

CS6200 Information Retrieval

Final Project Report

Instagram Post Search

Xun Wang

NUID: 001386142

GitHub: <https://github.com/xw321/instagram-search>

Introduction:

Instagram is a popular online social network platform. The app allows users to upload media with captions and organized by hashtags and geographical/accounts tagging. With over one billion monthly active users and over 40 billion photos and videos shared on Instagram, the platform has become one of the most popular platforms for advertisers/influences to market their product or services.

Other than the basic photo/video uploading and editing feature, Instagram also provides a search feature to users. Currently, the search feature only offers users to search for Accounts, Tags or Places. When a user types something in the search bar, Instagram will prompt some possible accounts, locations, and hashtags for the user to choose. The user will have to choose one of the prompted results and then lead to that account, location, or hashtag page. However, these search results are dedicated to only one destination for users - one account page, one location, or one hashtag page. It doesn't offer a more general text search where a user can type something vague in the search bar, and get many relevant posts returned. Another issue Instagram has failed to offer is search for the post caption text, which can have very informational content.

In this report, an alternative approach will be discussed and an Instagram posts search engine with caption text prototype will be built as a result. This prototype will try to solve the search problem with Instagram from a different perspective.

Problem Identified:

When we try to search for some relevant posts on Instagram, it would be easier if we could input a general text query and get several related results. Not all users are equipped with deep

knowledge about the correct hashtags or accounts to search for. It would be disappointing when a user cannot find anything on Instagram because of not knowing certain tags or accounts.

Currently, Instagram's intention for users to search for many posts relevant to a topic is through searching with hashtags. However, searching with hashtags doesn't always provide relevant results due to hashtag spamming and misinterpretation of the hashtag meaning.

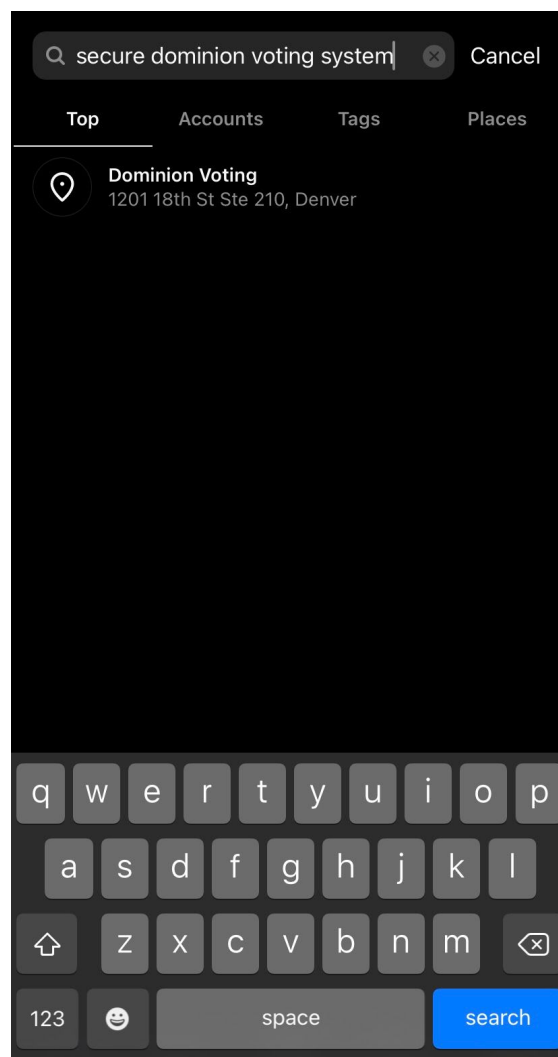
Here are some examples:

