chae Song-hwa	0.677	1.0	0.730
Secondary	0.672	1.0	0.649

Episode 2 – Sentiment Standard Deviation by Character (VADER)			
	Positive Sentiment STD (For Positive Sentences)	**Neutral Sentiment STD (For Neutral Sentences)**	**Negative Sentiment STD (For Negative Sentences)**
Ahn Jeong-won	0.310	0.188	0.126
Kim Jun-wan	0.309	0.187	0.104
Lee Ik-joon	0.258	0.160	0.125
Yang Seok-hyung	0.300	0.205	0.131
Chae Song-hwa	0.216	0.191	0.172
Secondary	0.283	0.179	0.182

Episode 2 Results Comment: no big and remarkable changes can be seen passing from the first to the second episode other than a drop of positivity from Ahn Jeong-won which can be seen in both algorithms' results. Still VADER's medians are slightly higher than the roBERTa ones.

-------------- EPISODE 3 ---------------

Episode 3 – Median Sentiment by Character (roBERTa)			
	Median Positive Sentiment (For Positive Sentences)	**Median Neutral Sentiment (For Neutral Sentences)**	**Median Negative Sentiment (For Negative Sentences)**
Ahn Jeong-won	0.616	0.770	0.575
Kim Jun-wan	0.798	0.739	0.702
Lee Ik-joon	0.700	0.720	0.733
Yang Seok-hyung	0.943	0.708	0.931
Chae Song-hwa	0.838	0.689	0.565
Secondary	0.778	0.704	0.602

Episode 3 – Sentiment Standard Deviation by Character (roBERTa)			
	Positive Sentiment STD (For Positive Sentences)	**Neutral Sentiment STD (For Neutral Sentences)**	**Negative Sentiment STD (For Negative Sentences)**
Ahn Jeong-won	0.148	0.105	0.120
Kim Jun-wan	0.159	0.105	0.129
Lee Ik-joon	0.135	0.104	0.145
Yang Seok-hyung	NaN	0.087	NaN
Chae Song-hwa	0.148	0.116	0.084
Secondary	0.148	0.117	0.122

Episode 3 – Median Sentiment by Character (VADER)			
	Median Positive Sentiment (For Positive Sentences)	**Median Neutral Sentiment (For Neutral Sentences)**	**Median Negative Sentiment (For Negative Sentences)**
Ahn Jeong-won	0.593	1.0	0.561
Kim Jun-wan	0.625	1.0	0.578
Lee Ik-joon	0.710	1.0	0.688
Yang Seok-hyung	0.750	1.0	1.000
Chae Song-hwa	0.589	1.0	0.565
Secondary	0.692	1.0	0.611

Episode 3 – Sentiment Standard Deviation by Character (VADER)			
	Positive Sentiment STD (For Positive Sentences)	**Neutral Sentiment STD (For Neutral Sentences)**	**Negative Sentiment STD (For Negative Sentences)**
Ahn Jeong-won	0.344	0.205	0.004
Kim Jun-wan	0.243	0.182	0.159
Lee Ik-joon	0.225	0.174	0.172
Yang Seok-hyung	0.169	0.167	NaN
Chae Song-hwa	0.284	0.192	0.172
Secondary	0.262	0.170	0.149

Episode 3 Results Comment: there don't seem be any big changes too comparing the third episode to the other one, but there are cases of scarcity of data.

This is because Yang Seok-hyung is the character who talked less in the third episode, in fact the only acted 15 lines of which:

- Both roBERTa and VADER agree that only 1 is negative.
- roBERTa detects 13 as neutral, while VADER identifies only 9 of them as such.
- roBERTa results state that only 1 sentence is positive, while VADER ones state 5 of them as such.

Because of the lack of lines from Yang Seok-hyung some NaNs are generated since if there's only 1 element it's meaningless to calculate variability metrics.

4.3.5 Plots

To show the results in a graphical way multiple plots have been created and described before.

Since the text pages which this report will be printed on could force the plots to be a bit small and difficult to read, they will be attached to the PDF version of this report afterwards.

