International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 04, April 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

TWITTER FOLLOWEE RECOMMENDER SYSTEM

Jitali Patel

Computer science Department, Nirma University, (India)

ABSTRACT

With the dawn of Internet Age, we have more and more social network popping up and billions of people accessing and exchanging information on numerous platforms ranging from Facebook to Reddit to Twitter. Twitter one such major platform provides real-time micro blogging. Information is shared between followers and followees network in which every follower receives the tweet that is tweeted by his followee. These users can be categorized into information seekers, information sources and friends/colleagues. Current work targets information seekers and help them find the perfect information sources for them to follow.

Recommender systems have been used in almost all such sites and this work aims at not only implementing the same but going one step ahead, using the meta information in form of follower-followee topology and twitter profiles of users to identify potential candidates and rank them accordingly. This allows the recommendations to be extremely personal and can revolutionize how information seekers access Twitter. This experiment uses real data from Twitter and has the nature to adapt to the user requirements.

I. INTRODUCTION

1.1 Background

A platform build to develop social relations among people who have real life connections or share some hobbies, activities or interests is called a social networking platform. Popularity of social networking platforms rose over the last few years because how easy it is to access online sites and the benefits it brings along. Currently Facebook, LinkedIn and Twitter can be considered three major forms of social networks and also the ones most popular.

People engage in social networking for various reasons, the main ones being: to be in contact with relatives, family and friends, to share information, to find information, to keep up with what others are doing.

Social networks are used not only to maintain friendships but also to promote products online, run awareness campaign, political promotions, hold contests, polls and anything that needs a large number of consensus to be involved. With the increasing availability of internet across platforms even desktop/laptop is not a constraint for a person to access these social networking sites.

With the large numbers comes a variety of people on social networking sites, all such people might not appeal to every person who uses the site. Hence most of the sites allow creation of friendship or such so that information shared only much these people are shown to the user. Recommender Systems plays a major role here, it would be tedious if not impossible for a user to manually find people whose interests align with his/her. Recommender System helps the user build his/her network, discover people across the globe who share interests with him/her and saves time.

Twitter is one such major platform, allows users to tweet up to 140 characters and hence known as a micro blogging service.

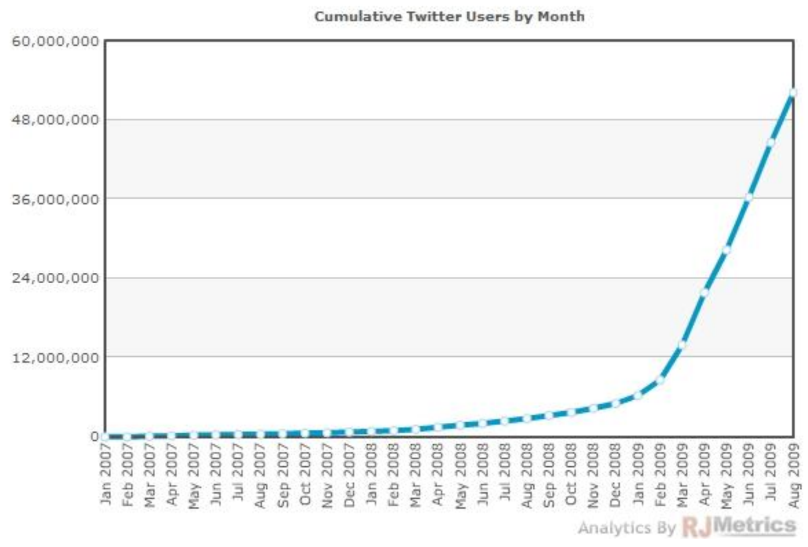


Figure 1 Growth of Twitter Users

Created in March 2006, Twitter has never looked back and has currently 500 million users and has reached the status of Top 10 searched websites. Comparing it with traditional blogging, Twitter is more about being concise and to the point, fast real time information sharing. A person in Twitter may follow a second person, but it is always not necessary that second person follows back the first one, this brings a new dynamic to the "Friendship" word thrown around in the social networking domain. We have two classes, *followers* who follow the given user and *followees* who the given user follows.

