Analysis of urban space emotions during FIFA championship in Russia

Jounaid DJOUDI

[a]Engineering School, ESIEE Paris, France
[b]Saint Petersburg, National Research University of Information Technologies, Mechanics and Optics, Russia.
[c]E-mail school : jounaid.djoudi@edu.esiee.fr
[d]E-mail pro :djoudi.jounaid@gmail.com

Abstract

In this paper, we present a sentiment analysis on Twitter about FIFA World Cup in Russia . The data set consists of tweets with several parameters such as id, text, coordinates, lang and others. The goal of this paper is to show steps until prediction of sentiment that we have chosen : joy, hope, love, sadness, anger, neutral, joy/hope and love/sadness.

Keywords :sentiment analysis, text mining, Twitter, world cup, Python ;

1. Introduction and Motivation

Nowadays social network are very important for datas scientists because there are more datas and information. For example we decided to work on Twitter to realize sentiment analysis because lot of people exprim their opinion and sentiment and if we analyze human language such as words or emoticons we can explain importance of a mega-event. Indeed FIFA world football championship taking place in Russia on June 14 - July 15 is a mega-event. We want to know if FIFA world cup in Russia provides more intensive emotions positive or negative . Does the mega-event increase urban happiness or makes people regret the days of the championship? We are analyzing people's emotions of the FIFA championship in eleven cities of Russia. Moreover we want to know what kind of emotional responses provoke this mega-event. We had different methods to start this project such as sentiment analysis on human language and non-verbal items such as emoticons and represent our result on a map.

2. Related Works

Nowadays a lot of datas are generated everyday so sentiment analysis and text mining methods are used in a lot of domains to exploit this huge datas. Indeed these are techniques very interesting to analyse human language and show how our society change and react during a big event so the best way to see that is to analyze social network, Twitter in our case. Many works have been done before us about sentiment analysis.

Indeed in 2011 [1] there was a study about sentiment analysis on Twitter and they analyzed the most popular event for english tweets and we realize a prediction about if the tweet is positive negative or neutral.
In 2016 [3] there was an article which talks about sentiment analysis on twitter about World Cup of football in 2014 so it is a similar project but they decided to predict polarity of tweets and not real sentiment like us. In addition there was another article in 2016 [2] which about sentiment analysis on Twitter about gold standard for the Brexit referendum. Their goal were to annotate 2000 Twitter messages which constituted their train base to realize prediction with an algorithm of machine learning.
In 2017 [4] there was also a study about sentiment analysis on Twitter's dataset and they decided to realize prediction about extreme positive, positive, extreme negative and neutral of each tweet this scientific article

give us to do sentiment analyse about feeling such as joy, anger and other to do different things.
In addition in 2017 [5] there was also another scientific article interesting which speaks about sentiment analysis on Twitter but on a mega-event : US Presidentials the goal of this article is to show how people react before and after the election day. In addition in this article they predict the behaviour of people. It is interesting because it is difficult to predict that they need to do a very good filter to collect tweets very relevant. This article permits us to choose our keywords to filter our scraping code very precisely.

3. Materials and Methods

We choose Twitter to collect datas for analyze urban space emotions during Fifa world cup in Russia
We use differents technologies : Google App Engine,several modules Python and MongoDB to achieve our goals.

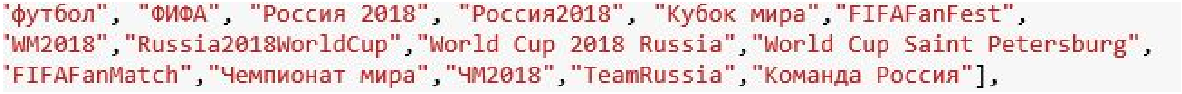
First we decided to use Google App Engine because it is a technology very interesting to improve computing power and work conditions because RAM and CPU of our computer is not sufficient for big datasets such as big JSON file. Google App Engine require SSH connection to guarantee authenticity, confidentiality and integrity of our datas. We use Putty because it permit to secure sessions on distant computer using SSH.

Second we decided to use Python to realize our goals because Python has several packages very useful to collect datas on Twitter, to do sentiment analysis, to create map.
The most important packages used are tweepy, pandas, nltk, folium and emoji.
Tweepy permit accessing to the Twitter API and collect tweets on Twitter.
Then pandas permit to use easily different data structures (for example json, csv or dataframe) and data analysis tools.
In addition nltk is an interesting package because we can do analysis on words or sentences about different languages it is very useful for sentiment analysis and preprocessing of our tweets.
Folium permit to create interactive map to visualize results of our preprocessing with Python.
Finally we use emoji because this package transform emoticons in a good format to realize sentiment analysis.

Third MongoDB permit us to store our tweets collected in collection, remove useless datas (such as a parameter of tweet is null) and download them in json file easily.

