# Information Retrieval

A personalized Search Engine for microblog content

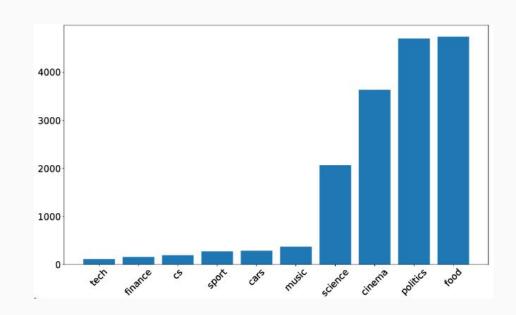
Matteo Angelo Costantini - 795125 Dario Gerosa - 793636 Michele Perrotta - 795152

### Goal

- Crawl tweets dealing with different topics using Twitter APIs
- Create a Search Engine
  - Must support both personalized and not personalized search
  - At least five user profiles with three topic of interest
  - Interest extracted from documents given by the user

## Crawler

- Python script using tweepy APIs wrapper
- Multiprocess approach to speed up the crawling process
- Online PostgreSQL database for persistence
- 10 topics



### Dataset

- ~ 36 000 000 tweet
- ~ 16 000 users
- Up to 3 200 tweet for each user
- 60 GB of data
- Tweets only in English
- JSON Format

Percentile	Follower
5	226
10	391
20	746
30	1,210
40	1,879
50	2,954
60	4,625
70	7,851
80	14,563
90	39,785
95	105,922
97	231,844
99	1,262,254
99.5	3,120,360
100	104,683,236

# Search Engine



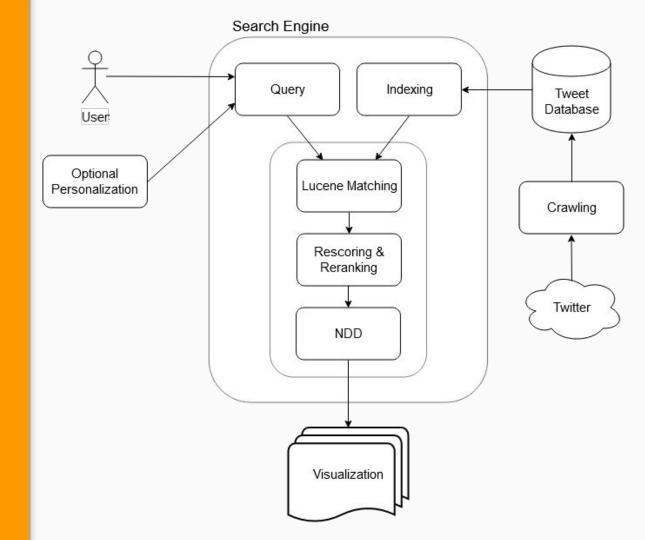
### Overview

The Search Engine uses Lucene **TF- IDF similarity** to index the documents crawled from Twitter.

Query **personalization** is achieved using *Query Expansion* and **bag-of-words** model.

The final output is the product of a **re-ranking** phase using a combination of the Lucene matching algorithms with other custom scores followed by a **Near Duplicate Detection** step.

A Web App is used to interact with the Search Engine and visualize the results



# Indexing

### Analyzer:

- Based on Lucene's Classic Analyzer
  - Preserves URLs, emails and numbers separated with hyphens
- Porter's Stemmer
- Removal of URLs and emails
- Stopword removal using Lucene default list
- Tokens normalized to lowercase

#### Indexer:

- Lucene's Classic Similarity
  - Refinement of the cosine-similarity based on TF-IDF
- Indexed Fields:
  - Text, Hashtags for text search
  - Date for range queries
  - Retweet count, author's data and other for scoring

# Queries

### Two main types of queries:

- Based on recent information
  Focused on new tweets
- Based on the whole information available in the dataset

#### Characteristics:

- Both queries support personalization based on user interests
- Re-Scoring based on Twitter's nature
- Near Duplicates Detection
- Range queries on Date

## Queries - 1

- Boolean Model retrieves relevant documents and chronological ordering of the results
- Selection of *n* most recent documents
- Scoring of the subset of documents using the Vector Space Model with Lucene's Classic
  Similarity and other Twitter-related factors such as:
  - Retweet rate
  - User's influence
  - Presence of URL

Nagmoti, Rinkesh, Ankur Teredesai, and Martine De Cock. "Ranking approaches for microblog search." *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 2010.

