IS- NLP Practical Work: Song analysis

Raúl González Duarte

1. Introduction

Many times we listen to different music depending on the state of mind in which we find ourselves. Some music programs like Spotify or apple music group songs based on the feeling that the songs convey. As a person who consumes many hours of music on a daily basis, an analysis of the songs has seemed very interesting to me to see the power of natural language processing. For this work, I have chosen an artist who is quite popular and who I usually listen to, Dua Lipa; and specifically, her album Future Nostalgia.

2. Results

The libraries we need are the following: tm, textclean, dplyr, ggplot2, ggwordcloud, udpipe, kableExtra, sentiment and sentimentAnalysis. The file requirements.R will install the packages if needed.

The dataset we are going to use is located in the data folder, and corresponds to Dua Lipa songs. Once loaded, we remove the id and artist columns as they are not useful. We will analyze and classify the Album Future Nostalgia, so we fist filter the 11 songs of this album into a new data frame.

Next, we create a corpus and perform the following cleaning:

- Transforming upper to lower case words
- Replace a strange character that appear in lyrics
- Replace the contractions (don't -> do not)
- Remove the punctuations
- Remove the numbers
- Remove the stop words
- Strip the white spaces

Once we have cleaned the corpus, we create a Term Document Matrix and use it to create a data frame with the frequency of the words and with that data frame, create a word cloud for the album.

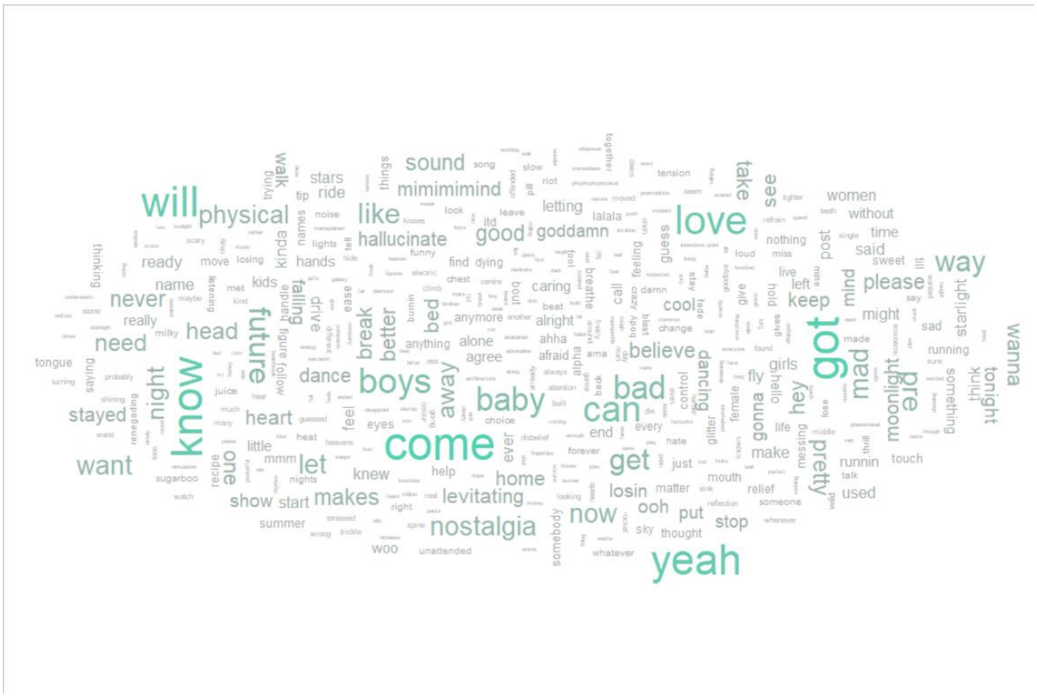
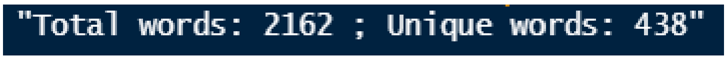


Figure 1: Album word cloud

Then we count the number of words in the album and also the number of unique words.



```
"Total words: 2162 ; Unique words: 438"
```

Figure 2: Total and unique words in the album

With the term document matrix, we also create a data frame with the 10 most frequent words in the album and plot them in a horizontal bar plot.

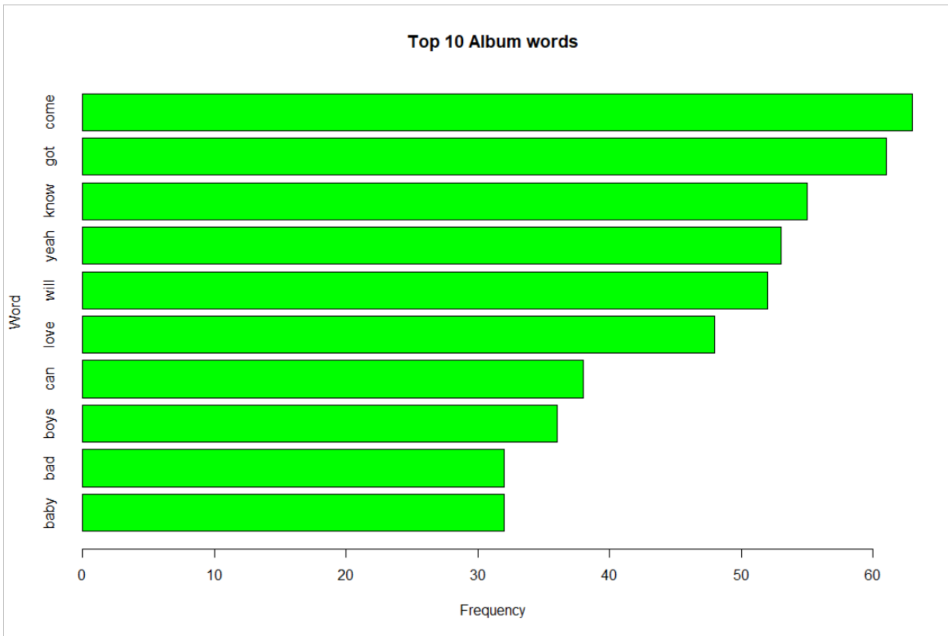
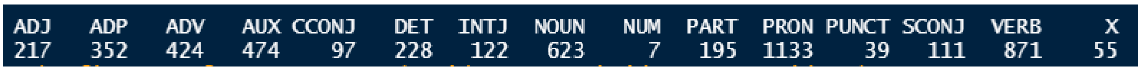