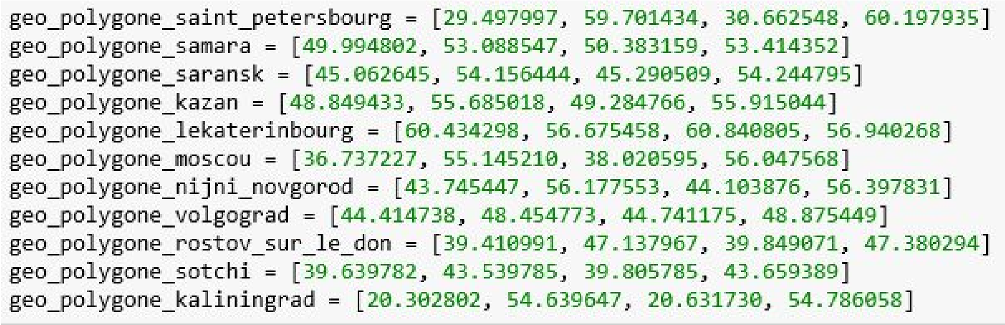
We have established different methods throughout this project.

The first step is to collect datas on Twitter about Fifa World Cup in Russia.
To realize this we must create Twitter account and then create an application to obtain our identification variables to access Twitter API.
Then we need to create a class call **“MyListener”** that permit to write each tweet collected in a json file.
We use also filter to stream all tweets with words (***Fig.1***) and locations (***Fig.2)*** that we have chosen.
Indeed we have defined a rectangle for each city of the World Cup in Russia which represent coordinate and thus permit to match all tweet only in eleven cities defined.



```
'футбол", "ФИФА", "Россия 2018", "Россия2018", "Кубок мира","FIFAFanFest",
'WM2018","Russia2018WorldCup","World Cup 2018 Russia","World Cup Saint Petersburg",
'FIFAFanMatch","Чемпионат мира","ЧМ2018","TeamRussia","Команда Россия"],
```

Fig.1



```
geo_polygone_saint_petersbourg = [29.497997, 59.701434, 30.662548, 60.197935]
geo_polygone_samara = [49.994802, 53.088547, 50.383159, 53.414352]
geo_polygone_saransk = [45.062645, 54.156444, 45.290509, 54.244795]
geo_polygone_kazan = [48.849433, 55.685018, 49.284766, 55.915044]
geo_polygone_lekaterinbourg = [60.434298, 56.675458, 60.840805, 56.940268]
geo_polygone_moscou = [36.737227, 55.145210, 38.020595, 56.047568]
geo_polygone_nijni_novgorod = [43.745447, 56.177553, 44.103876, 56.397831]
geo_polygone_volgograd = [44.414738, 48.454773, 44.741175, 48.875449]
geo_polygone_rostov_sur_le_don = [39.410991, 47.137967, 39.849071, 47.380294]
geo_polygone_sotchi = [39.639782, 43.539785, 39.805785, 43.659389]
geo_polygone_kaliningrad = [20.302802, 54.639647, 20.631730, 54.786058]
```

Fig.2

Once the json file comprising tweets with different attributes has been generated, the second step is to create a MongoDB database.
We must import our json generated in a collection in the database and remove all datas which no contain coordinate of tweet because we want to work only with tweets geolocated.
Then we read our json file with Python to work with it but we can work easily with this file because we need to realize preprocessing and classify these datas
.
The third step is to realize preprocessing on our dataset. First we classify tweets and their other parameters such as date of tweet, text, id or coordinates by language and use only tweets which are in english and russian. To realize this we created a function called **"store_lang"**.
Then to difference each tweet we need to realize preprocessing on attribute **"id"** because there are useless element to make a good identification with a good format. To realize this we created two functions called **"preprocess_id"** and **"good_id"** which permit also to store all id in two lists for english and russian. We stored also text of tweets in two lists for english and russian to realize preprocessing on language human.
Another step of preprocessing is lemmatisation which is important because it permit to keep words of the same family.
We created a function called **"lemmatiz"** which permit to realize also some important preprocessing before lemmatisation such as remove punctuation, digit, and web link and convert emoticons in good format to analyze them in our sentiment analysis. In addition we created a function called **"lowerword"** which permit to transform all words in lowercase. It is important because if a same word is first time in lowercase and second time in uppercase for the program it will be a different word.
Then we created a function called **"delete_stw"** which consist to compare list of tweets of each language with list of stopwords, if a word is in a tweet and list of stopwords we delete it from the tweet. We used NLTK downloader which permits to download differents packages such as list of stopwords for english and russian. Stopwords are words which don't add a meaning to a sentence when we analyze a text that's why we exclude these words from dictionary of words.
To keep root form of words we need to create a stemmer for each language (english and russian in our case). Indeed ***"stemming"*** is a sort of normalization because several variants of word have the same meaning. The reason why it is important to realize this task is that allows to shorten the search and thus normalize sentences. We need to create a function called **"stem_tweet"** which permits to realize stemming on each tweet in english and russian. Then we created two dictionary one in english and another in russian whose keys are sentiment and values words of this sentiment. Feelings that we chose are joy, hope, love, sadness, anger, neutral, joy/hope and love/sadness. We created a function called **"stem_lexical"** that permits to realize stemming on both dictionary that we created previously but also another function called **"length_lexical_word"** that allows