# Queries - 2

- Default Lucene's approach: Boolean Model followed by Vector Space Model to score the documents
- Select the n most relevant documents
- Re-score & Re-rank the subset using a linear combination of the Similarity Score and other Twitter-related factors such as:
  - Retweet rate
  - User's influence
  - Presence of URL

Nagmoti, Rinkesh, Ankur Teredesai, and Martine De Cock. "Ranking approaches for microblog search." *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 2010.

# Scoring

To Re-Score the documents we took into account the score given using the revised cosine similarity computed by Lucene but also other factors such as:

- The number of followers of the author of a tweet
- The number of retweets
- The length of the text
- The presence/absence of an URL
- The tweet being a quote/retweet or not

### Scoring - Equations

Base Lucene Score

$$bS = lw \cdot \frac{s_d}{max_{i \in R} \ s_i}$$

Follower Score

$$fS = fw \cdot \frac{ufi_d}{ufi_d + ufo_d}$$

Retweet Score

$$rS = rw \cdot \frac{r_d + fav_d}{max_{i \in R} \ r_i + fav_i}$$

Quote Score

$$qrS = q_d \cdot qw + r_d \cdot rw$$

Length Score

$$lS = lw \cdot \frac{l_d}{max_{i \in R} \ l_i}$$

URL Score

$$uS = uw \cdot u_d$$

Final Score: S = bS + fS + rS + qrS + lS + uS

### Personalization

#### Outline

- Document based personalization
- Bag-of-Words model for each topic
  - Topic dependent
- The bag is computed indexing the given documents and either using the whole set or using a subset of most informative (based on TF-IDF) terms
- Query expansion to include the terms defining the interests of the user

#### Profiles:

- Five users with different topics of interest
- At least three topics for each user
- At least 10 documents for each topic
- A customizable user using the web app

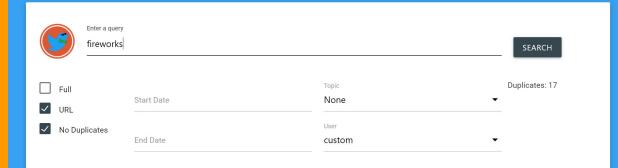
# Near Duplicate Detection

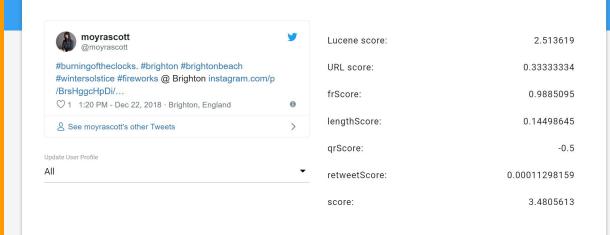
- We chose to use the Overlap coefficient since the Jaccard coefficient led to a lot more false negatives when one tweet overlaps with another
- The set used to compare the documents consists of the bi-grams extracted from the text of the tweet.
- Not much slower that Jaccard coefficient approach on a limited number of documents (tested with about 150 documents)
- Threshold at 80% of overlap

### Interface

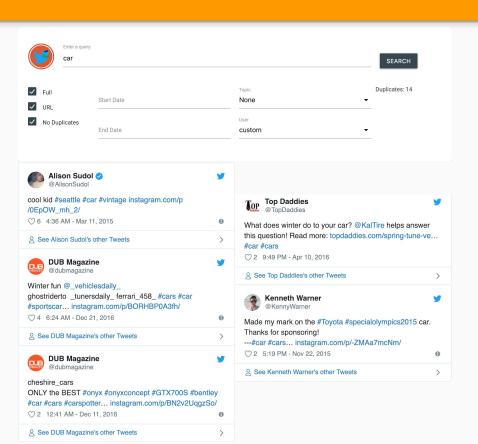
#### Supports:

