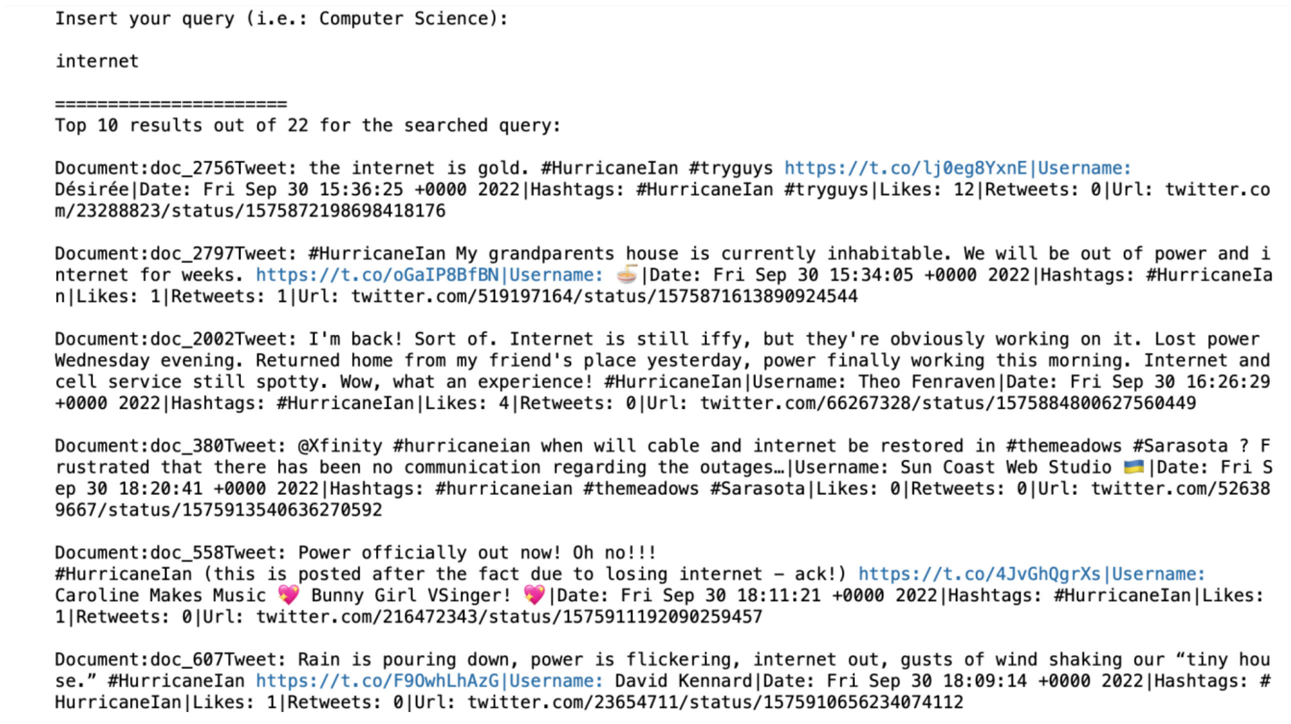
Jordi Badia **IRWA** 18/10/22
Nil Agell
Cristian Bassotto

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):