1. Trying to search for a hashtag but results are irrelevant because the tag can be interpreted other ways.



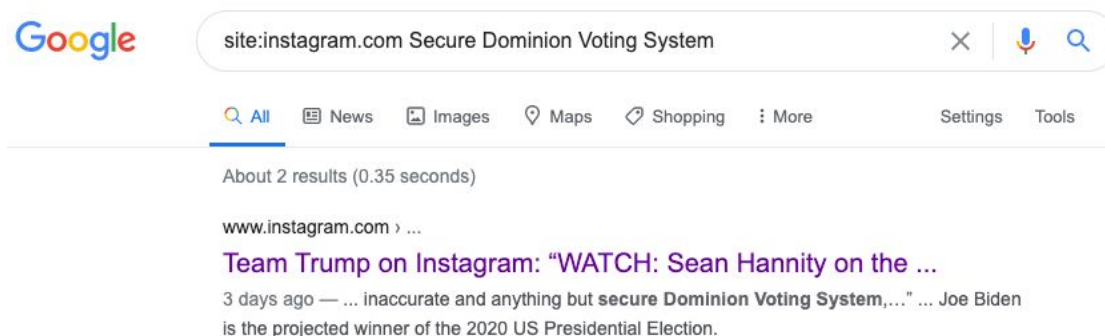
In this example, all returned results of #siliconvalley are tech companies based in Silicon Valley. However, the user wished to search for the TV show named Silicon Valley. The classification of posts based on the actual meaning of the tag can be difficult, and search by tags is not reliable. With a search with captions feature, it can provide more context and provide better results.

2. Trying to search text in the post caption but failed.

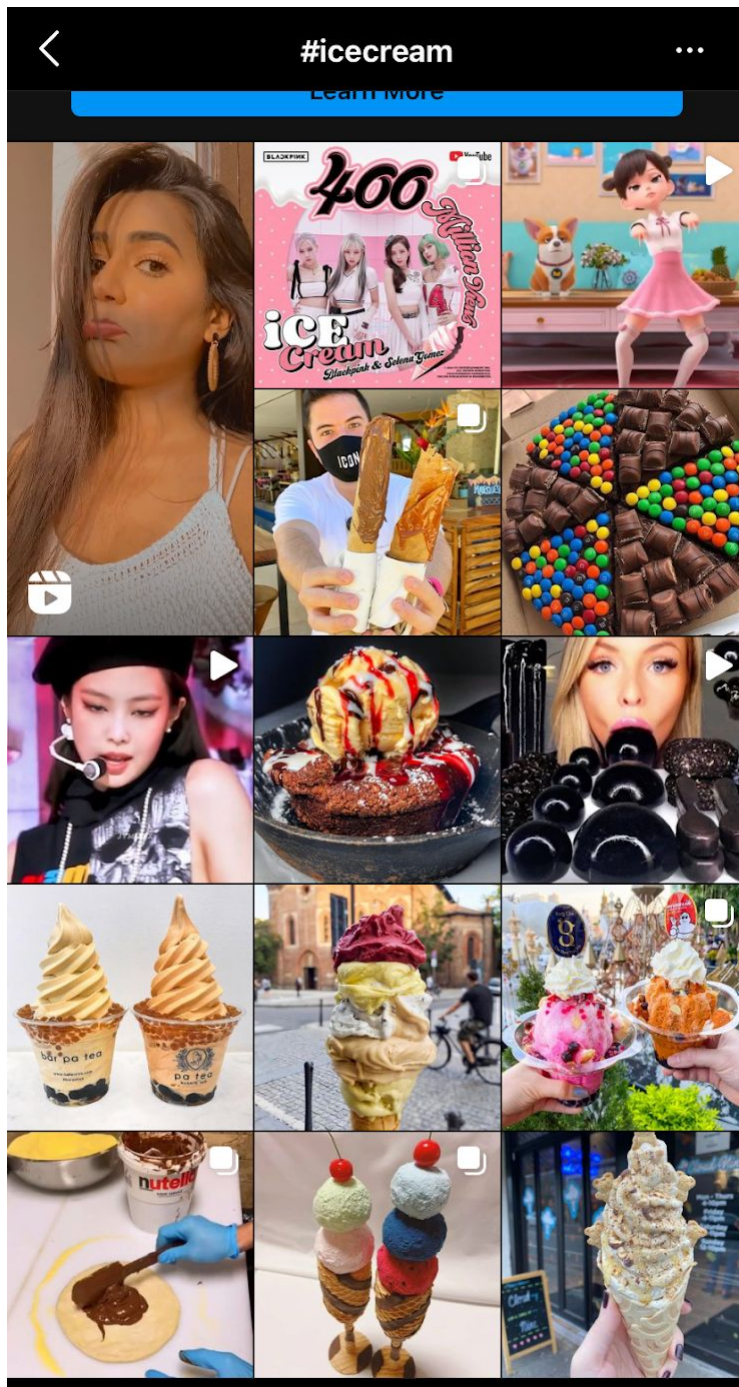


In this example, user Trump posted a video with captions related to “Secure Dominion Voting System” without other tags. A user trying to search such keywords would only get a location result which is not very helpful.

