Content Based Friend Recommendation System

- KIRUBA DHAYALAN
- MAYANK REDDY SADDI

Problem Overview

- Online Social Networking platforms are a great place to make new connections and meet people of similar tastes.
- The growing popularity of such social networks has raised new areas of applications for recommender systems.
- The main aim of such recommender systems is to identify and suggest new friends for a user such that they have similar interests.
- The most popular methods utilized for making recommendations exploit the social network information or the similarity of user generated content.

Problem Overview

- The aim of this project is to build such a recommender system which profiles the user and generates the topics of interest of the user.
- Then compare his topics of interest with that of other users in the network whom the user is not friends with.
- Based on similarity of the topic distribution users are ranked and recommended to the main user.

Methodology

- Data Collection
- Data Preprocessing
- Model Selection and Training
- Results
- Model Evaluation

Data Collection

- The source for all the data collected for the project is TwitterAPI by using tweepy library for python.
- The initial aim of the project was to allow recommendations from all users who are at n hops from the main user (n and main user being a configurable parameter).
- However during the Data Collection phase we realized that we have grossly underestimated the amount of data that needed to be collected.
- This humungous task was further complicated by the strict Rate Limits which Twitter applies on TwitterAPI.

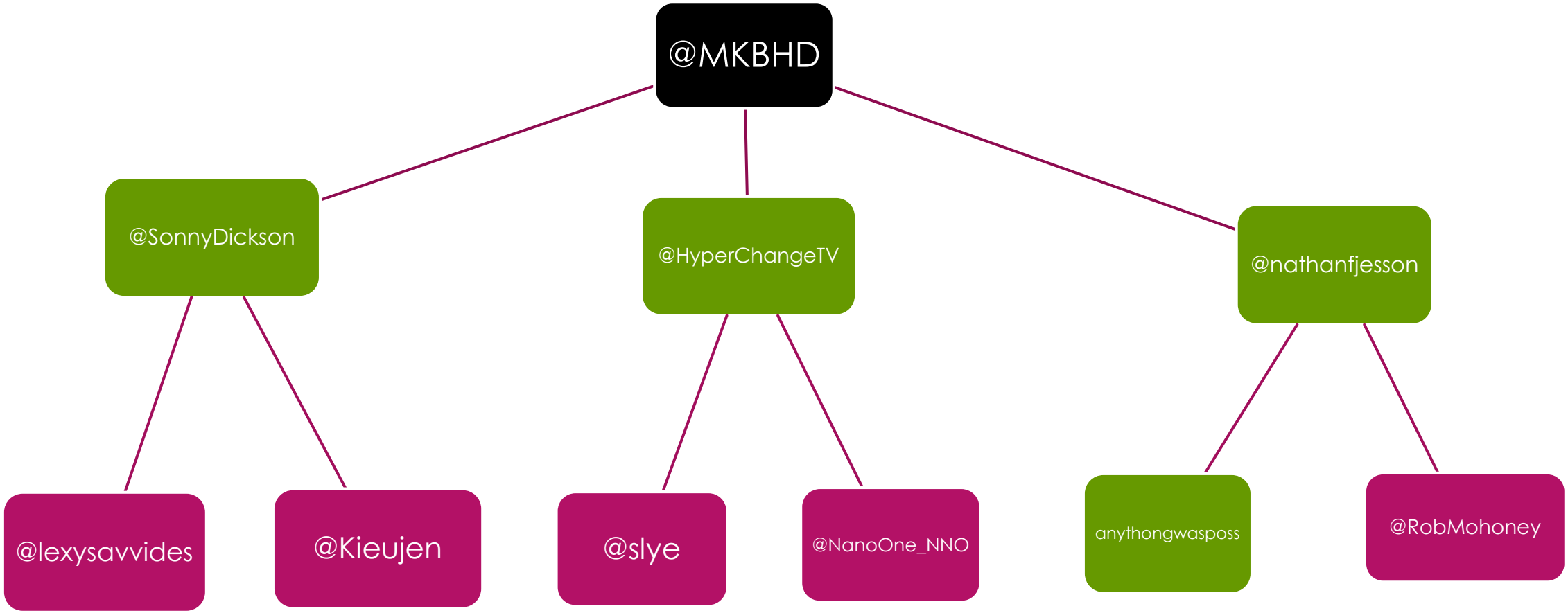
Data Collection

- Thus we decided to choose a fixed main user (Marques Brownlee @MKBHD, a popular tech Youtuber) and provide recommendations to him.
- In order to build our recommender system we collected all the tweets of the main user @MKBHD.
- Then we collect 400 most recent 2nd degree users and their corresponding tweets.
- Only 400 have been collected for rate limiting purposes.