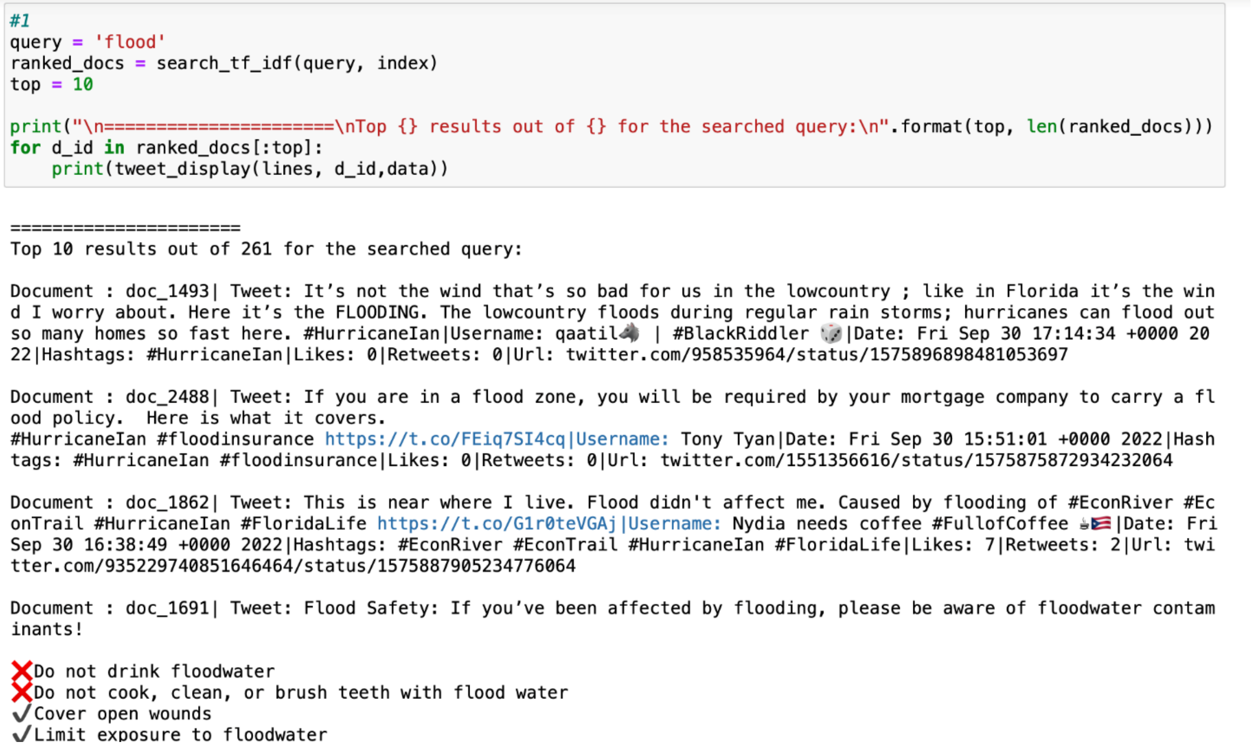


```
Insert your query (i.e.: Computer Science):

internet

======================
Top 10 results out of 22 for the searched query:

Document:doc_2756Tweet: the internet is gold. #HurricaneIan #tryguys https://t.co/lj0eg8YxnE|Username:
Désirée|Date: Fri Sep 30 15:36:25 +0000 2022|Hashtags: #HurricaneIan #tryguys|Likes: 12|Retweets: 0|Url: twitter.co
m/23288823/status/1575872198698418176

Document:doc_2797Tweet: #HurricaneIan My grandparents house is currently inhabitable. We will be out of power and i
nternet for weeks. https://t.co/oGaIP8BfBN|Username: 🍜|Date: Fri Sep 30 15:34:05 +0000 2022|Hashtags: #HurricaneIa
n|Likes: 1|Retweets: 1|Url: twitter.com/519197164/status/1575871613890924544

Document:doc_2002Tweet: I'm back! Sort of. Internet is still iffy, but they're obviously working on it. Lost power
Wednesday evening. Returned home from my friend's place yesterday, power finally working this morning. Internet and
cell service still spotty. Wow, what an experience! #HurricaneIan|Username: Theo Fenraven|Date: Fri Sep 30 16:26:29
+0000 2022|Hashtags: #HurricaneIan|Likes: 4|Retweets: 0|Url: twitter.com/66267328/status/1575884800627560449

Document:doc_380Tweet: @Xfinity #hurricaneian when will cable and internet be restored in #themeadows #Sarasota ? F
rustrated that there has been no communication regarding the outages…|Username: Sun Coast Web Studio 🇺🇦|Date: Fri S
ep 30 18:20:41 +0000 2022|Hashtags: #hurricaneian #themeadows #Sarasota|Likes: 0|Retweets: 0|Url: twitter.com/52638
9667/status/1575913540636270592

Document:doc_558Tweet: Power officially out now! Oh no!!!
#HurricaneIan (this is posted after the fact due to losing internet - ack!) https://t.co/4JvGhQgrXs|Username:
Caroline Makes Music 💖 Bunny Girl VSinger! 💖|Date: Fri Sep 30 18:11:21 +0000 2022|Hashtags: #HurricaneIan|Likes:
1|Retweets: 0|Url: twitter.com/216472343/status/1575911192090259457

Document:doc_607Tweet: Rain is pouring down, power is flickering, internet out, gusts of wind shaking our “tiny hou
se.” #HurricaneIan https://t.co/F9OwhLhAzG|Username: David Kennard|Date: Fri Sep 30 18:09:14 +0000 2022|Hashtags: #
HurricaneIan|Likes: 1|Retweets: 0|Url: twitter.com/23654711/status/1575910656234074112
```