Figure 3: Top 10 album words

We can see that the most frequent words are mainly verbs and nouns.
With the library udpipe, we show and analyze the types and number of words of each type in the album and we show them with the library kableExtra.

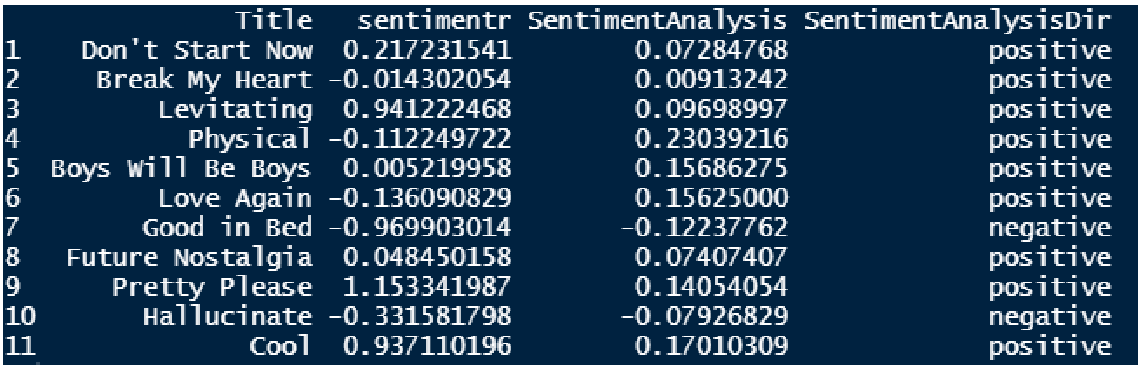


```
ADJ   ADP   ADV   AUX CCONJ   DET  INTJ  NOUN   NUM  PART  PRON PUNCT SCONJ  VERB     X
217   352   424   474    97   228   122   623     7   195  1133    39   111   871    55
```

Figure 4: Types and number of words of each type in the album

The most common types of words are pronouns, nouns and verbs, as expected due to those words are the core of any phrase.

Finally, we used 2 different libraries for classify the songs in the album based on their sentiments and we create a data frame with the results.



```
               Title   sentimentr SentimentAnalysis SentimentAnalysisDir
1     Don't Start Now  0.217231541        0.07284768             positive
2      Break My Heart -0.014302054        0.00913242             positive
3          Levitating  0.941222468        0.09698997             positive
4            Physical -0.112249722        0.23039216             positive
5   Boys Will Be Boys  0.005219958        0.15686275             positive
6          Love Again -0.136090829        0.15625000             positive
7         Good in Bed -0.969903014       -0.12237762             negative
8    Future Nostalgia  0.048450158        0.07407407             positive
9       Pretty Please  1.153341987        0.14054054             positive
10        Hallucinate -0.331581798       -0.07926829             negative
11               Cool  0.937110196        0.17010309             positive
```

Figure 5: Sentiment classification

Here we can see that depending of the library used, the songs are classified different. In my opinion, for this specific case the sentimentAnalysis library did better when classified the songs because this album has positive songs, and that library classified more correctly than the first one. These libraries classify the songs based on the words that appears in each song. Depending of the word, a score is given and finally the overall score for the song is calculated. Positive score means that the song has positive sentiment, 0 score means neutral, which is very difficult to obtain; and finally, negative score means that the song has a negative sentiment.

Last, but not least, we did a loop to create a word cloud, the top 10 most frequent words and the total and unique words for each song.

All this process can be shown more in detail in the r and rmd files with more description and discussion of the results obtained.

With this project I have learn some basics about natural language processing, how powerful it is and how much usage will have in the nearly future.

3. How to run the code

Unzip the compressed folder and open it in RStudio. Then set the folder as working directory.

For the r file, open it and click run.

As an alternative, a RMarkdown file is also attached in the folder with the html file preview with more explanations of each section. For this file, pressing the Knit button in RStudio will generate an html file with the results.

The source code can be found in the GitHub repository https://github.com/Sixtrax/IS-NLP

4. References

https://www.kaggle.com/deepshah16/song-lyrics-dataset

https://cran.r-project.org/web/packages/corpus/vignettes/corpus.html

https://www.rdocumentation.org/packages/tm/versions/0.7-8

https://rdrr.io/cran/textclean/man/replace_contraction.html

https://cran.r-project.org/web/packages/ggwordcloud/vignettes/ggwordcloud.html

https://cran.r-project.org/web/packages/udpipe/vignettes/udpipe-annotation.html

https://cran.r-project.org/web/packages/sentimentr/readme/README.html

https://cran.r-project.org/web/packages/SentimentAnalysis/vignettes/SentimentAnalysis.html