However, a site search on Google will easily return this post:



3. Trying to search with hashtags but failed due to hashtag spamming.



In this example, less than half of the top results of searching #icecream are relevant, and they were ranked lower than other irrelevant posts. This is because some of the irrelevant posts are also tagged with #icecream.

From the above examples, we can see that searching for hashtags is unreliable, and search for keywords/phrases would get nothing. It is surprising to learn that Instagram didn't provide a

search function with any caption information. This can be very irritating for new Instagram users and very inefficient for both the marketers/advertisers and users on this platform.

Solution Proposed:

The solution I proposed is to build a new and improved search engine for Instagram. The search engine would index words in post captions to provide a general text search for posts. The ranking results will mainly be based on the keywords occurrences in the caption, as well as post engagements(number of likes, comments, etc).

I will be using Elasticsearch to build the caption text index. I will also need to design a ranking function to properly balance the weights of positive query term matches and post engagements. The proposed search engine will have the post caption as the corpus. I will extract the caption text from the dataset and build an inverted index. The search results would return posts based on relevance of the query terms. Given the social nature of Instagram, popularity can be an important factor for posts. The ranking of these relevant posts will also be calculated based on the number of comments and likes. More comments and likes of a post, then more relevant the post can be. I will include these factors when designing the ranking function.

The success finished search engine should easily solve use case 2 mentioned above and improve other use cases.

Dataset:

Given the many restrictions of Instagram API, I used an Instagram Scraper to scrape 50 posts(if there are 50) of each user from the Top 1000 instagram influencers in US and Top 500 instagram influencers in UK. The reason I chose these two countries is to have most posts description/caption in English. The scraped data will be in JSON format.

After processing the raw data and excluding non-text captions, I had 78,738 Documents/Posts, with metadata information. Then I feed these documents to Elasticsearch to build the index for user queries. These documents took about 100 MB in the Elasticsearch cluster. A sample snippet of the scraped json file looks like the image below.


```

37 {
38   "id": "2442942789296427357",
39   "shortcode": "CHnE4YKnZFd",
40   "type": "GraphImage",
41   "is_video": false,
42   "dimension": {
43     "height": 1080,
44     "width": 1080
45   },
46   "display_url": "https://scontent-sjc3-1.cdninstagram.com/v/t51.2885-15/e35/125203614_404555480727148_6476145414737743322_n.jpg?_nc_ht=scontent-sjc3-1.cdninstagram.com&_nc_cat=1&_nc_ohc=JuFTFRcu48YAX-3YGyF&tp=18&oh=518e753c11b0a50f1f590c1128056f88&oe=5FDC59DF",
47   "thumbnail_src": "https://scontent-sjc3-1.cdninstagram.com/v/t51.2885-15/sh0.08/e35/s640x640/125203614_404555480727148_6476145414737743322_n.jpg?_nc_ht=scontent-sjc3-1.cdninstagram.com&_nc_cat=1&_nc_ohc=JuFTFRcu48YAX-3YGyF&_nc_tp=24&oh=b2c1f125e4898589141130b4ad1b4f&oe=5FDB42E5",
48   "owner": {
49     "id": "16278726",
50     "username": "bbcnews"
51   },
52   "description": "One of the creators of a coronavirus vaccine has said life should be back to normal by this time next year, if a high vaccination rate is achieved by autumn 2021. Prof Ugur Sahin, co-founder of BioNTech, thinks this winter would still be hard as the vaccine would not have a big impact on infection numbers. Last week, BioNTech and co-developers Pfizer said preliminary analysis showed their vaccine could prevent more than 90% of people from getting Covid-19. Tap the link in our bio to find out more. (📷 Getty Images)\n\n#Covid-19 #Coronavirus #Vaccine #bbcnews",
53   "comments": 1529,
54   "likes": 92008,
55   "comments_disabled": false,
56   "taken_at_timestamp": 1605441496,
57   "location": null,
58   "hashtags": [
59     "#Covid",
60     "#Coronavirus",
61     "#Vaccine",
62     "#bbcnews"
63   ],
64   "mentions": []
65 },

```

The JSON format is convenient to parse and index into Elasticsearch. In the back-end, I basically created a class to represent a Post object. By parsing and assigning values for each field, then I indexed them into Elasticsearch.