As we can see we print the tweets like this:

Doc_Name | Tweet | Username | Date | Hashtags | Likes | Retweets | Url

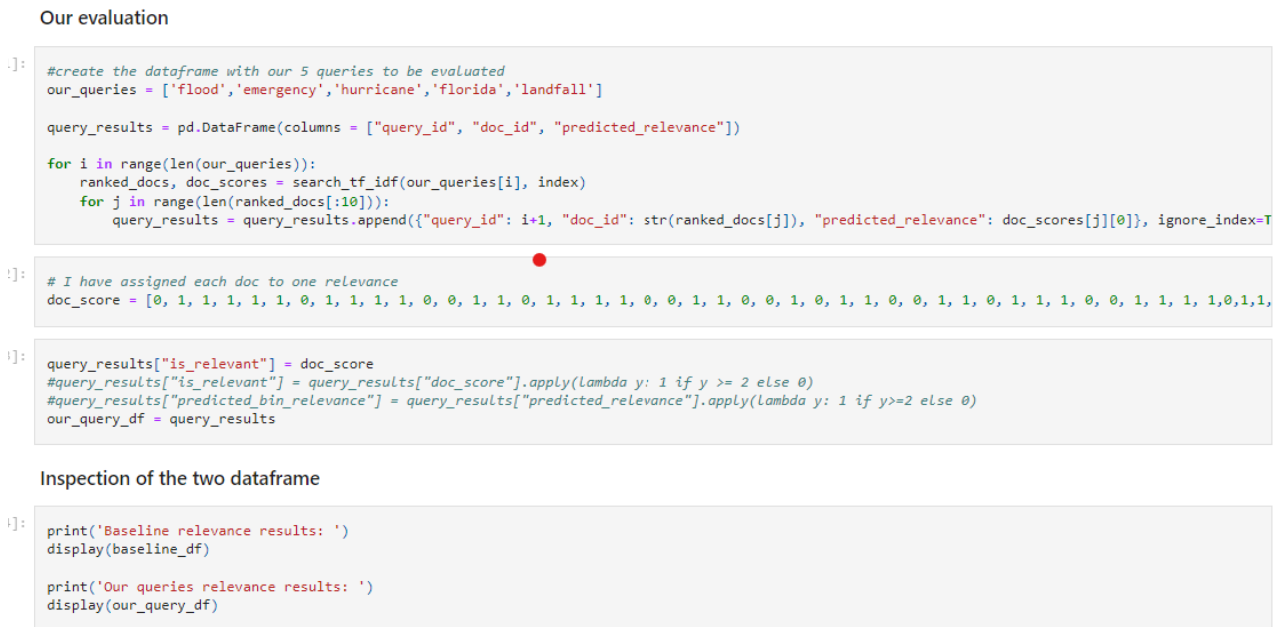
After that we make the 5 queries. Example of query 1:



```
#1
query = 'flood'
ranked_docs = search_tf_idf(query, index)
top = 10

print("\n=====================\nTop {} results out of {} for the searched query:\n".format(top, len(ranked_docs)))
for d_id in ranked_docs[:top]:
    print(tweet_display(lines, d_id,data))
```

```
=====================
Top 10 results out of 261 for the searched query:

Document : doc_1493| Tweet: It’s not the wind that’s so bad for us in the lowcountry ; like in Florida it’s the win
d I worry about. Here it’s the FLOODING. The lowcountry floods during regular rain storms; hurricanes can flood out
so many homes so fast here. #HurricaneIan|Username: qaatil🐺 | #BlackRiddler 🎲|Date: Fri Sep 30 17:14:34 +0000 20
22|Hashtags: #HurricaneIan|Likes: 0|Retweets: 0|Url: twitter.com/958535964/status/1575896898481053697

Document : doc_2488| Tweet: If you are in a flood zone, you will be required by your mortgage company to carry a fl
ood policy.  Here is what it covers.
#HurricaneIan #floodinsurance https://t.co/FEiq7SI4cq|Username: Tony Tyan|Date: Fri Sep 30 15:51:01 +0000 2022|Hash
tags: #HurricaneIan #floodinsurance|Likes: 0|Retweets: 0|Url: twitter.com/1551356616/status/1575875872934232064

Document : doc_1862| Tweet: This is near where I live. Flood didn't affect me. Caused by flooding of #EconRiver #Ec
onTrail #HurricaneIan #FloridaLife https://t.co/G1r0teVGAj|Username: Nydia needs coffee #FullofCoffee ☕🇵🇷|Date: Fri
Sep 30 16:38:49 +0000 2022|Hashtags: #EconRiver #EconTrail #HurricaneIan #FloridaLife|Likes: 7|Retweets: 2|Url: twi
tter.com/935229740851646464/status/1575887905234776064

Document : doc_1691| Tweet: Flood Safety: If you’ve been affected by flooding, please be aware of floodwater contam
inants!

❌Do not drink floodwater
❌Do not cook, clean, or brush teeth with flood water
✔Cover open wounds
✔Limit exposure to floodwater
```