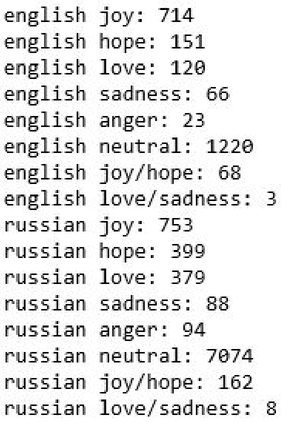
to show how many words we have in our both dictionaries for each sentiment.

The fourth step is to realize a prediction of sentiment on each tweet. To achieve this step we realize firstly a function called **“count”**, in this function we created a dictionary whose keys are the name of sentiment and for each the value 0. Thus if the word of the tweet corresponds to one of the words of dictionary’s feeling, the counter corresponding to this feeling will be incremented. This function allows to return a dictionary with a numeric value which equals to number of matches for each sentiment words of tweet and dictionary of sentiment with root form. In addition a function called **“predict_sentiment”** has been created to predict sentiment for one tweet after called our functions of preprocessing. Thanks to these functions we are be able to prediction dominant feeling for each tweet whatever the language such as english and russian in our case. In addition we can know how many tweets have in each sentiment. To complete this step we created a dictionary whose keys are id of tweet and values dominant sentiment which is predicted.

The last step is to create a map using coordinates of tweet and the result of predictions. To realize this we stored all latitude and longitude in a list for each language and put dominant sentiment of each tweet in another list. Then we assign a color for each sentiment to put a point with correct color on our map.

4. Results

We have collected 20018 tweets usable. Indeed there was a lot of tweets that we could not use because they did not contain coordinates of tweet so for us it was useless. Then we used only 11322 tweets because we analyzed only russian and english language, it will have been very long to create a list of words for each sentiment in every language.



```
english joy: 714
english hope: 151
english love: 120
english sadness: 66
english anger: 23
english neutral: 1220
english joy/hope: 68
english love/sadness: 3
russian joy: 753
russian hope: 399
russian love: 379
russian sadness: 88
russian anger: 94
russian neutral: 7074
russian joy/hope: 162
russian love/sadness: 8
```

Fig.3

In the figure 3 above we see our prediction of dominant sentiment of each tweet. We notice that there are more positive tweets than negative tweets. Thus we can say that FIFA world cup in Russia provokes positive emotional response on russian people but also foreigners people who speaks english. We can suppose that people liked share positive feelings with others fans and watched football match at the stadium or at the FIFA Fan Fest. We can say that there are few tweets which are negative because the FIFA world cup was well organized and there were no bad events.