Main User

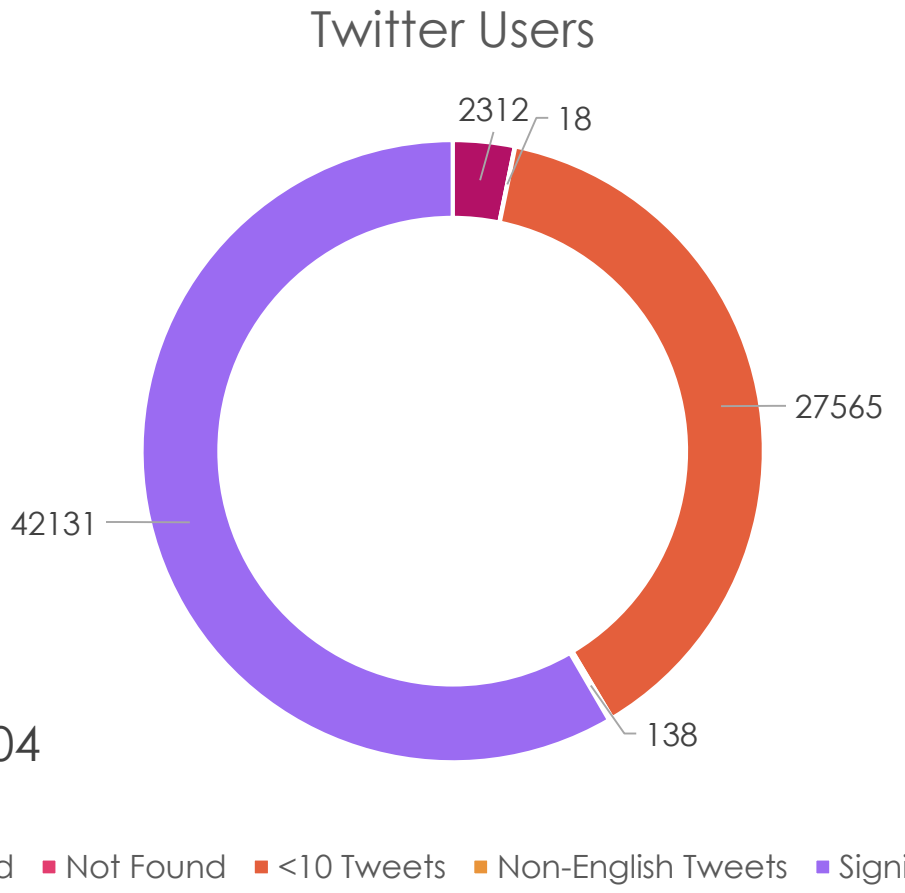
Friends

Not Following 2nd Degree users of our Interest



Data Collection Stats

- Total number of friends of @MKBHD : 342
- Total number of 2nd degree users : 72,164
  - Protected Users : 2312
  - Not found : 18
  - Users with < 10 tweets : 27,565
  - Users with non-English tweets: 138
  - Users with significant workable tweets : 42131
- Number of tweets collected for main user: 3204



Data Preprocessing

- Tweets data obtained is often very noisy with many acronyms, slang, non-English characters, emojis, urls, etc.
- Before we train any model, we clean the tweets data by:
  - Detect and eliminate users with non-English tweets.
  - Eliminating any url links in tweets.
  - Separating hashtags and mentions from the text in a tweet.
  - Using Unicode to detect emojis and eliminate them.
  - Discarding any special characters or words with characters less than 3.
  - Lemmatizing the text and then tokenizing it.
  - Identify parts of speech and keep only nouns and verbs.