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.



Our evaluation

```
#create the dataframe with our 5 queries to be evaluated
our_queries = ['flood','emergency','hurricane','florida','landfall']

query_results = pd.DataFrame(columns = ["query_id", "doc_id", "predicted_relevance"])

for i in range(len(our_queries)):
    ranked_docs, doc_scores = search_tf_idf(our_queries[i], index)
    for j in range(len(ranked_docs[:10])):
        query_results = query_results.append({"query_id": i+1, "doc_id": str(ranked_docs[j]), "predicted_relevance": doc_scores[j][0]}, ignore_index=T
```

```
# I have assigned each doc to one relevance
doc_score = [0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,0,1,1,
```

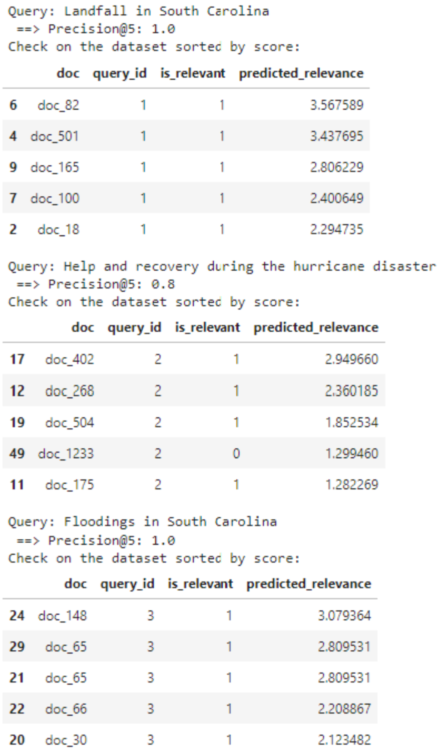
```
query_results["is_relevant"] = doc_score
#query_results["is_relevant"] = query_results["doc_score"].apply(lambda y: 1 if y >= 2 else 0)
#query_results["predicted_bin_relevance"] = query_results["predicted_relevance"].apply(lambda y: 1 if y>=2 else 0)
our_query_df = query_results
```

Inspection of the two dataframe

```
print('Baseline relevance results: ')
display(baseline_df)

print('Our queries relevance results: ')
display(our_query_df)
```

Here above we set our query dataframe, defining us the relevance.

