Content based Friend recommendation System

Kiruba Dhayalan, Mayank Saddi

ABSTRACT:

In this project we aim to design a system to suggest user with most similar members who are within n hops from the user.To do this we mine for user's personality traits using his tweets/retweets/likes data.Once we have the user's personality information we check for traits for the members who are under n hops from the user.Then cluster based on each personality trait and rank users with highest likelihood of link using link prediction methods. Once we have ranked users for each trait we can aggregate the values and obtain top x users who are similar to user in all aspects and recommend them.

INTRODUCTION:

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. In this project we try 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 the similarity of the topic distribution, users are ranked and recommended to the main user.

Network topology methods use the structure of the subgraph around the user. It uses properties of the network like the number of common neighbors between the users and ranks users based on their similarity**.** Using just the topology of the network may not be enough metric to determine if a user connects with another. So in this project, we consider the user's attributes as well to make such recommendations for better results. As people in the real world do not just connect with one another based on how many common neighbors each have but they become friends with people based on how similar they are. In order to do this, we mine for the user's personality traits using his tweets/retweets data to obtain areas of the user's interest.

BACKGROUND/RELATED WORK:

[1] is about friend recommendation system based on user similarity and trust degree.The authors in [1] considers user's behaviour on Sina micro-blog to find their interest. They build micro-blog topic model based on users' operations and the concept of time slices. Then they calculated the user similarity based on topic probability distribution that we get through the topic model. After

that, they cluster the users to get social circles.They recalculated the user similarity based on circle structure and calculated user's trust degree of other users.
[2] is built a friend recommendation system based on Friend-Space. Friend-Space is developed using K-Means, Apriori, Ranking and Recommendation algorithms are developed for Friend-Space Application. They also mentioned that Friend-space Cluster based application gives better performance in case of execution time for K-Means algorithm.The authors [3] propose a link prediction method based on user actions with the post which includes clicks, shares, likes, forwards and comments.This link prediction model combines user action metrics and topological structure metrics. In our project we have considered the users tweets along with network topology for friend recommendation.
The authors of [4] has extended FriendTNS that takes into account the degrees of the nodes, and the direct links between them by calculating the strength of the tie between two users into account. [5] tried to build a news recommendation system based on the user's Twitter account. From twitter profile building is done by considering the tweets, re-tweets, and hashtags. They also used simple cosine similarity measure, to compare the differences among the user profiles, and also among the recommended news lists, in order to check the discriminative power of the proposed method. The authors of [6] proposed a system that makes use of latent features, extracted with a diversity of dimensionality reduction methods, to infer the personality of Twitter users using textual content-based features, and they compared the performance of the different techniques. In [7] instead of using textual tweets produced by users they examined the accounts that our users follow, and use them to determine the high-level interests of these users, then they use these areas of interest as features for predicting perceived personal traits.

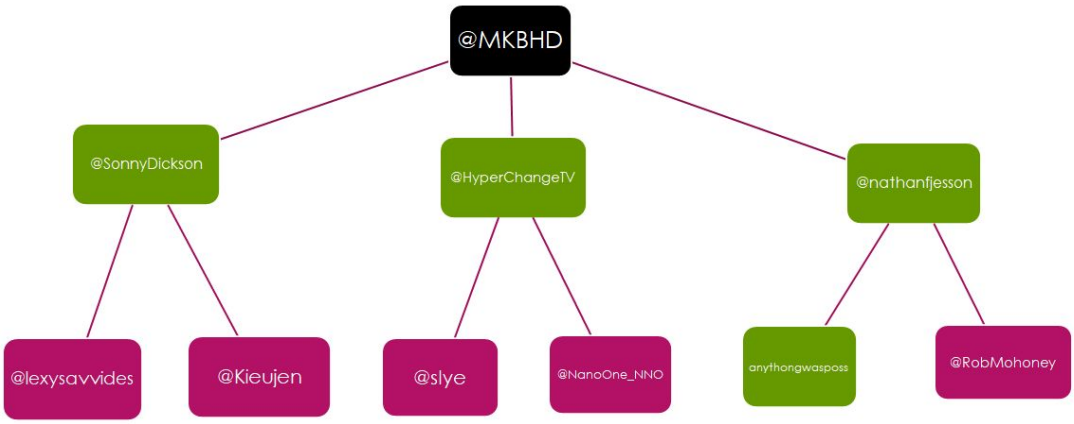
APPROACH:

DATA COLLECTION

Twitter streaming Application Programming Interface provides a streaming API to stream real-time data and tweepy library for python. In order to access the twitter API, we should create a developer account that provides the access token which is used in authenticating the API call.