1.2 Problem Statement

Success of a social network strongly depends upon its recommender system, following the right people gives the user access to the exact information he/she expects and needs and achieves a certain level of satisfaction while on the other end every incorrect or inappropriate suggestions can cause distress while surfing the website because the irrelevant information or the redundant information brings down the whole social network experience of that website for the user.

Statistics show that only 22.2% of the Twitter relations are two ended. Thus leading us to a number of 77.8% of Twitter connections are one sided, also 67.6% users are not followed by anyone proves very clearly that majority of users use Twitter as a source of information rather than a social networking site.

Recommender Systems are mainly using one of the following categories: similarity, influence, content filtering and collaborative filtering[4]. Some may use combination of two more of the above mentioned technique but the problem arises when you have to deal with millions of people and creating a single recommendation technique which appeals to all of them is not possible.

We need a personalized recommendations which may follow some standard backbone structure but has to do macro and micro level adjustments with changing users and observing the patterns the user follow to further mold the algorithm[3].

1.3 Objectives

The two main objectives of the work are, identifying potential candidates for recommendation and ranking them according to target user giving him personalized recommendations.

There will be multiple strategies employed to rank, using both the semantic network and twitter profiles along with many machine learning techniques. Finally we will be adding up all such weighted scores and rank the people and give the user such list from which he/she can easily follow people who share interests with him/her and expand his network for a better social networking experience.

II. RELATED WORK

2.1 Cosine Similarity and Vector Space Model

Cosine Similarity calculates similarity between two vectors, vector space model is mainly used to represent a sentence, or a set of sentences or a document in term of a vector of words. Each new word will add a unique dimension to the vector with its weight being the magnitude. This will highlight how important or relevant that word is in the context of the given document.

$$\text{Cosine Similarity}(A,B) = \cos(\theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Equation 1

Cosine Similarity=1 means the documents are the same and 0 means they have no correlation.

2.2 Page Rank

Google first coined the word PageRank and was used to rank the results of their search engine. It is used assign weights to nodes in a network and the weights are propagated through edges.

Higher the weight of a node in a graph more is its influence over the network.

$$PR(A) = \frac{(1-d)}{N} + d \sum_{i=1}^{N} \frac{PR(T_i)}{C(T_i)}$$

Equation 2

Webpage A has links T(1..n) incoming, where d is the damping factor generally set between 0 and 1. Even if a page is not linked no any website at somepoint in the time if a user visits that website through the domain name directly it will be assigned some page rank.

$$P(i,j) = \frac{1-d}{N} + \frac{d * linkCounts(i,j)}{outDegree(i)}$$

Equation 3

Page ranks can also be calculated using a probability matrix, which creates a N cross N matrix, the link count represents the links going from i to j, and the out degree represents links going out from i, and d is the damping factor as already discussed above.

2.3 Generic Recommendation Algorithms

The three main categories include:

- Collaborative Filtering
- Content Based Filtering
- Hybrid Filtering

Collaborative Filtering is the technique used most commonly, as the name suggest it is not dependant on content but recommends using what other people think about the content. With the rise of social network there has been a lot of opinionated text on the internet about everything ranging from movies to food etc. So this technique utilizes all this opinions and recommends accordingly. It suffers from “cold star” a phenomenon that means that without already existing data set of opinions one cannot start recommendation[4].
Content Based Filtering is a technique that uses the content to find similar items to the ones that the user already likes and recommends accordingly
To avoid problems of collaborative filtering, hybrid approaches can be used which takes the positive parts of both the algorithms.

III. CANDIDATE SEARCH-RANKING FACTORS

3.1 Candidate Search

Users followed by the followers of target user followees are the possible candidates for recommendation.