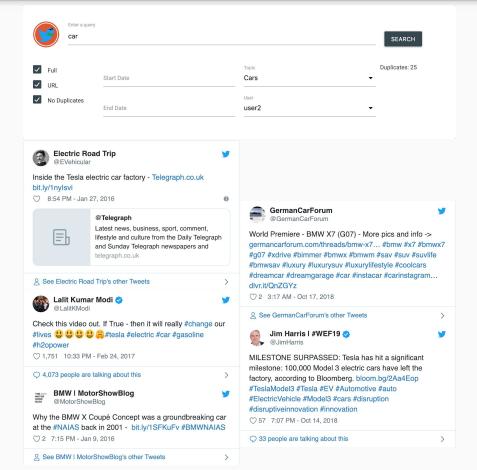
- Interactive tweets
- Personalization based on user profile and topic
- Range queries on the Date Fields
- Duplicates detection and filter
- URL scoring
- User profile creation



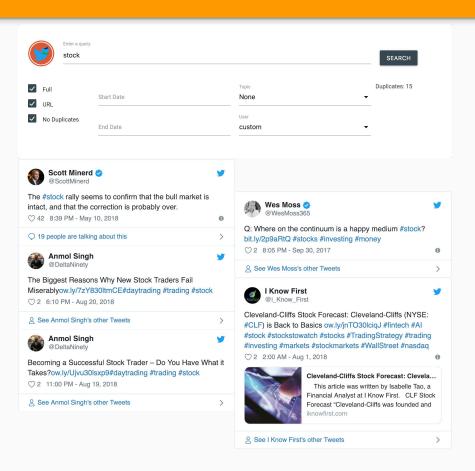


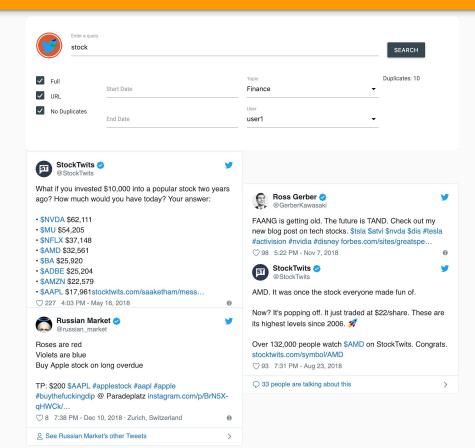
### **Query Personalization - Examples**



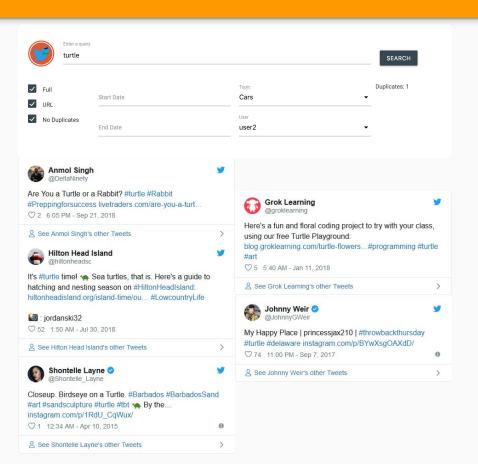


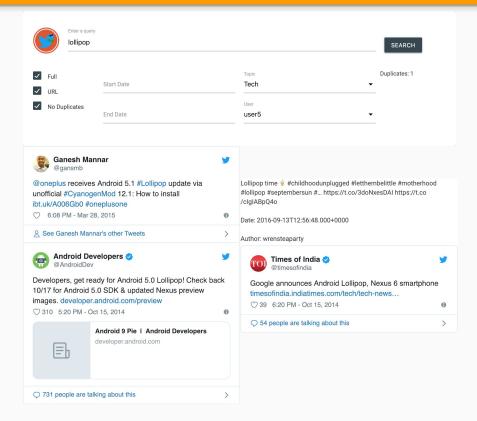
### **Query Personalization - Examples**





### **Query Personalization - Examples**





# Thanks for your attention

#### User 1:

cs: java programmer cinema: avengers superheroes morricone sergio leone tech: intel amd ryzen nvidia politics: trump intel top secret russia finance: nvidia amd apple microsoft stock

#### User 2:

tech: apple music: motorhead ace of spades metallica soad cars: tesla bmw

#### User 3:

sport: federer serena williamsscience: physics higgscars: audi volkswagen

#### User 4:

sport: golf francesco molinarimusic: madonna 50 centsfood: fruit pasta

#### User 5:

**tech**: lollipop android **food**: lollipop candy marshmallows **sport**: manchester united premier league chelsea arsenal