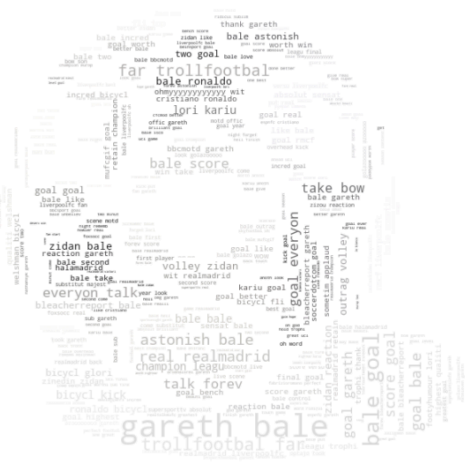
EXPLORING TWEETS USING NLP AND UNSUPERVISED LEARNING

TWITTER NLP



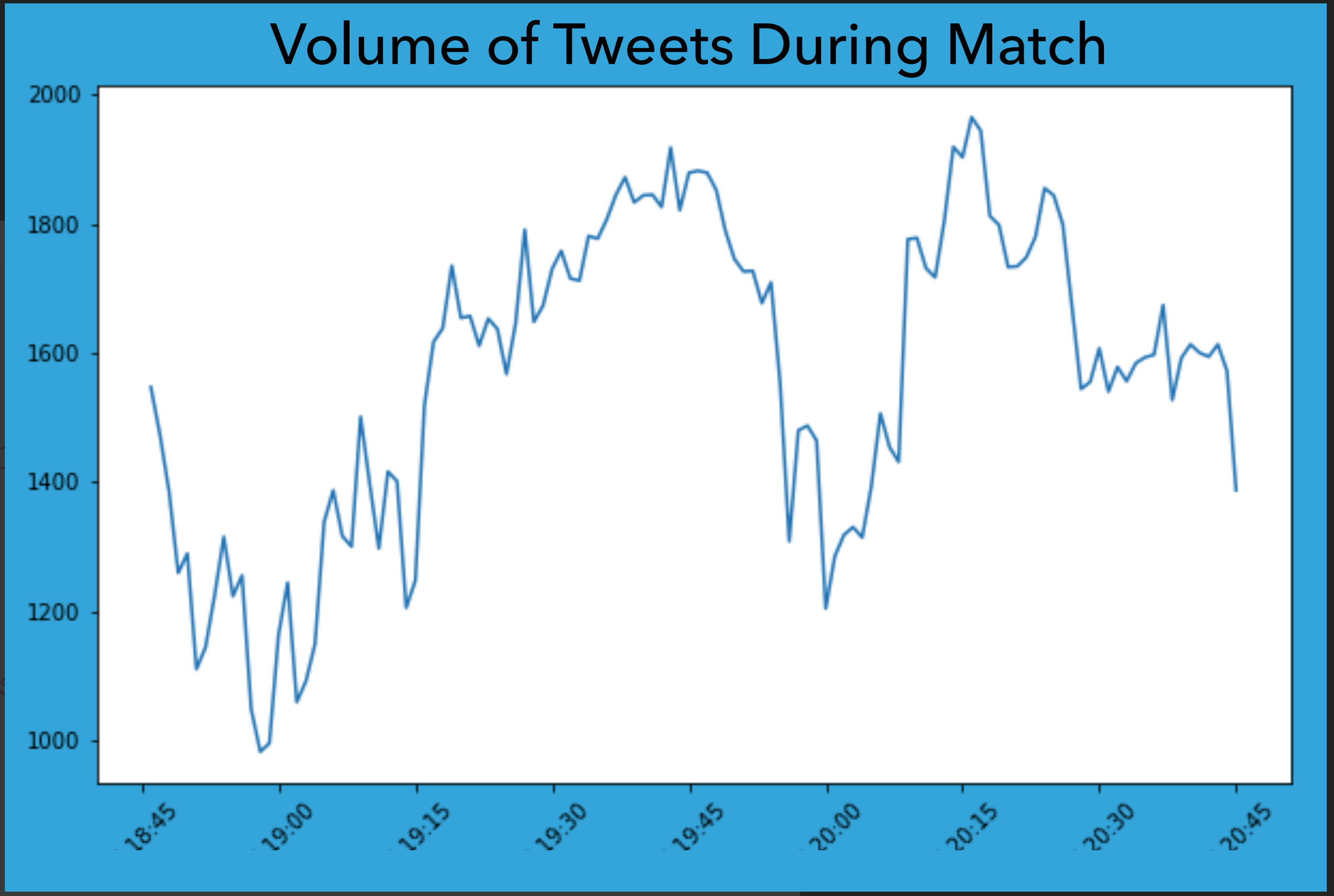
PROJECT OVERVIEW

- Project focusses on a twitter dataset containing ~350k tweets during a football match - the 2018 Champions League Final in which Real Madrid beat Liverpool 3-1.
- The main question explored as part of this project was:
  - Is it possible to use tweets to identify key events in a football match? A real world application would be to create a "match commentary" bot that could be used by media outlets to generate real time match commentary.

EXPLORATORY DATA ANALYSIS

- As the data was originally sourced from the twitter API, the fields were well documented and the data was mostly clean.
- Of the 354,586 records there were 24,202 of empty records, these were omitted from analysis
- The population was further reduced by omitting non-english tweets, leaving 187,592 tweets

{
"statuses": [
{
"created_at": "Sun Feb 25 18:11:01 +0000 20
"id": 967824267948773377,
"id_str": "967824267948773377",
"text": "From pilot to astronaut, Robert H.
American to be selected as an astronaut by any na… https
"truncated": true,
"entities": {
"hashtags": [],
"symbols": [],
"user_mentions": [],



EXPLORATORY DATA ANALYSIS

- As my intention was to create features from tweet text, the EDA focussed on text analysis rather than correlations.

- Based on frequency, the top 5 words were:

Word	Count
#uclfinal	148170
rt	123654
The	77562
a	51855
to	37562

- However, by removing “Stop Words” and producing a word cloud I gained some more interesting insights

NLP AND KMEANS

- Further pre-processing was performed, including removing hashtags and building a custom list of stop words. Tweets were reduced to Tokenized Words.
- With the Tokenized Words, features were built using:
  - CountVectorizer - Simple frequency count of each word per tweet
  - TfidVectorizer - Provides a weighted score for each word per tweet based on it's occurrence in the wider population
- Given the lack of labels for this data I opted to use KMeans to cluster the data. I ran various iterations of the model with differing parameters for both type of features to identify an optimal combination.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

NLP AND KMEANS

- After identifying a good combination of parameters the Elbow Method was used to identify the optimal number of clusters which was 8. Of these 8 there were 6 with very clear topics, this is demonstrated through sample tweets from a number of clusters below. The topic/cluster titles were manually established.

Cluster 0 - No particular topic

1. real 1 - 1 southampton #ucl2018 #uclfinal
2. just got in from playing cricket. come on @lfc #uclfinal #ynwa #allezallezallez
3. so many people were waiting for a salah goal but ramos, the destroyer of dreams happened. my sis just called him thanos😭😭😂 #uclfinal