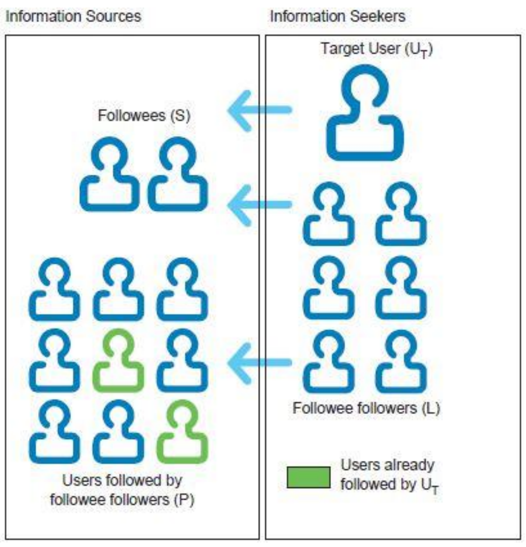


Figure 2

```
Pseudo Code:
Ut   = input
ids = get_followees(ut)
For each id in ids:
        ids2[get_followers(id)]+=1
For each id in ids2:
        ids3[get_followees(id)]+=1
        ids3[followees]*=ids2(id)
Sort(ids3)
Return top 10 values
```

Each element in this set is a possible candidate for recommendation, this set is used in this work for all the experiments, same will be used for ranking alternatives for finalizing recommendations. There can and will be repetitions and will be used to weigh the influence of users over the network

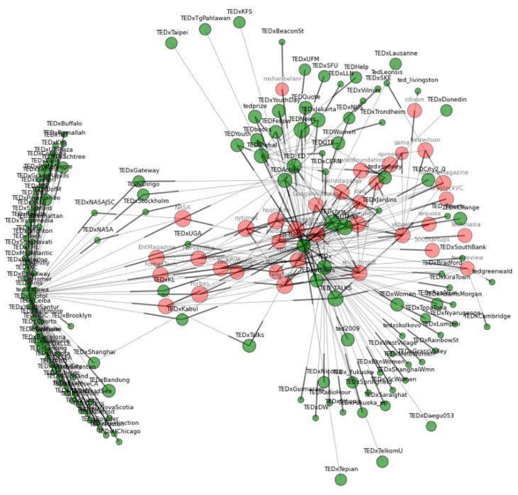


Figure 3

3.2 Similarity Strategy

Description Content Similarity, the profile description generally contains keywords which best describe the user, where he works, what he likes, his hobbies, school etc. It can be an extremely good factor in recommendation but it is not reliable because most of the information seekers we are targeting don't fill up the necessary profile and still expect results of high quality.

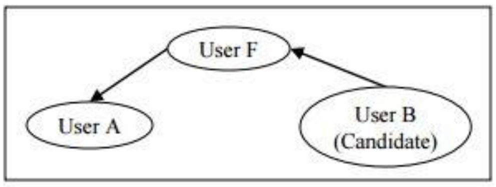
Tweet Content Similarity, target users are profiled on the basis of the people he/she is following, because it is expected that the target user is an information seeker and doesn't post many tweets, while on the other hand Candidate Users are profiled on the basis of his/her tweets because he obviously is an information source and must have its own content. Cosine similarity is calculated between the two documents by combining all tweets and forming a term vector[1].

Hash Tag Similarity, hashtags are considered important keywords in the tweet and hence are given extra weightage while defining similarity. Same as above we will be defining hash tag similarity as cosine similarity between two vectors consisting of collection of respective hash tags used.