As mentioned in the scope, we are considering fixed main users and only 200 most recent users 2nd-degree users. Below is a graph that explains the 2nd-degree user (not a friend of the main user but a friend of main user's friend)

Not Following 2nd Degree users of our Interest

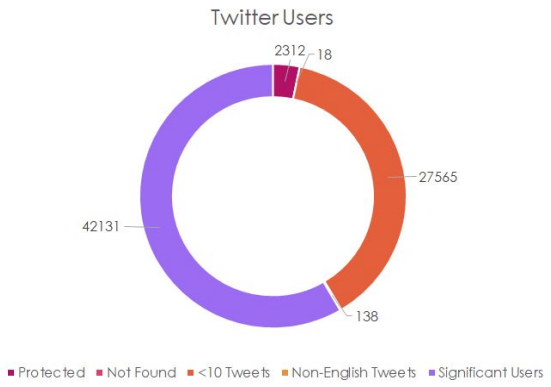


Statistics:

- ➢ Total number of friends of @MKBHD : 342
- ➢ Total number of 2nd-degree users: 72,164
  - ❑ Protected Users : 2312
  - ❑ Not Found: 18
  - ❑ User with < 10 tweets : 27,565
  - ❑ Users with non-English tweets: 138

After removing protected, not found, <10 tweets, non-English twitter users, the significant workable tweets is 42131.

Number of tweets for the main user is 3204 tweets.



DATA PREPROCESSING:

Tweets data obtained is often very noisy with many acronyms, slang, non-English characters, emojis, URLs, etc. Data preprocessing was the most difficult than expected and time-consuming task as this involved many stages and needed a lot of fundamental cleaning of the data that is collected was necessary before analysis of data. Before we train any model we have to clean the tweet 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.

NLTK library has been used for lemmatization and tokenization. Spacy has been used to identify parts of speech. Textbob has been used for eliminating non-English words. Regular expressions are used to identify hashtags, mentions, and emojis and eliminate them.

Once we have all the text tokenized, we put together all the tokens obtained from the tweets of a single user as a single corresponding document. Then a dictionary is constructed over all the documents using the Gensim library's Dictionary. Using the tokens and the dictionary constructed we then generate a Bag of Words Corpus using doc2bow. We save this dictionary and corpus for ease use in the 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 the main user.
Two approaches can be used for this:

- **TF*ID model** which matches using the term frequency structures for each document.
- **Topic Modelling (LDA)**: This approach predicts k topics being discussed in each document and match documents with similar topics.

TF*ID Model:

In this model, we convert the documents into a vector by making use of our previously generated Bag of Words Corpus and 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 the cosine

similarity for all 2nd-degree users and the main user. Now we will sort the users based on the similarity score obtained.

Topic Modelling (LDA):

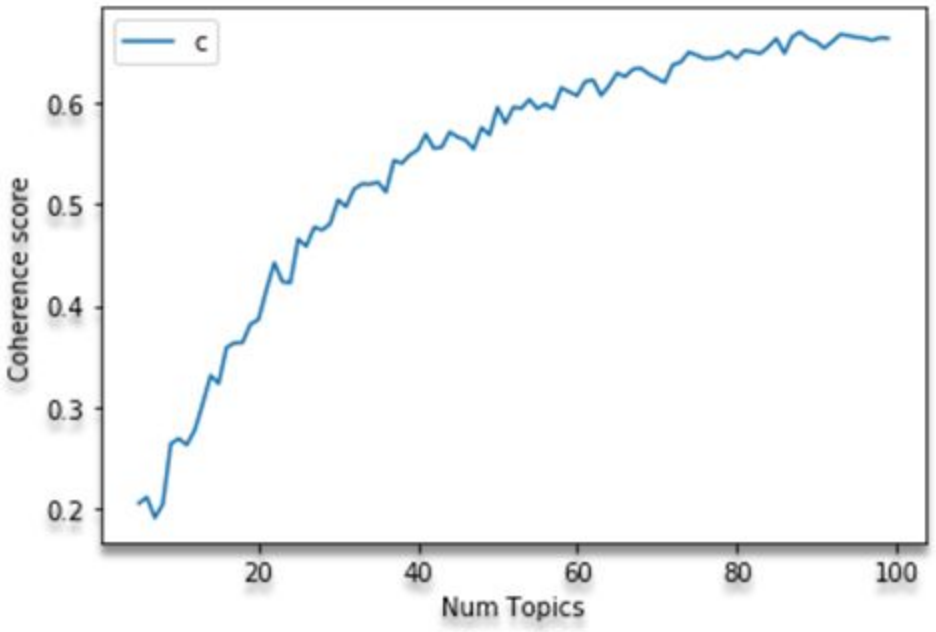
Latent Dirichlet Allocation (LDA) is an unsupervised generative model that assigns topic distributions to documents. LDA assumes that each document can be explained by the distribution of topics and each topic can be explained by the distribution of words. Once we specify the number of topics it assumes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