Data Preprocessing

- Once we have all the text tokenized, we put together all the tokens obtained from the tweets of each user as a single corresponding document.
- Then a dictionary is constructed over all these documents using Gensim library's Dictionary.
- Using the tokens and the dictionary constructed we then generate a Bag of Words corpus using doc2bow.
- We now save our dictionary and the corpus for ease of use during model building and training.

Model Selection and Training

- We can formulate finding the most similar candidates and suggesting friends as a document matching and ranking problem.
- Where tokens of each user is a separate document and we match it with our main user.
- For this, there are 2 main approaches which can be used:
  - 1. TF*IDF model which matches using the term frequency structures for each document.
  - 2. Topic Modelling (LDA): This approach predicts k topics being discussed in each document and match documents with similar topics.

Model Selection and Training: TF*IDF

- In this method we convert the documents into vector by making use of our previously generated bag of words corpus and constructing a TF-IDF vector for each document.
- We then create similar vector for the main user using all his 3204 tweets.
- Once we have the vectors we compute Cosine similarity for all 2nd degree users and the main user.
- Then sort the users based on the similarity score obtained.

Model Selection and Training: LDA

- Latent Dirichlet Alloction (LDA) is an unsupervised generative model which assigns topic distributions to documents.
- LDA assumes that each document can be explained by a distribution of topics and each topic can be explained by a distribution of words.
- Once we specify the number of topics it assumes that the document is explained only by those number of topics and generates two variables:
  - 1. A distribution over topics for each document.
  - 2. A distribution over words for each topics.

Model Selection and Training: LDA

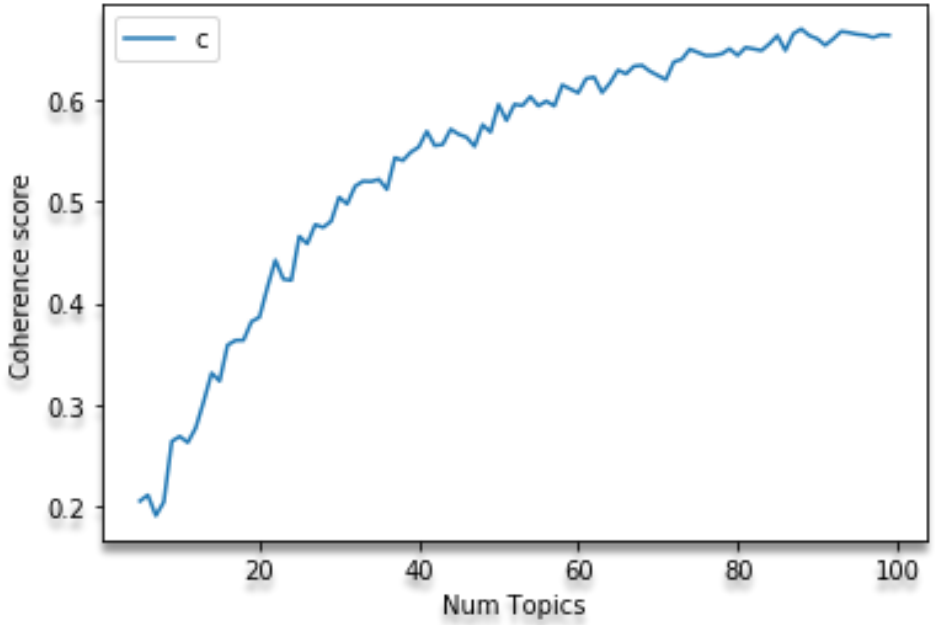
- We can use the distribution over topics for each document to compare and retrieve similar documents.
- For comparing the documents we take the Jensen-Shannon distance between them.
- Jensen-Shannon distance is a method of measuring similarity between two probability distributions.
- Using Jensen-Shannon we can get which documents are statistically “closer” (and therefore more similar), by comparing the divergence of their distributions.