3.3 Followee of Followee strategy

Recommends users are who are followed by many of user's followees, mainly recommends users with whom the user is friend with in real life[2].

$$FeFe(u_i) = \sum_{\forall v_j \in Fe(u), v_j \neq u_i} Followee(u_i, v_j)$$



Equation 4

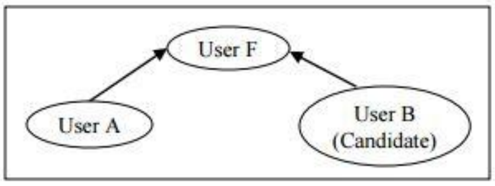
Figure 4

If A is followed by B1,B2…BN and if there is some C who is followed by a large proportion of B's then we can confidently say that C might be providing something that A is interested in. Also there will is a lot of possible redundancy, which we can exploit to get better ranks. We can just increase weights for the users that are coming through multiple Bs

3.3 Followee of Follower strategy

Recommends users are who are following many of user's followees, mainly recommends users with who have common interests and connections.

$$FeFr(u_i, u) = \sum_{\forall v_j \in Fr(u), v_j \neq u_i} Followee(u_i, v_j)$$



Equation 5

Figure 5

If A is following B1,B2…BN and if there is some C who is following a large proportion of B's then we can confidently say that C might be providing something that A is interested in. The user may be his friend in person or may share some place/interests. This strategy is totally based on the semantic topology of the followee-follower structure.

3.4 Follower of Followee strategy

Again a strategy purely based on the topology of follower-followee network. Recommends users are who are followed by people who follow the target user.

$$FrFe(u_i, u) = \sum_{\forall v_j \in Fe(u), v_j \neq u_i} Follower(u_i, v_j)$$

Equation 6

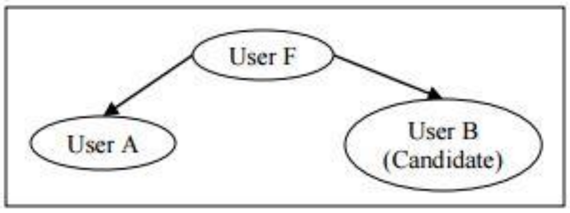


Figure 6

If A is followed by B1,B2…BN and if there is some C who is also followed a large proportion of B's then we can confidently say that C might be providing something that A is interested in. They may or may not share same interests or may not know each other but surely have some similarity.

3.5 Follower of Follower strategy

Exploits the topology to find some incomplete loops of following in the network and helps complete them. Recommends users are who are followed by people who are followed by the target user.

$$FrFr(u_i, u) = \sum_{\forall v_j \in Fr(u), v_j \neq u_i} Follower(u_i, v_j)$$

Equation 7

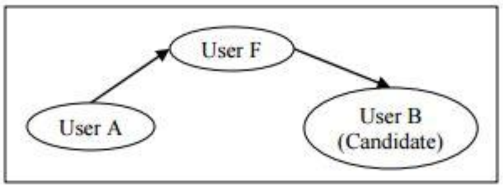


Figure 7

If A is following B1,B2…BN and if there is some C who is also followed a large proportion of B's then we can confidently say that C might be providing something that A is interested in.

3.6 Using Page Ranks

Page Ranks can be used to determine influence of a certain user over the topology, basically a celebrity or any popular person will surely have a higher page rank because of higher number of incoming links to it. This can be used to our advantage as we rank up the people with higher ranks in our list because if the interests are aligning then surely the target user will be willing to follow the celebrity[5].

3.7 Mentions

Yet another approach to identify popular users on the topology in context with our target user. It is assumed that whenever a user is mentioned in someone's tweet he/she is related with him/her in some way and have some similar interests. We count the frequency of mentions and thus rank users simply based on it giving us a rough idea about the popular users of that part of the topology.

$$MS(u_i) = \sum_{u_k \in U} MC(u_k)$$

Equation 8

IV. IMPLEMENTATION

4.1 Implementation

```
TextFile Compose-PageRank-Input( ) {

        pageRankInputFile = Create new TextFile
        userList = get list of all users from database
        for each user in userList {
                userID = get ID of user
                retweetList = get list of all retweets of userID
                for each retweet in retweetList{
                        authorUserID = Find original author ID of retweet
                        retweetLink = Create link between userID and authorUserID
                        Save retweetLink to pageRankInputFile
                }
        }
        return pageRankInputFile
}
```

