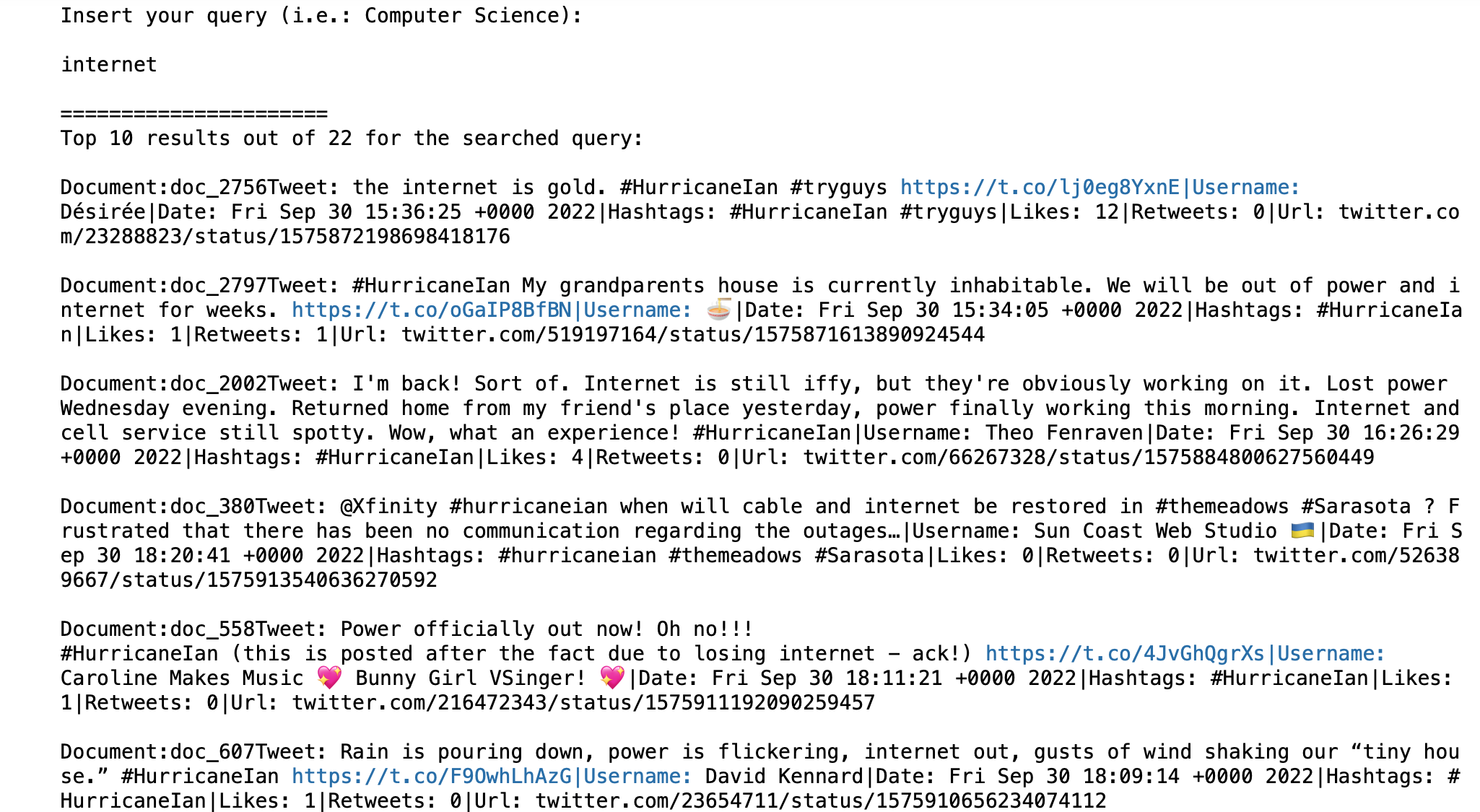
# IR Project: Part 2

We are continuing working with the part 1 of the project. The final of Part 1 was a ranked search (example of word: internet):

As we can see we print the tweets like this:

**Doc\_Name | Tweet | Username | Date | Hashtags | Likes | Retweets | Url**

After that we make the 5 queries. Example of our first query (flood):