Model Selection and Training: LDA

- With our model basics done we can train our LDA model on the data. However LDA needs that we provide the number of topics in a document a-priori.
- Selection of number of topics is very important for the resulting topic distributions.
- We measure the quality of a Topic distribution using ‘coherence’.
- Topic coherence measures each topic by scoring it based on calculating the degree of semantic similarity between words in the topic. It is often considered as a metric to evaluate the quality of a topic

Model Selection and Training: LDA

- In order to obtain a quality topic distribution we can perform coherence analysis.
- We take topics ranging from 5 to 100 and analyze the coherence scores for them.
- We can see that scores flatline after 90 topics so 90 would give a good quality topic distribution.



Model Selection and Training: LDA

- Now we train the LDA model for 90 topics on our data .
- Once trained we can predict the topic distributions for other users tweets by passing the bag of words corpus for it to the model.
- We then compare the topic distributions of the main user and 2nd degree users using Jensen-Shannon distance and sort them.
- This gives us the most similar users based on their tweets.

Results

- TF*IDF

```
Most SImilar users are as follows:
Name                        score

'1. SteveLevitan     0.15528728067874908'

'2. bradybuzz     0.15149781107902527'

'3. CamAndCompany     0.15093570947647095'

'4. LoneSuspect     0.14409850537776947'

'5. maenadsnest     0.14183053374290466'

'6. tolleycasting     0.13477978110313416'

'7. LuciaFranceSays     0.127369225025177'

'8. GabbyGiffords     0.1244729682803154'

'9. shannonrwatts     0.12183833122253418'

'10. geekylonglegs     0.11530221998691559'
```

Results

- LDA

Topics obtained can be seen to have converged well with all related terms in one topic.

```
lda.show_topic(topicid=64, topn=20)
```

```
[('car', 0.088930525),
 ('race', 0.042851936),
 ('porsche', 0.022248613),
 ('lamborghini', 0.019692246),
 ('bmw', 0.017071104),
 ('driver', 0.015411248),
 ('ford', 0.014030993),
 ('drive', 0.013471677),
 ('new', 0.013082672),
 ('suv', 0.012773525),
 ('motor', 0.011486006),
 ('vehicle', 0.010402274),
 ('racing', 0.010303565),
 ('track', 0.009814409),
 ('road', 0.009607816),
 ('wheel', 0.00953608),
 ('engine', 0.009273619),
 ('ferrari', 0.008969438),
 ('auto', 0.0089062415),
 ('sport', 0.008764437)]
```

```
lda.show_topic(topicid=1, topn=20)
```

```
[('car', 0.039846744),
 ('china', 0.03732198),
 ('flight', 0.0330994),
 ('tesla', 0.030119138),
 ('model', 0.026335724),
 ('trip', 0.025633821),
 ('travel', 0.024099456),
 ('vehicle', 0.02114598),
 ('japan', 0.020112662),
 ('driver', 0.015841782),
 ('tokyo', 0.01463205),
 ('airport', 0.014080499),
 ('electric', 0.0131783625),
 ('uber', 0.011972773),
 ('nissan', 0.011518922),
 ('korea', 0.011330447),
 ('road', 0.010881123),
 ('aston', 0.010262737),
 ('production', 0.009332807),
 ('airline', 0.009161385)]
```

```
ctionary,corpus,lda = train_lda(df_short)
```

```
me to train LDA model on  16269 articles:
```