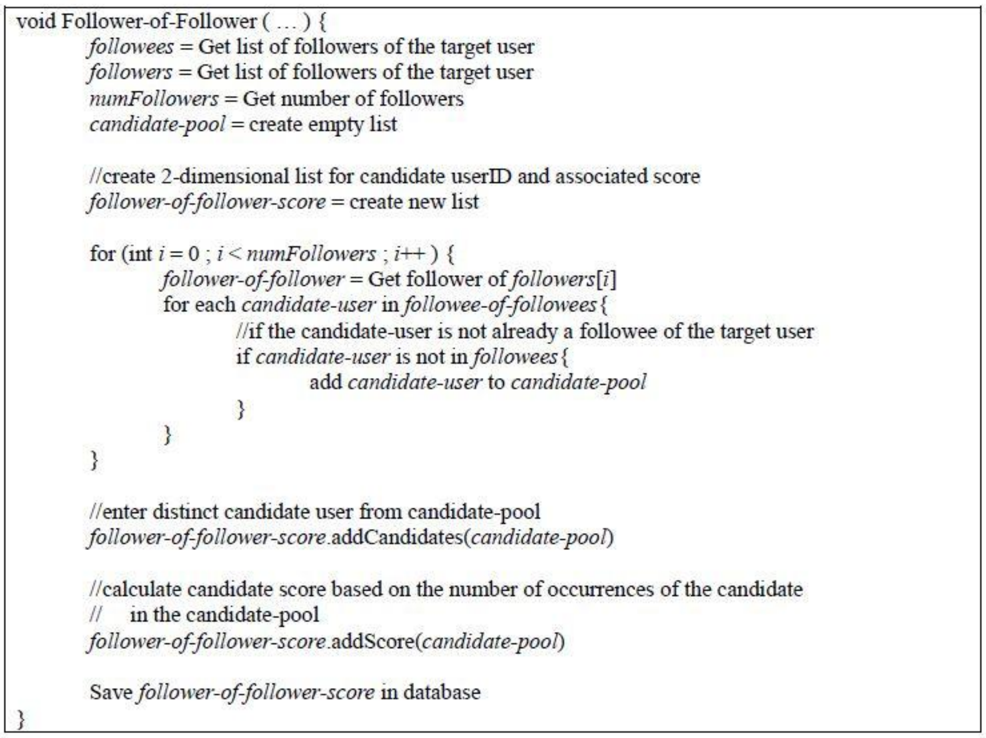
```
void Calculate-Retweet-PageRank( ) {
        inputFile = composePageRankInput ( )
        //Compute PageRank based on formula 2.1, 2.2 and 2.3
        ranking-scores = PageRank(inputFile)
        save ranking-scores in database
}
```

```
void Calculate-Similarity( ) {
    numUser = Get total number of users in the dataset
    users = Get list of users in the dataset

    for (int i = 0; i < numUser; i++) {
        tweets = Get tweets of user[i]
        Compose Document[i] using tweets of user[i]
        Create vector[i] of Document[i]

    }

    scoreList = create new score list

    for (int i = 0; i < numUser; i++)
    {
        for (int j = i+1; j < numUser; j++)
        {
            similarity-score = Calculate Similarity Score (vector[i], vector[j])
                       using Formula 2.4
            Save score in scoreList

        }

        find top k similar users from scoreList
        save similarity-score in Database

    }

}
```

```
void Followee-of-Followee ( ) {
    followees = Get list of followees of the target user
    numFollowees = Get number of followees
    candidate-pool = create new list

    //create 2-dimensional list for candidate userID and associated score
    followee-of-followee-score = create new list

    for (int i = 0; i < numFollowees; i++ ) {
        followee-of-followees = Get followees of followees[i]
        for each candidate-user in followee-of-followees {
            //if the candidate-user is not already a followee of the target user
            if candidate-user is not in followees {
                add candidate-user to candidate-pool
            }
        }
    }

    //enter distinct candidate user from candidate-pool
    followee-of-followee-score.addCandidates(candidate-pool)

    //calculate candidate score based on the number of occurrences of the candidate
    //    in the candidate-pool
    followee-of-followee-score.addScore(candidate-pool)

    Save followee-of-followee-score in database
}
```

```
void Followee-of-Follower ( ) {
        followees = Get list of followees of the target user
        followers = Get list of followers of the target user
        numFollowers = Get number of followers
        candidate-pool = create empty list

        //create 2-dimensional list for candidate userID and associated score
        followee-of-follower-score = create new list

        for (int i = 0 ; i < numFollowers ; i++ ) {
                followee-of-follower = Get followee of followers[i]
                for each candidate-user in followee-of-followees {
                        //if the candidate-user is not already a followee of the target user
                        if candidate-user is not in followees {
                                add candidate-user to candidate-pool
                        }
                }
        }
        //enter distinct candidate user from candidate-pool
        followee-of-follower-score.addCandidates(candidate-pool)

        //calculate candidate score based on the number of occurrences of the candidate
        //    in the candidate-pool
        followee-of-follower-score.addScore(candidate-pool)
        Save followee-of-follower-score in database
}
```

```
void Follower-of-Followee ( ) {
        followees = Get list of followees of the target user
        numFollowees = Get number of followees