The entire scraping job was done in 3 hours with some trial and error, mainly due to Instagram trying to block frequent scraping work and requiring an active session ID to access Instagram website. The indexing to Elasticsearch took about 30 minutes.

Implementation:

The finished project has an interactive website using JavaScript and React framework as front-end, and a Java Spring Boot framework as back-end. It has a running Elasticsearch cluster in the background.

Back-end:

The back-end logic is written in Java. A Spring Boot project is built to handle front-end RESTful requests. Maven is also used to manage dependencies used in this project.

The first thing to do in the back-end is to process the crawled data and index it into Elasticsearch. I used the Gson library to parse the JSON files into a Post object, and index it into a Elasticsearch cluster.

After the indexing job is done, I implemented and built the search and rank function using Elasticsearch's Search API. By issuing the user query request using Elasticsearch Search API, the initial result set is created. The query is built using "matchQuery" with a minimum score of 0.7, which ensures we don't fetch irrelevant or low quality results. The maximum size of returned documents is set to 20, as I don't want to exceed Instagram's API rate limit. The "matchPhrase" type of query API can effectively retrieve exact phrase matches, but it is too restrictive and not suitable for general search use cases, so I stick with the "matchQuery" option for the search API.

When retrieved the initial results from Elasticsearch, I will run a function to tune their score by incorporating the number of likes and comments. Then the final results will be pushed into a max heap and later formatted as an array to send back to the front-end. The returned results has a "hits" field which contains information such as score, total matched documents, and a Map representation of the original document, which is convenient for calculating and re-ranking the results.

I used a weighted sum of likes and comments, and scaled it down to their score level, then added it to the raw score to get the final score for ranking. I also notice that some posts had their comments disabled, in this case I will just weigh the likes. The re-calculated score can affect the original ranking a little bit, but not much. This is because I still want to prioritize positive query matches over post engagements.

There are many options of the Elasticsearch APIs, I explored some of them and tried to incorporate them in, but they are either too restrictive or too loose for the retrieval. If given more time and more data, I would love to explore and test different options Elasticsearch can offer.


In testing and presentation, some queries will have less relevant results. This is largely due to the scale of the dataset. Many general topics are not included in the dataset as I tried to scrape popular posts in English. These celebrity/influencer posts may not be broad enough to cover the majority of topics on Instagram.