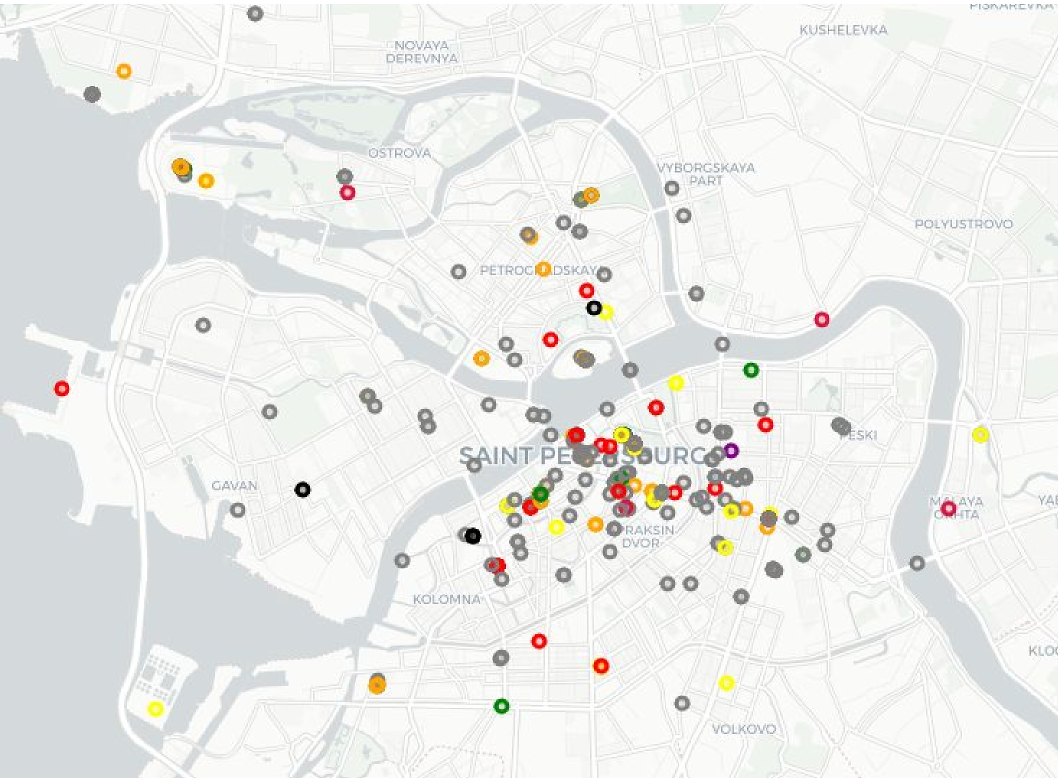


Fig.4

The figure 4 above we put all tweets in english on a map. We notice that there are a lot of tweets at nevsky avenue because there are FIFA Fan Fest and russian and foreigners people wen them every match during world cup. We associated one colour to each sentiment so for neutral point is grey, for joy point is orange, for anger point is crimson, for love point is red, for sadness point is black, for hope point is yellow, for joy/hope point is green and for sadness/love point is purple. We noticed that there are a lot of different sentiment next to FIFA Fan Fest because during football match people have different emotions if their favourite team win or lose at the moment of a tweet.

.

5. Discussion

We realized all tasks of our project but we would have been able to do improvement if we had more time.

There are many others ideas that we would like to test if we continue this project.

First to create a training base with a lot of tweets which have been annotated with each sentiment. Indeed this task could be interesting because it would allow to test our own algorithm and compare with a classification algorithm such as Naives Bayes and see the accuracy of our algorithm.

Second another idea is to realize sentiment analysis on another language to show sentiment of all communities.

Moreover we would be able to do a map before, during and after an important date during FIFA world cup for example semi-final or match of Russia to show evolution of tweets and feelings we are dominant.

Finally we would be able to do better visualization for results for example with Flask. Indeed this framework we would have allowed to realize a search engine of tweets by feelings and do link with our database MongoDB, it would have been useful.

6. Conclusion

This project can be improved in many ways and differents methods and implementations, but it show a new vision of FIFA world cup about the way that people lives during this mega-event. We tried to do a work understand by all and realize a visualization with map.
This project was a two months and half project and differents tasks to realize. It could be use by other people such as a beginning for a big project of sentiment analysis concept. This project permits us to see that sentiment analysis is a vast domain and there are different ways to realize that thanks to a lot of Python's packages.

7. Acknowledgement

We thank our tutors and collaborators Alexandra Nenko, Artem Konuchov, Marina Petrova and Polina Kashina.
We thank Alexandra Klimova for joining us in this project.
Finally we want to thank ITMO for their welcome and to allow us to work on a concrete project related to our sector.

References

[1] Thelwall, M., Buckley, K., & Paltoglou, G. (2011). Sentiment in Twitter events. Journal of the American Society for Information Science and Technology from https://doi.org/10.1002/asi.21462

[2] Hürlimann, M., Davis, B., Cortis, K., Freitas, A., Handschuh, S., & Fernández, S. (2016). A Twitter Sentiment Gold Standard for the Brexit Referendum. In Proceedings of the 12th International Conference on Semantic Systems - SEMANTiCS 2016. https://doi.org/10.1145/2993318.2993350

[3] Barnaghi, P., Ghaffari, P., & Breslin, J. G. (2016). Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment. In Proceedings - 2016 IEEE 2nd International Conference on Big Data Computing Service and Applications, BigDataService 2016. https://doi.org/10.1109/BigDataService.2016.36

[4] Parveen, H., & Pandey, S. (2017). Sentiment analysis on Twitter Data-set using Naive Bayes algorithm. In Proceedings of the 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology, iCATccT 2016 from https://doi.org/10.1109/ICATCCT.2016.7912034

[5] Yaqub, U., Chun, S. A., Atluri, V., & Vaidya, J. (2017). Sentiment based Analysis of Tweets during the US Presidential Elections. In Proceedings of the 18th Annual International Conference on Digital Government Research - dg.o '17. https://doi.org/10.1145/3085228.3085285

[6] MongoDB Docs : https://docs.mongodb.com/

[7] Twitter Developer : https://developer.twitter.com/en.html

[8] Natural Language Toolkit : https://www.nltk.org/

[9] Python Documentation : https://docs.python.org/fr/3/

[10] Tweepy Documentation : http://docs.tweepy.org/en/v3.5.0/

[11] Google Compute Engine Documentation : https://cloud.google.com/compute/docs/