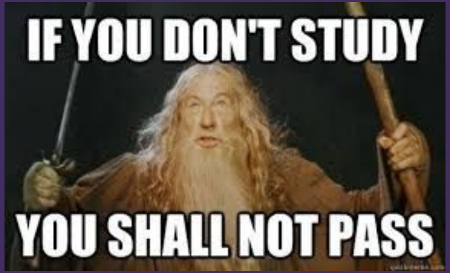
Efficient Meme Retrieval System

Aditya, Sanidhya, Vishal

Problem Statement.

A meme is an image, gif or video (usually funny) which has some text content on it.



Basic problem: when the user enters a search query, a set of memes is retrieved from a database, where each element of the database comprises of an image and some text.

Dataset & Collection

- We scraped more than 14,000 posts from 3 popular public Instagram meme profiles using an open source software, ‘instaloader’.
- Each post crawled resulted in a jpg and a corresponding json containing caption, hashtags, and other metadata of the post.

- To capture a vast diversity of memes, all content posted on the page since inception has been scraped.

Gold Standard Dataset

- Made a set of 10 queries on the 9gag website and picked 10 relevant results for each query.
- Created csv file with columns: Query, Captions, Hashtags
- Use this as Gold Standard to calculate Precision, Recall, F1, MAP score.

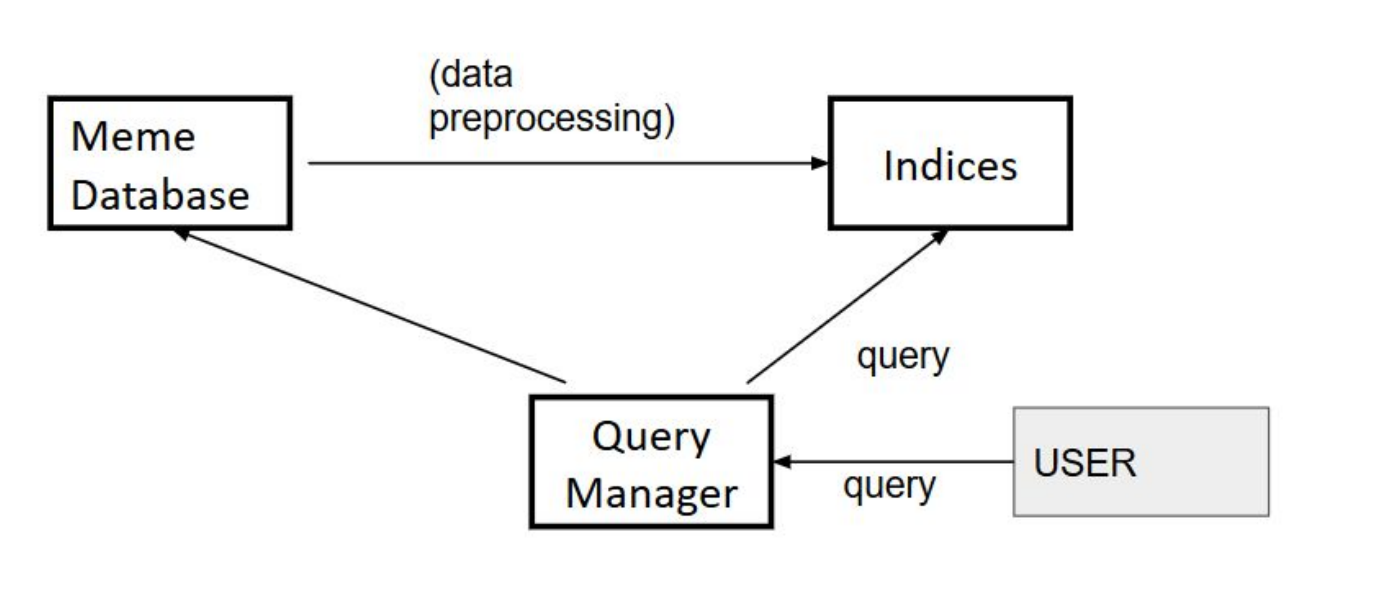
Challenges

- Absence of perfect correlation between caption and user query.
- Information contained in the meme image element itself is hard to retrieve.

OCR in Weighted Model

- To improve the results, we used Pytesseract to read the text from the meme images.
- We created one json for each image file containing this text, and then calculated tf-idf scores.
- We created a combined weighted model for the tf-idf scores of the actual text-based and the image-extracted text, by taking a weighted average of the tf-idf scores.