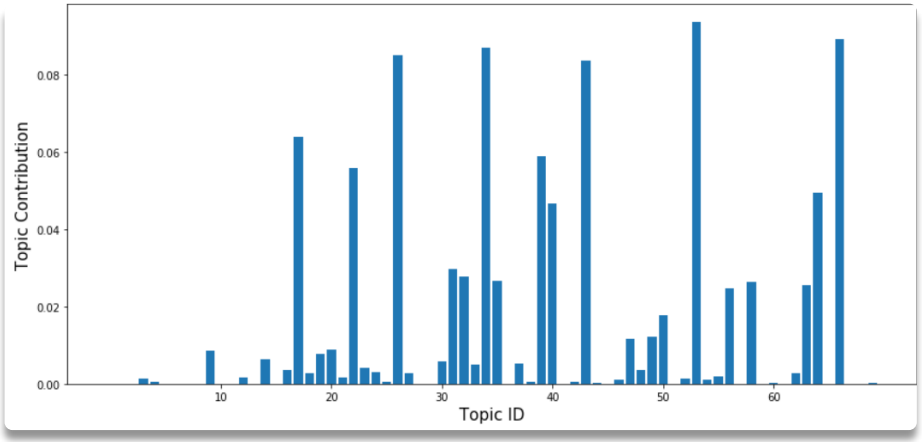
```
a.show_topic(topicid=4, topn=20)
```

```
'game', 0.12863763),
'stream', 0.041356992),
'play', 0.014230518),
'gaming', 0.013466893),
'xbox', 0.013353333),
'twitch', 0.013163195),
'player', 0.012194321),
'update', 0.0104355365),
'world', 0.010284799),
'pokemon', 0.010253668),
'ps4', 0.009159109),
'today', 0.008711291),
'nintendo', 0.007949925),
'streamer', 0.007242893),
'steam', 0.0071263677),
'games', 0.007113024),
'time', 0.0070354105),
'fortnite', 0.006866427),
'community', 0.006722191),
'mode', 0.0066952403)]
```

Results

- LDA

Topics distribution of main user



Results

- LDA Predictions

mentions	hashtags	tokenized_text	handles	Similarity_scores
[[krishaamer], [ooliganpress], [wearehawthorne...	[[elearning, transmedia], [], [ai, artificiali...	[kinovan, focus, transmedia, storytelling, tha...	Kinovan_C	0.830906
[[], [], [], [], [], [], [], [], [], [], [], [...	[[], [], [], [], [], [], [], [], [], [], [], [...	[person, person, person, person, person, perso...	victorycini	0.828316
[[dick_in_milk, burstousobbing], [dick_in_milk...	[[], [], [], [], [], [], [], [], [], [], [], [...	[там, уже, вторая, вторая, еще, лучше, первой,...	kisimiaka_ururu	0.825571
[[TribLIVE, ForbesLife], [MarshProductio1, hag...	[[horsepower, musclecar], [carscoops, autoshri...	[thank, thank, thank, thank, thank, exoticar, ...	ExoticarsPgh	0.823557
[[], [], [], [], [], [], [], [], [], [], [], [...	[[], [FollowBack, TeamFollowBack], [], [TeamFo...	[background, page, guy, tip, follower, followe...	FocusedMark	0.823547
...	...	...	...	...
[[KuzcoLeDesma, UltimateTigers], [], [KuzcoLeD...	[[Ultimate], [], [], [], [], [], [], [], [], [...	[today, ultimate, frisbee, tomorrow, morning, ...	UltimateTigers	0.760467
[[thespunta], [], [], [ADampSandwich, MrNiceGu...	[[], [], [], [], [], [], [], [], [], [], [], [...	[human, water, update, mei, routine, add, bell...	RayApollo	0.760442
[[], [Asmar3313], [], [Asmar3313], [], [Asmar3...	[[], [], [checkra1n], [], [], [], [], [che...	[coupon, code, discount, app, device, apps, cy...	hacx	0.760338
[[warriorsvox, 957thegame], [epaschall, spidad...	[[Warriors], [Warriors, Jazz], [IAGLTY, Warrio...	[story, friendship, tonight, arc, childhood, f...	LaurenceScott	0.760233
[[sharkcat13], [TonyCelloTweets], [NotStephenB...	[[], [], [], [], [], [], [], [], [], [], [], [...	[sorry, thank, thank, canada, interest, team, ...	bublywater	0.760233

Evaluation

- To evaluate the models we add the friends of the main users to our data and predict the same number of suggestions.
- Using the TF*IDF method we added 342 and predicted 342 users of which True Positives were 27.
- Using the Topic Modelling method the True Positives were 84 ~ precision of 24.56%

Future Work

- It can be seen that using Topic modelling methods provide friends suggestions but does not perform too well. However, if more access to data is obtained the model is bound to perform much better.
- Also the scores obtained can be used in tandem with topological link prediction methods to obtain better predictions.

Thank You