- A distribution over topics for each document
- A distribution over words for each topic

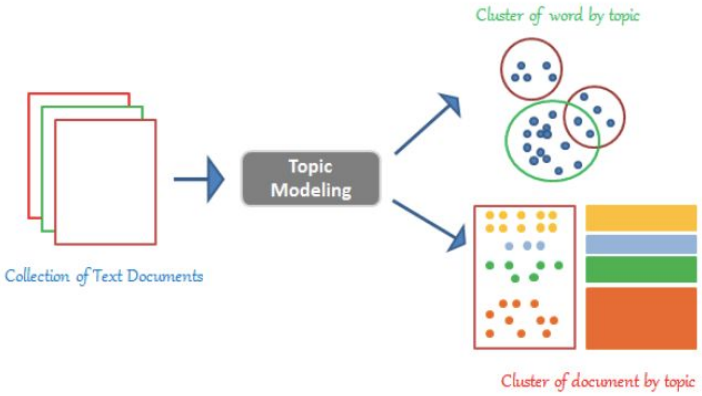
We can use the distribution over topics for each document to compare and retrieve similar documents.Cosine similarity works well for comparison between two vectors but LDA gives us a distribution so we choose Jensen-Shannon. This metric is used to calculate the distance of two distributions .Using Jensen-Shannon we can get which documents are statistically "closer" (and therefore more similar), by comparing the divergence of their distributions. For comparing the documents we choose to take Jensen-Shannon distance between them.

After data preprocess, we can train our LDA model on the data. However, we should provide the number of topics in a document a-priori to LDA. The 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. In order to obtain a quality topic distribution, we can perform a coherence analysis. We take topics ranging from 5 to 100 and analyze the coherence scores for them.



From the coherence score vs Number of topics graph, we can see that scores flatline after 90 topics so 90 would give a good quality topic distribution. Now we train the LDA model for 90 topics on our data. Once trained we can predict the topic distributions for other user's tweets by passing the corpus 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.



Overview of the LDA model.

EXPERIMENTAL RESULTS:

TF*-ID:

The top 10 similar 2nd degree users recommended for the main user by TF*-ID score.

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

'1. CiaranBlu     0.5227833986282349'

'2. ridetheferry     0.14518731832504272'

'3. MKUYM     0.10184516012668 61'

'4. PhotographyAF     0.08935341238975525'

'5. HiddenHoboken     0.07322193682193756'

'6. lajollamom     0.07268744707107544'

'7. Tinklabs     0.071419358253479'

'8. coolmomtech     0.07071090489625931'

'9. CAParksNow     0.0689782053232193'

'10. LangeSoehne     0.06803765147924423'

'11. 1Hotels     0.06619332730770111'

'12. DanStrumpf     0.06396479159593582'

'13. evokadagency     0.06323044002056122'

'14. Marketplace     0.062070079147815704'

'15. Carnage4Life     0.06180575489997864'
```

LDA:

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

In topic id 64, we see that the topics are about cars and in topic id 4, the topics are about games, which proves the correctness of the LDA model.