Search System



Methodology

- ✓ Dataset- Memes from popular public domains [~14K]
- ✓ Extract captions: Perform caption-based content retrieval
- ✓ Preprocess captions: stop-word removal, tokenize, stem (with Snowball Stemmer)
- ✓ Build index: Compute search index based on bag of words assumption with TF-IDF

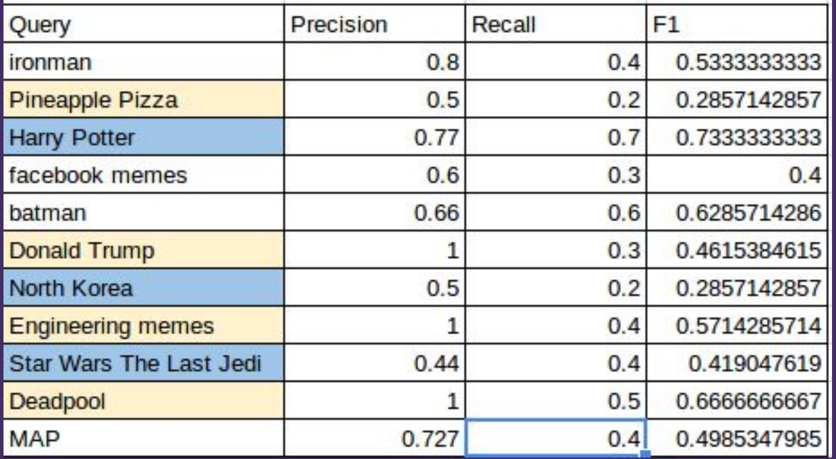
Methodology (contd.)

- ✓ Basic query handler: boolean query for starting off
- ✓ Better query handler which ranks results, using a vector space model (TF-IDF)
- ✓ Build intuitive GUI for easy deployment, which shows top 25 retrieved results in one screen currently

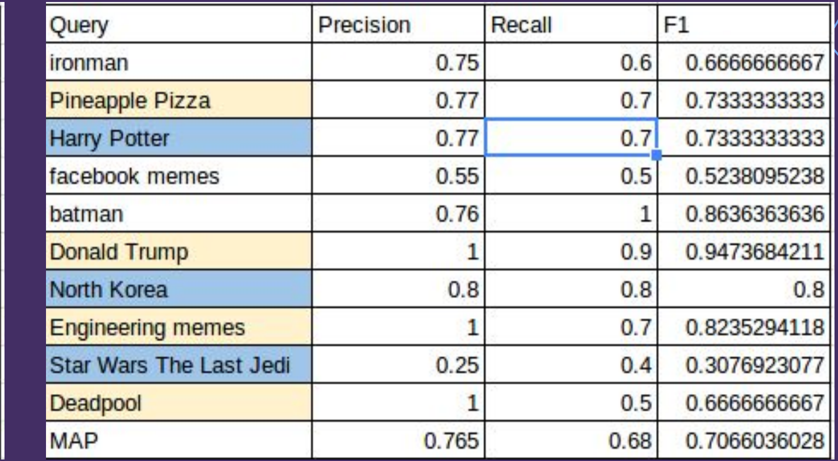
Methodology (contd.)

- ✓ Improved the relevance, extracted and utilized features from the images along with textual features.
- ✓ Prepared a Gold Set for evaluation
- ✓ Compared the results

Results



Query	Precision	Recall	F1
ironman	0.8	0.4	0.5333333333
Pineapple Pizza	0.5	0.2	0.2857142857
Harry Potter	0.77	0.7	0.7333333333
facebook memes	0.6	0.3	0.4
batman	0.66	0.6	0.6285714286
Donald Trump	1	0.3	0.4615384615
North Korea	0.5	0.2	0.2857142857
Engineering memes	1	0.4	0.5714285714
Star Wars The Last Jedi	0.44	0.4	0.419047619
Deadpool	1	0.5	0.6666666667
MAP	0.727	0.4	0.4985347985



Query	Precision	Recall	F1
ironman	0.75	0.6	0.6666666667
Pineapple Pizza	0.77	0.7	0.7333333333
Harry Potter	0.77	0.7	0.7333333333
facebook memes	0.55	0.5	0.5238095238
batman	0.76	1	0.8636363636
Donald Trump	1	0.9	0.9473684211
North Korea	0.8	0.8	0.8
Engineering memes	1	0.7	0.8235294118
Star Wars The Last Jedi	0.25	0.4	0.3076923077
Deadpool	1	0.5	0.6666666667
MAP	0.765	0.68	0.7066036028

Gold Set: Baseline results

Gold Set: Improved Model results

Results and Conclusion

Our improved model performed significantly better in terms of recall and marginally better in terms of Precision than the model without using OCR. Hence, we can conclude that combining image-based features with the text-based features had an appreciable improvement on our IR system.

Thank You