Then we have implemented different functions to calculate the precision ,recall, the F1 score, the average precision, and NDGC, the MAP and MRR. 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
Check on the dataset sortec by score:
```

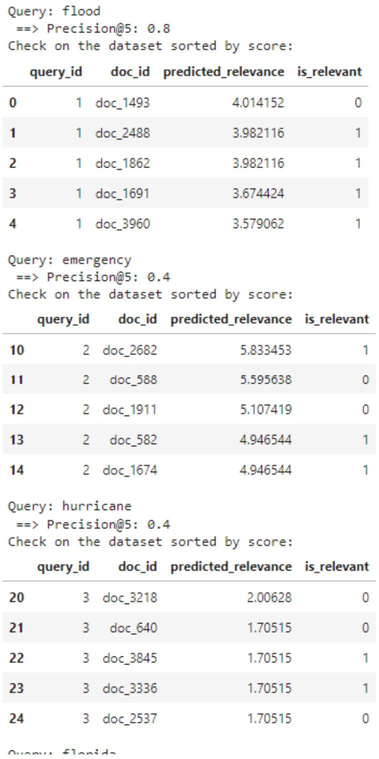
	doc	query_id	is_relevant	predicted_relevance
6	doc_82	1	1	3.567589
4	doc_501	1	1	3.437695
9	doc_165	1	1	2.806229
7	doc_100	1	1	2.400649
2	doc_18	1	1	2.294735

```
Query: Help and recovery during the hurricane disaster
 ==> Precision@5: 0.8
Check on the dataset sortec by score:
```

	doc	query_id	is_relevant	predicted_relevance
17	doc_402	2	1	2.949660
12	doc_268	2	1	2.360185
19	doc_504	2	1	1.852534
49	doc_1233	2	0	1.299460
11	doc_175	2	1	1.282269

```
Query: Floodings in South Carolina
 ==> Precision@5: 1.0
Check on the dataset sortec by score:
```

	doc	query_id	is_relevant	predicted_relevance
24	doc_148	3	1	3.079364
29	doc_65	3	1	2.809531
21	doc_65	3	1	2.809531
22	doc_66	3	1	2.208867
20	doc_30	3	1	2.123482



```
Query: flood
 ==> Precision@5: 0.8
Check on the dataset sorted by score:
```

	query_id	doc_id	predicted_relevance	is_relevant
0	1	doc_1493	4.014152	0
1	1	doc_2488	3.982116	1
2	1	doc_1862	3.982116	1
3	1	doc_1691	3.674424	1
4	1	doc_3960	3.579062	1

```
Query: emergency
 ==> Precision@5: 0.4
Check on the dataset sorted by score:
```

	query_id	doc_id	predicted_relevance	is_relevant
10	2	doc_2682	5.833453	1
11	2	doc_588	5.595638	0
12	2	doc_1911	5.107419	0
13	2	doc_582	4.946544	1
14	2	doc_1674	4.946544	1