        candidate-pool = create empty list

        //create 2-dimensional list for candidate userID and associated score
        follower-of-followee-score = create new list

        for (int i = 0 ; i < numFollowees ; i++ ) {
                follower-of-followee = Get followers of followees[i]
                for each candidate-user in followee-of-followees {
                        //if the candidate-user is not already a followee of the target user
                        if candidate-user is not in followees {
                                add candidate-user to candidate-pool
                        }
                }
        }

        //enter distinct candidate user from candidate-pool
        follower-of-followee-score.addCandidates(candidate-pool)

        //calculate candidate score based on the number of occurrences of the candidate
        //    in the candidate-pool
        follower-of-followee-score.addScore(candidate-pool)

        Save follower-of-followee-score in database
}
```



```
void Follower-of-Follower ( … ) {
        followees = Get list of followers of the target user
        followers = Get list of followers of the target user
        numFollowers = Get number of followers
        candidate-pool = create empty list

        //create 2-dimensional list for candidate userID and associated score
        follower-of-follower-score = create new list

        for (int i = 0 ; i < numFollowers ; i++ ) {
                follower-of-follower = Get follower of followers[i]
                for each candidate-user in followee-of-followees {
                        //if the candidate-user is not already a followee of the target user
                        if candidate-user is not in followees {
                                add candidate-user to candidate-pool
                        }
                }
        }

        //enter distinct candidate user from candidate-pool
        follower-of-follower-score.addCandidates(candidate-pool)

        //calculate candidate score based on the number of occurrences of the candidate
        //    in the candidate-pool
        follower-of-follower-score.addScore(candidate-pool)

        Save follower-of-follower-score in database
}
```

V. CONCLUSION

This marks the conclusion of description of this work, "Twitter Followee Recommender System". Final outcome were multiple strategies being developed for recommendations and the basic modules being implemented. Putting all together it shows how helpful it can be in enhancing the internet experience using some very simple methodologies.

REFERENCES

[1] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao and Yong Yu, Collaborative Personalized Tweet Recommendation

[2] Marcelo G. Armentano, Daniela Godoy and Analía Amandi, Towards a Followee Recommender System for Information Seeking Users in Twitter

[3] Masudul Islam, PERSONALIZED RECOMMENDER SYSTEM ON WHOM TO FOLLOW IN TWITTER

[3] J. Lin, A. Sharma, D. Wang and R. Zadeh, WTF: The Who to Follow Service at Twitter

[4] J. Hannon, M. Bennett and B. Smyth, Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches

[5] T.Y. Liu , Learning to Rank for Information Retrieval

[6]. Ahmed Tijjani Dahiru. "RELIABILITY CENTERED MAINTENANCE MODELLING ON POWER SYSTEMS." International Journal of Advanced Technology in Engineering and Science 3.Special Issue No. 01 (2015): 42-48.