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

After this we start this new part where we load one dataframe with the queries given and another one with the queries chosen by us and decide if is relevant or not (for our queries):

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Then we have implemented different functions to calculate:

* Precision@K
* Recall@K
* F1 score
* **A**verage Precision@K
* NDGC
* MAP (Mean Average Precision)
* MRR (Mean Reciprocal Rank)

We calculate this for each query of each Dataframe, we calculate precision, recall and F1 score separately. And the results of the evaluation of the system are the following:

Query : Landfall in South Carolina

==> Precision@5: 1.0

==> Recall@5: 0.5

==> Average Precision@5: 0.5

==> F1-score of first 5: 0.6666666666666666

==> NDCG@5: 1.0

Query : Help and recovery during the hurricane disaster

==> Precision@5: 0.8

==> Recall@5: 0.4

==> Average Precision@5: 0.38

==> F1-score of first 5: 0.5333333333333333

==> NDCG@5: 0.8539

Query : Floodings in South Carolina

==> Precision@5: 1.0

==> Recall@5: 0.5

==> Average Precision@5: 0.5

==> F1-score of first 5: 0.6666666666666666

==> NDCG@5: 1.0

----------------------------------------------------

==> Mean Average Precision (MAP) @5: 0.45999999999999996

==> Mean Reciprocal Rank (MRR) @5: 1.0

And here about our queries:

Query 1: flood

==> Precision@5: 0.8

==> Recall@5: 0.5

==> Average Precision@5: 0.33958333333333335

==> F1-score of first 5: 0.6153846153846154

==> NDCG@5: 0.6608

Query 2: emergency

==> Precision@5: 0.4

==> Recall@5: 0.2857142857142857

==> Average Precision@5: 0.19999999999999998

==> F1-score of first 5: 0.3333333333333333

==> NDCG@5: 0.4704

Query 3: hurricane

==> Precision@5: 0.4

==> Recall@5: 0.4

==> Average Precision@5: 0.18

==> F1-score of first 5: 0.4000000000000001

==> NDCG@5: 0.3452

Query 4: florida

==> Precision@5: 0.6

==> Recall@5: 0.6

==> Average Precision@5: 0.2866666666666666

==> F1-score of first 5: 0.6

==> NDCG@5: 0.4469

Query 5: landfall

==> Precision@5: 0.8

==> Recall@5: 0.5714285714285714

==> Average Precision@5: 0.5714285714285714

==> F1-score of first 5: 0.6666666666666666

==> NDCG@5: 0.8688

----------------------------------------------------

==> Mean Average Precision (MAP) @5: 0.31553571428571425

==> Mean Reciprocal Rank (MRR) @5: 0.6666666666666667

Here we can say that query 1 and 5 are more precise than the others with a k= 5, but also query 4 has a good precision associated with the best recall. That will be interesting to do a P/R graph. However the better recall of query 4 is not enough as we could see by looking at the f1-score that is the highest for query 5 that has also the highest average prediction.

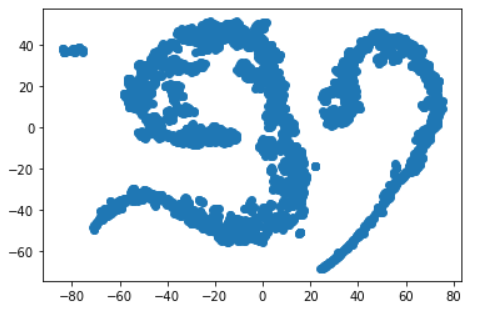
If we care more about finding the best result in the first place, query 5 is the best because has the highest NDCG, while query 3 is the worst, in fact we can find two non-relevant documents in first places.

Also mean average precision is lower than the given queries.

Mean Reciprocal rank is also lower because not all the queries found a relevant doc in first place.

We can conclude that the query Landfall in South Carolina is the one with the better results and from our selected queries, the query one ( landfall ) is the one with better results.

After this we are making a plot of vectors-tweets, this is the result:



THIS IS THE REPOSITORY: <https://github.com/JordiBadia01>