```
Query: hurricane
 ==> Precision@5: 0.4
Check on the dataset sorted by score:
```

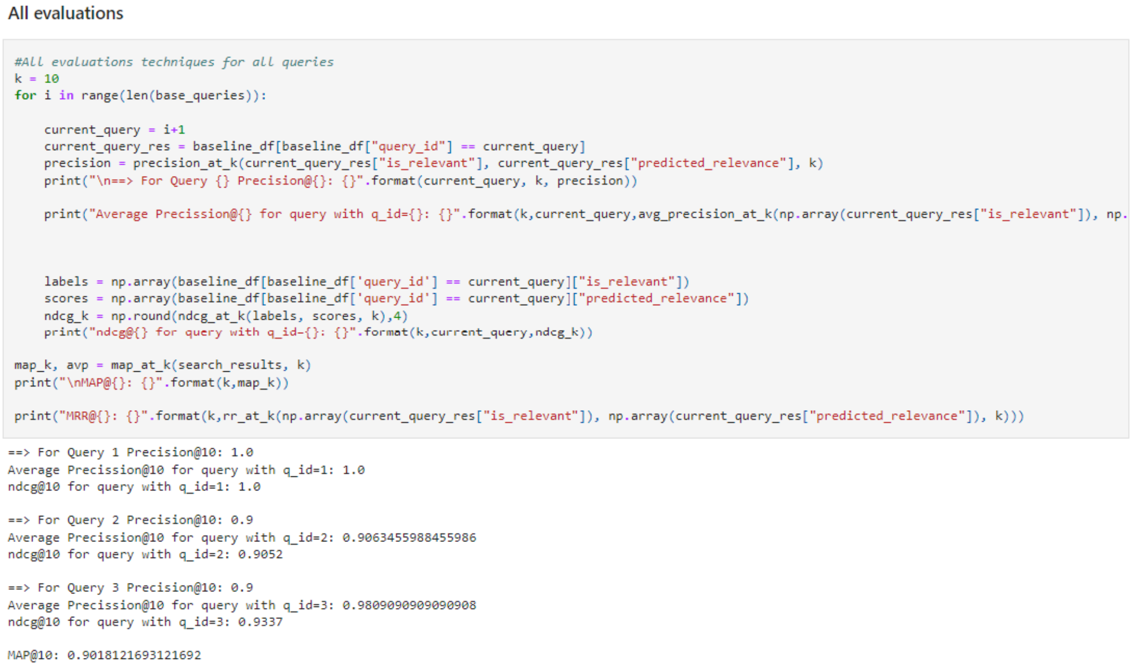
	query_id	doc_id	predicted_relevance	is_relevant
20	3	doc_3218	2.00628	0
21	3	doc_640	1.70515	0
22	3	doc_3845	1.70515	1
23	3	doc_3336	1.70515	1
24	3	doc_2537	1.70515	0