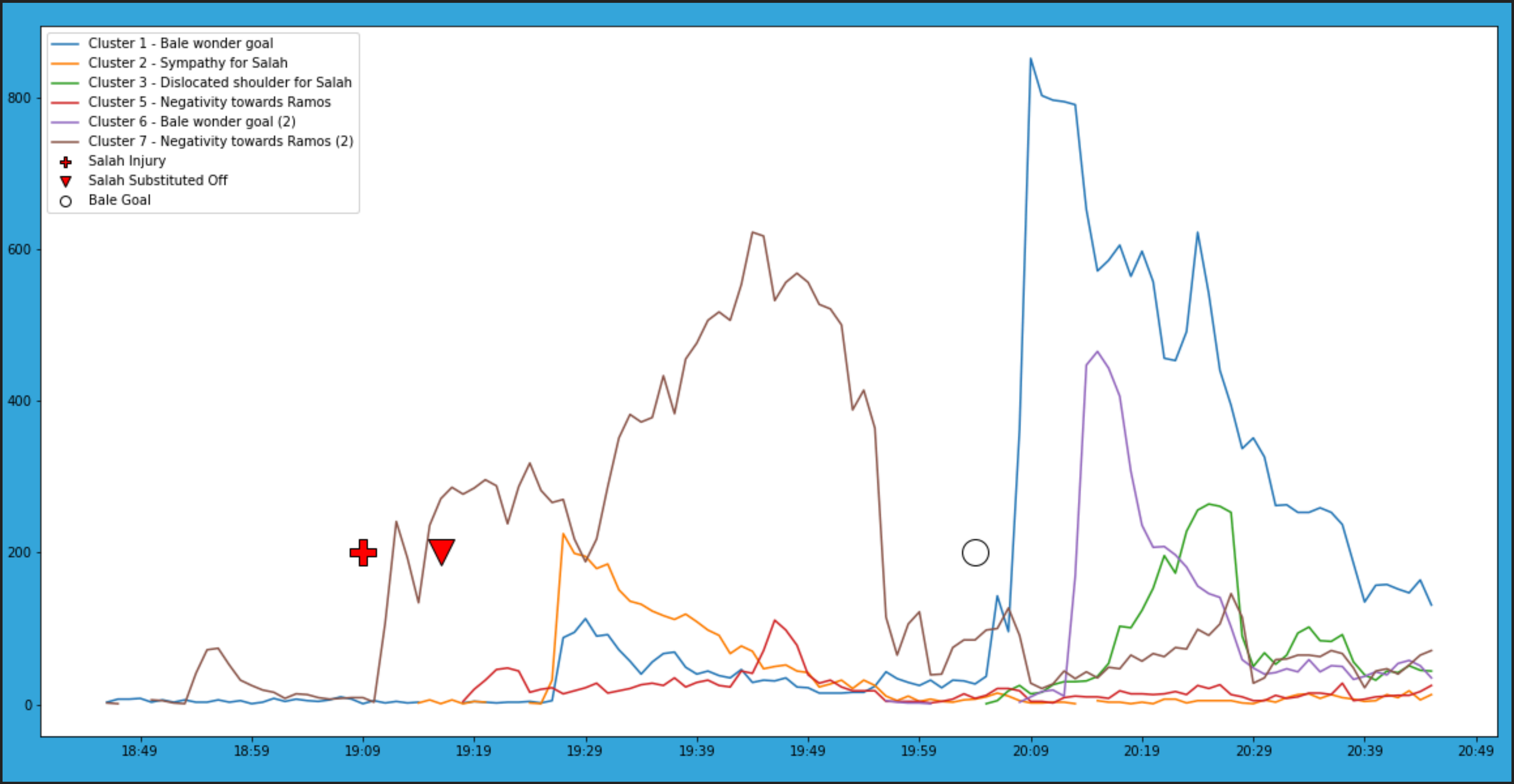
Cluster 2 - Sympathy for Salah

1. a sad way for salah's season to finish 😢 #uclfinal
2. so sad for salah, his best season ends with the worst way... 😢😢 #uclfinal
3. this is sad https://t.co/ymexxm4zgf