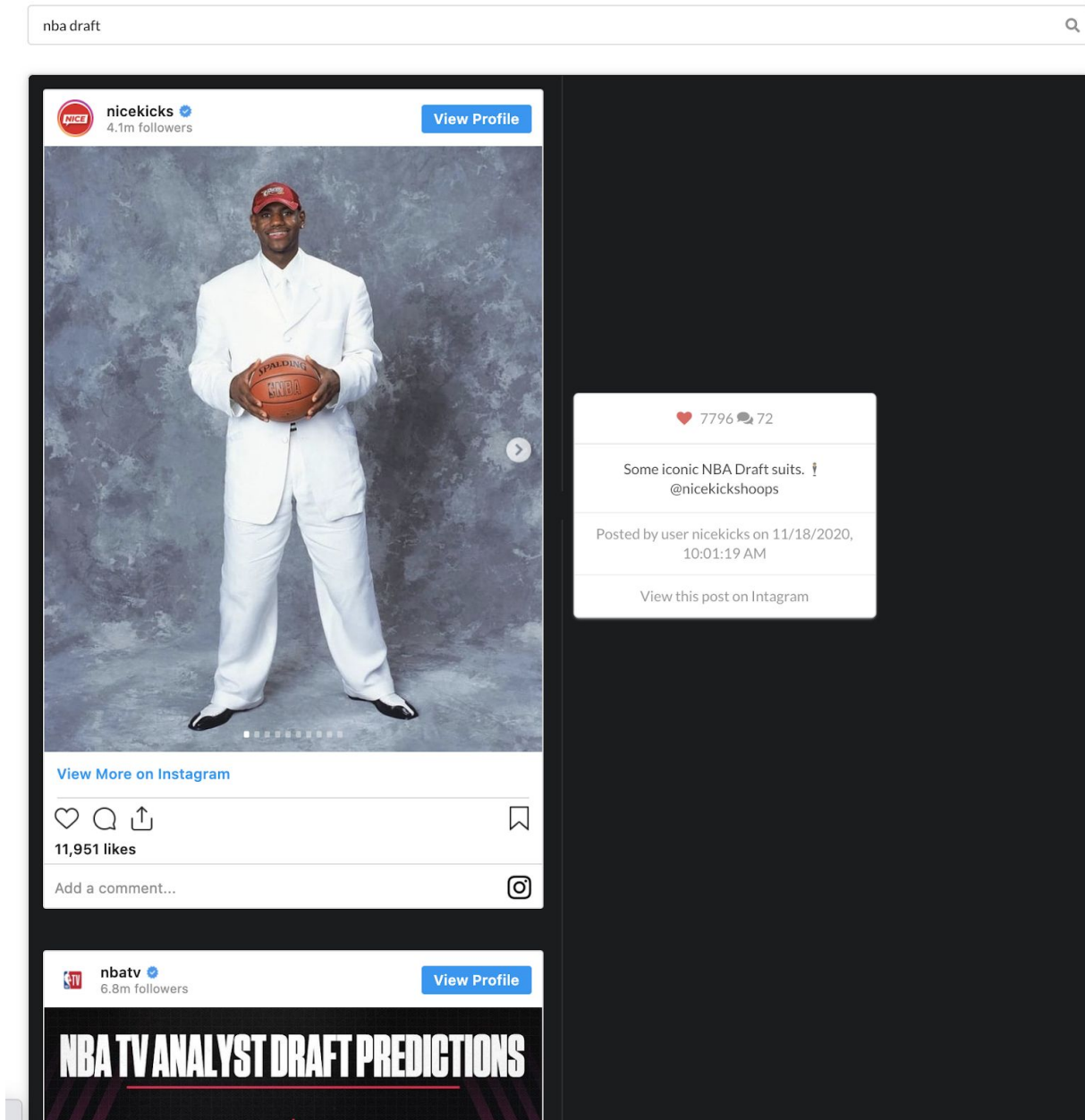
Front-end

The front-end UI is tuned using Semantic UI library. The website content is simple, a single page with a search box will be provided when landing the website. After inputting a query and

hitting Enter, relevant results will be returned in a list under the search box. Each element in the result list is an embedded Instagram post on the left using Instagram Embedded API, and caption text, number of likes and comments, and user information on the right. Each item of the results list will also have a link pointing to the original Instagram post, in case users would like to like, comment or interact with the post.

INSTAGRAM POST SEARCH



The logic behind this is quite straightforward. When a user hits Enter, the query text will be extracted and sent to the back-end using axios. When the response is returned, the front-end will parse and process the returned data (which are JSON Strings) and render them on the web page using the format mentioned above.

Note that when testing the search function, sometimes the post media is not shown in the results. This could happen because the crawled data is not up-to-date, and the owner of the post could archive/delete the post, or change their profile privacy settings to make it not visible to the public. Another reason can be the rate limit of the Instagram Embedded API, this happened on the presentation where all results media are not shown. There is not much I can

do as the website relies on the Instagram API. I have changed the default return size smaller to decrement the number of API calls.

Conclusion

This Instagram post search engine was built as a prototype to solve the complete searching functionality of Instagram. The proposed solution and implementation discussed in this report addressed and solved the problem to some extent. However, there are many future works and improvements to make for this project. For example, the keywords in the results can be highlighted. If we can have better control over Instagram APIs and data, we can build the inverted index in real time and update their ranking score on the fly. We could also incorporate text in comments as they can also provide some information.