Check that for the example of the plots too.

Here we'll be reporting only the caption for each plot:

- **roBERTa Major Sentiment Plot**: as it stands very clear, the majority of the sentences are neutral and for this reason the percentage of neutral sentiment in the episode is higher than the other two. This is valid for every episode.
- **Sentiments Median by Character:** as shown by the number we can see a very stable trend throughout the episodes where the sentiment medians are very similar to each other. The curious numbers due to the lack of data for Yang Seok-hyung are visible in the plots as peaks between all the other bar groups.
  NOTE: only roBERTa results are shown in these plots (1st, 2nd and 3rd episode).

- **Sentiments Standard Deviation by Character:** the same data visualization strategy of the last plot applies here too and the values are still very similar to each other. The same detail about Yang Seok-hyung data is visible here too where in the third plot (for the third episode) two of the three bars of the character’s bar group are missing because of the lack of data.
- **Median Positive, Neutral and Negative Sentiment by Character:** each of these plots shows a median of the sentences classified with one specific sentiment. Nothing remarkable to say.

5. Conclusions

There are some very interesting facts which have been discovered throughout the analysis and the comparison between that and the actual footage.

The conclusions will be a summary of what has been described in detail in the report, this chapter will be divided in paragraph where each one represents one fact.

Difference Between Bag of Words and String of Words Approach

As we have already said multiple times during the analysis there's a remarkable performance difference between the two algorithms which have been used to execute the sentiment analysis and collect data about the characters' sentiment.

Of course, this difference comes from taking into consideration the context in which the words are located and ignoring it like the Bag of Words does isn't the best.

On the other side the Bag of Words approach is a lot faster and can run many times faster, but with the trade-off of not very precise scores.

Neural networks are great for this purpose (sentiment analysis), but for their complexity they require a lot of processing power and will slow down the entire process.

Lack of Data for Some Characters

One very important key factor of the results' meaning is the number of lines for each character, which has proven to be crucial to obtain correct scores that otherwise, wouldn't represent the true emotions and sentiment of the character itself.

For each episode there are some that talk more and other who talk less, and that can tell us which are the main character of the episode, because more lines mean more scenes, thus more time for one actor than the others which is coherent with the actual footage.

Example: because in the first episode Ahn Jeong-won's father dies, more scenes are dedicated to him and that part of the story.

A True Match Between Characters' Emotions and Sentiment Scores

After watching the three episodes anyone can confirm that the scores which are returned by the algorithms are coherent with the emotional state and sentiment of the characters.

Being Hospital Playlist a dramatic TV series it doesn't have many ultra-happy moments, apart from the ones in which the group of friends plays together as a musical band.

In general, we could say that although the scores obtained will not necessarily represent the exact sentiment of each single frame of the footage, it greatly summarizes them a gives a true perspective on the characters' feelings.

Even the fact that some characters like Yang Seok-hyung don't talk so much is real representation of their behaviour.

Example: being not very sociable Yang Seok-hyung doesn't talk very much in all the episodes, and the confirmation arrives from the only 15 lines he acts in the third episode.

An Almost Constant Sentiment Trend

Apart from some parts in the actual footage which show the desperation and sadness of some specific event which either traumatize or hurt the emotions of some characters, the general trend of the sentiments is very similar throughout the episodes.

Sentiments can vary by character for the same reason discussed before, but don't drastically change from episode to another.

Some event can also have an impact on the medians shown before, like the negativity of a trauma given from the loss of a patient or the discovery of being betrayed by a fiancée like it happened to Chae Song-hwa.

In conclusion we could say that the true goal of the analysis has been reached and the comparison of the sentiment scores has been crucial to understand the differences between the algorithms and which one was better based on the chosen approach too.

The sentiment of the characters has been explained through numbers obtained from the models and their coherent with the actual footage.

There are no remarkable changes of general sentiment between the episodes with the exception of some characters which could have been influenced by specific emotional/stressful events.