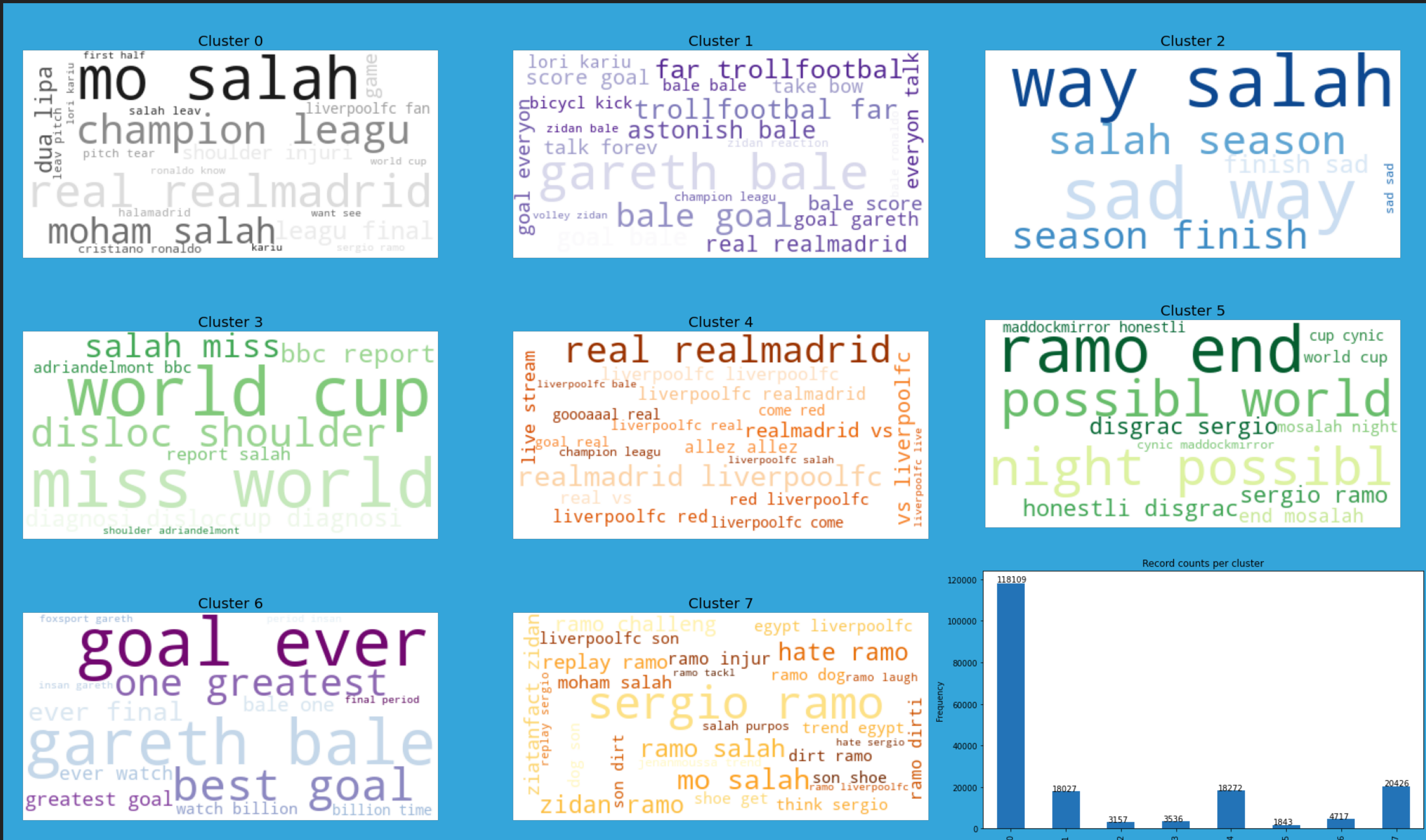
Cluster 6 - Bale Goal

1. best goal i've ever seen
2. easily one of the best goals i’ve ever seen
3. that bale goal. one of the best i’ve ever seen. what a game! #uclfinal

NLP AND KMEANS



NLP AND KMEANS



FUTURE WORK

- Cluster 0 seems to hold a lot of information that is yet to be unlocked, this needs further analysis
- Explore “good” clusters in more depth and establish what impact retweets have in defining a cluster
- Combine KMeans with Topic Modelling to label and enhance clusters
- Consider how findings can be transformed into a classification model that could work with real time data