```
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)]
```

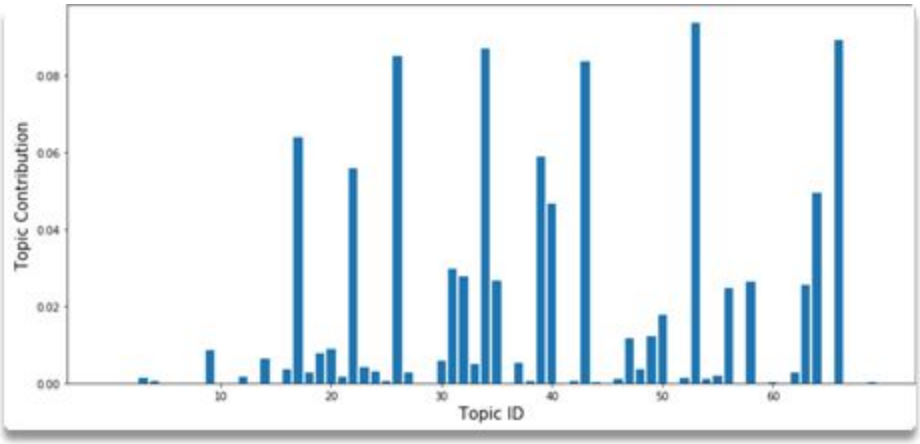
```
ctionary,corpus,lda = train_lda(df_short)

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)]
```

Topics distribution of the main user:



LDA Predictions:

	tokens	score
EvernoteStatus	[system, system, time, service, release, servi...	0.822119
LRUltimate	[field, pickup, tonight, week, pickup, field10...	0.800098
ExoticarsPgh	[thank, thank, thank, thank, thank, exoticar, ...	0.799151
newsdiva	[thank, blog, blog, thank, blog, blog, thank, ...	0.799132
stas_satori	[техник, напрягся, общем, сняли, новый, выпуск...	0.796908
...	...	...
estherschindler	[people, conversation, topic, care, reminder, ...	0.411424
mlnestel	[head, speaker, rock, concert, amherst, ornith...	0.409545
dagrantla	[help, picture, love, rome, pasta, ferraris, f...	0.408406
randypaulino	[importance, self, reward, thief, game, credit...	0.407693
nicie_panetta	[thread, moment, unity, andy, goldworthy, fans...	0.404924

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 5.

Using the Topic Modelling method the True Positives were 15 ~ precision of 4.61%

True positive and the precision value are low, this may be because of fewer data. Since LDA has correctly classified topics, with an increase in data, this model's performance will increase.

CONCLUSION:

- ➢ It can be seen that using Topic modeling 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 strong predictions.
- ➢ Sentiment Analysis can be done on the LDA, which will give better predictions.
- ➢ Rate limit restriction of Twitter API has to be considered and have to allocate enough time for data collection.

REFERENCES:

1. Wang, J., Gao, S., Wang, L., & Yu, Z. (2018). Micro-Blog Friend-Recommendation Based on Topic Analysis and Circle Found. 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService) . doi: 10.1109/bigdataservice.2018.00033
2. Tasgave, P., & Dani, A. (2015). Friend-space: Cluster-based users similar post friend recommendation technique in social networks. 2015 International Conference on Information Processing (ICIP) . doi: 10.1109/infop.2015.7489465
3. . Srilatha, P., & Manjula, R. (2016). User behavior based link prediction in online social networks. 2016 International Conference on Inventive Computation Technologies (ICICT) . doi: 10.1109/inventive.2016.7823266
4. Ahmed, C., & Elkorany, A. (2015). Enhancing Link Prediction in Twitter using Semantic User Attributes. Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM 15 . doi: 10.1145/2808797.2810056
5. Lee, W.-J., Oh, K.-J., Lim, C.-G., & Choi, H.-J. (2014). User profile extraction from Twitter for personalized news recommendation. 16th International Conference on Advanced Communication Technology. doi: 10.1109/icact.2014.6779068
6. Moreno, D. R. J., Gomez, J. C., Almanza-Ojeda, D.-L., & Ibarra-Manzano, M.-A. (2019).Prediction of Personality Traits in Twitter Users with Latent Features. 2019 International Conference on Electronics, Communications and Computers (CONIELECOMP) . doi:10.1109/conielecomp.2019.8673242
7. Volkova, S., Bachrach, Y., & Durme, B. V. (2016). Mining User Interests to Predict Perceived Psycho-Demographic Traits on Twitter. 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService) . doi: 10.1109/bigdataservice.2016.28