```
Query: florida
```

We see that the given ones, the query 1 and 3 are the bests. The chosen ones the highest precision queries are 1 and 5.
Then we do the same for the recall, which will be relevant for knowing how many relevants of the total have we evaluated. With highest recall in 1 and 3, and for ours 4 and 5. That will be interesting to do a P/R graph. The average precision and F1 scores are calculated the same way and 1 and 3 are the queries with bigger score and for the chosen by us the 5.

Then we do the average precision, giving us again the queries 1 and 3 the highest ones and from our queries the 5.

Finally we evaluate the Mean Average Precission and the Mean Reciprocal Rank:



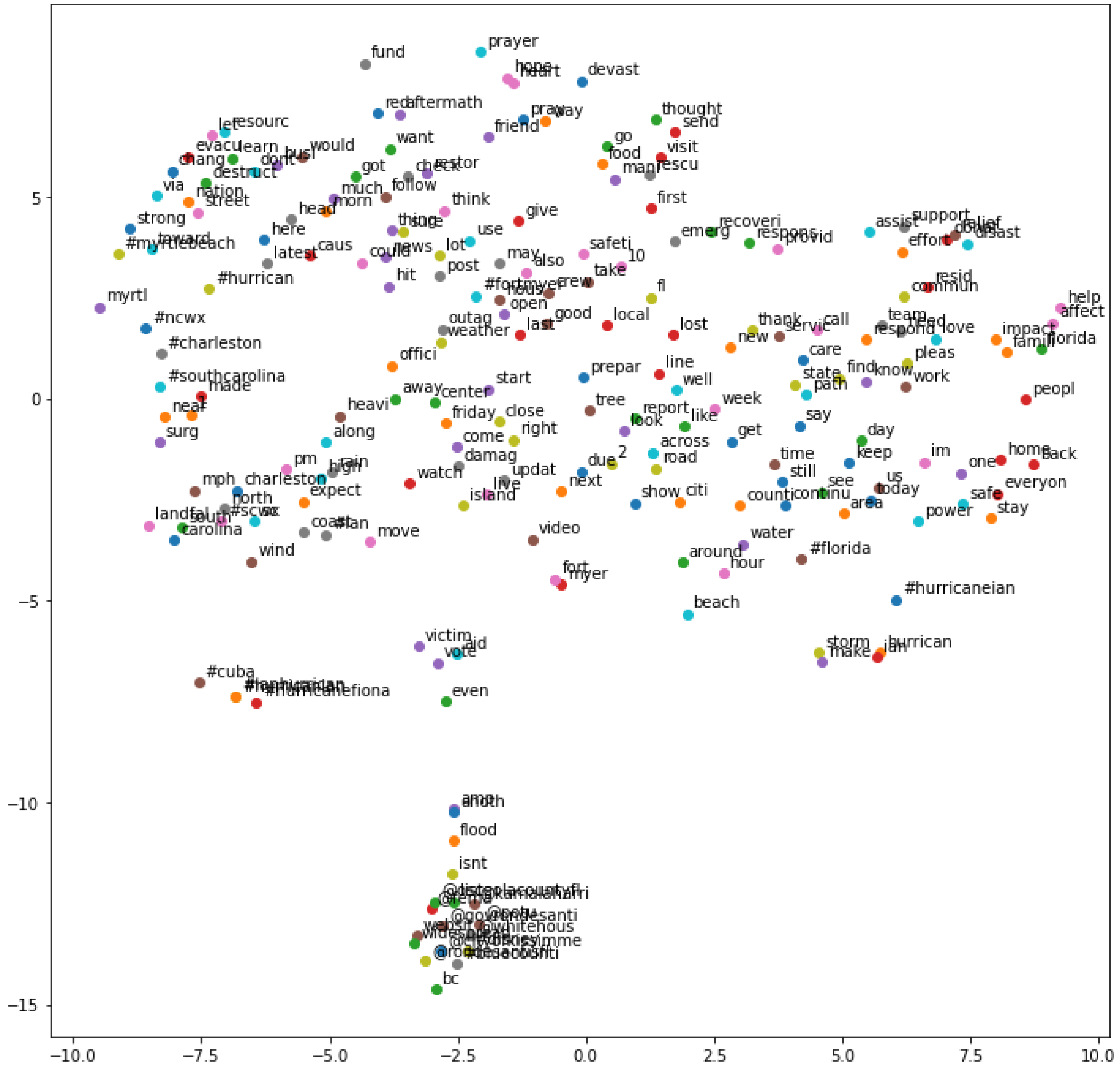
All evaluations

```
#All evaluations techniques for all queries
k = 10
for i in range(len(base_queries)):

    current_query = i+1
    current_query_res = baseline_df[baseline_df["query_id"] == current_query]
    precision = precision_at_k(current_query_res["is_relevant"], current_query_res["predicted_relevance"], k)
    print("\n==> For Query {} Precision@{}: {}".format(current_query, k, precision))

    print("Average Precission@{} for query with q_id={}: {}".format(k,current_query,avg_precision_at_k(np.array(current_query_res["is_relevant"]), np.



    labels = np.array(baseline_df[baseline_df['query_id'] == current_query]["is_relevant"])
    scores = np.array(baseline_df[baseline_df['query_id'] == current_query]["predicted_relevance"])
    ndcg_k = np.round(ndcg_at_k(labels, scores, k),4)
    print("ndcg@{} for query with q_id-{}: {}".format(k,current_query,ndcg_k))

map_k, avp = map_at_k(search_results, k)
print("\nMAP@{}: {}".format(k,map_k))

print("MRR@{}: {}".format(k,rr_at_k(np.array(current_query_res["is_relevant"]), np.array(current_query_res["predicted_relevance"]), k)))
```

```
==> For Query 1 Precision@10: 1.0
Average Precission@10 for query with q_id=1: 1.0
ndcg@10 for query with q_id=1: 1.0

==> For Query 2 Precision@10: 0.9
Average Precission@10 for query with q_id=2: 0.9063455988455986
ndcg@10 for query with q_id=2: 0.9052

==> For Query 3 Precision@10: 0.9
Average Precission@10 for query with q_id=3: 0.9809090909090908
ndcg@10 for query with q_id=3: 0.9337

MAP@10: 0.9018121693121692
```

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-words, this is the result:



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