

Trust_System

How to know which fact is relevant and useful within a social networking site.

¹Hemant Kumar Kushwaha, ²Ms. J. Jeysree

¹M.Tech.Scholar, ²Assistant Professor

Department of Information Technology, Faculty of Engineering and
Technology, SRM University, Kattankulathur-603203, India

¹hemantsrmit@gmail.com , ²jeysree.j@ktr.srmuniv.ac.in

Abstract— Trust plays a very useful role in our daily life. In the modern life where everything is based on the virtual world say World Wide Web, it is very difficult to know about whom to trust and why. Most of the social recommender systems are based on the trust rating of users. Hence, it's important to calculate the possible relations between users in the social network and to evaluate those relations properly to improve the social recommender system for a particular user. This particular user would be termed as ego user in our Trust_System. As the popularity of social network sites grow on the worldwide web, each individual has a long friend-list on facebook. While evaluating closeness among friends is difficult, fortunately there are many user actions which can be used to evaluate their relation on the social network. The trustworthiness of a product can be determined by the facts which are being calculated by their opinion about the given facts. In this research work Trust_Friend algorithm and True_Fact algorithm are used for solving both problems e.g. closeness of friends for social recommender system (Trust_Recommender for Trust_System) on a popular social networking site facebook.

Index Terms—Component Trust, Trustworthiness of facts, social network, internet, worldwide web, truefact algorithm, facebook4j, Trust_Recommender, Trust_Friend, Java, netbeanIDE.

I. INTRODUCTION

Worldwide web plays a very important role in modern life of a common man and ever growing popularity of the internet and worldwide web had lead to virtualises even their friends on the social networking sites like Facebook, Twitter, and Google-Plus. For our daily life and source of information we are dependent on the internet e.g. If anyone wants to book a show ticket for tomorrow he would like to check the reviews of movies or before buying a product he would like to check the internet for product review and ask opinion about the product to his/her friends. Today's life has become so busy that he cannot ask for everything by disturbing them on call or meetings. Then how to get the information we are searching for?

Till date much research work has gone into solving the problem related to trust calculation between users within a social network to implement a more relevant recommender system, and how to provide highly accurate data in IR system with a single click on the worldwide web. Like review algorithms for recommender system, trust metric can be used for calculating the trust between end users. Authority hub analysis and page rank for IR systems are examples of few of them. Many scholars are working on the same problem to make it much clear and accurate.

1. *Search Engines And their functionalities* - Basically it is an information retrieval (IR) system. The popular search engine Google uses Page rank algorithm for searching the WebPages based on the link quality to that page.
2. *Trust metric* - Trust metrics are tools for predicting the trust of a user in another user. Trust and trust metrics, are used to evaluate trust relationships between individuals. These are defined in different ways according to their scope. But having a closer look on them, these can be categorized according to their functionalities - Global Trust metrics and local trust metrics. Trust_system uses the concept of local trust metrics as it is able to calculate the trust of user within a group.
3. *Review algorithms* - It is being used in both IR system as well as the social recommender system on social networking sites. As it is said that modern market is consumer based. Yes! It is due to the virtualization of procurement and online marketing. The people would like to test a new product according to the past experiences of other consumers.

II. ANALYSIS AND DESIGN OF SYSTEM

A. The System Definition:

The Trust_System puts complete interface design of system using the Java language on the Netbean IDE to implementing algorithms and fetching data from facebook and posting the messages on the facebook. For database, it is using MYSQL database and JDBC technology at the client side to create a database connection for storing the fetched data and retrieving them for use in algorithms. For fetching the data sets from Facebook it is using FQL (facebook query language) and facebook4j library.

B. Functional Requirement

- Function of Application Registration: For accessing data from Facebook we registered as facebook developers and created our new application named Trust_System. From there we have provided an application_id and application_secret to use them in our coding modules.
- Function of User Authentication: When anyone wants to use our application, he has to grant access for getting the different permissions which are used to access information related to the trust_system user.
- Function of facebook premises: There are many more permissions available on facebook which can be used for

different type of applications. Permissions allow application to fetch the user information from the facebook. E.g. read_friend_list permission allows the application to fetch the friend list of the user, basic_info allows application to fetch information of the user.

- Good Network Connection without any 3rd party proxy: as for using our application we have to login on facebook without third party proxy otherwise we are getting error message: `FacebookException {statusCode=-1, response=null, errorType='null', errorMessage='null', errorCode=-1, errorSubcode=-1}`. We need network connection with good strength. If we use 3rd party proxy the proxy may block our connection.

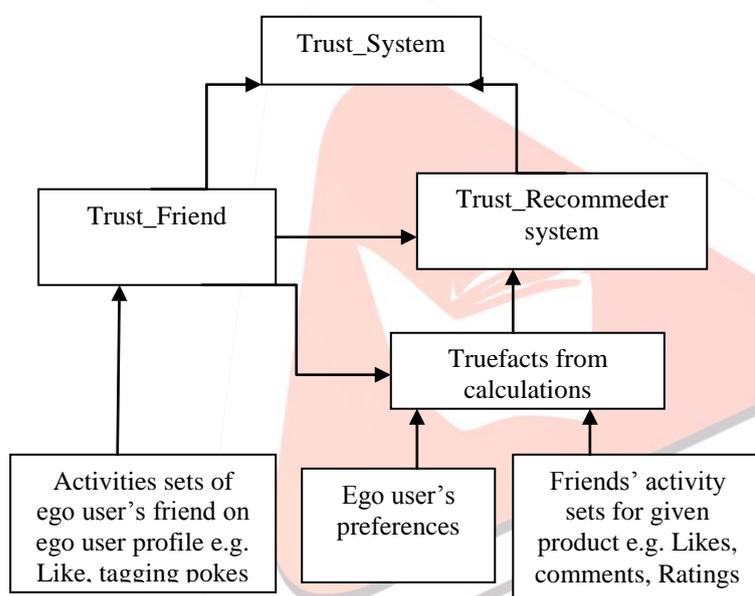
C. Detailed Design

The architecture diagram of Trust_System:

The Trust_System model is being deployed using the bottom-up design model.

E.g. the defined system is using two modules. One for calculating the closeness of friends on the facebook and another is the social recommender system on facebook.

1. Trust_Friend: It is calculated by an algorithm called Trust_Friend_Algo. This module is a part of our Trust_System and is also useful in determining the Trust_Recommender system.
2. Trust_Recommender: It is useful to determine the recommender product to the ego user.



{fig.1 Trust_System Architecture}

The friend's activities set are able to define Trust_Friend and the trustfact is being defined by the friend's activities sets for the given product and the ego user preferences.

III. TRUST_SYSTEM CALCULATION MODEL

While searching for information regarding any topic or preferred product in case of Trust_System, it's very difficult on www to determine that the given set of information is correct.

A. How it works?

The Trust_System deals with two main problems.

1. Discovering the trust between social users: Trust_System is using an algorithm called Trust_Friend_Algo to find the trust rating of different friends for an ego user based on friend's activity sets with the ego user.
2. Second, we are using their ratings and reviews and applying these in another algorithm called True_Fact. True_Fact selects best reviews and trusted products for the ego user which are recommended by Trust_Recommender to the ego user.

B. Algorithms.

1. Trust_Friend: Before going to the Trust_Friend algorithm, we should have a look at the trust-metric and trust in social networks as Trust_Friend is a type of a trust metric within a friend group.
 - a. *Trust metric*: There are different types of trust metrics. Each metric is designed to serve different purposes. They are used for trust modelling and reasoning about trust. They are closely related to [reputation systems](#). The first commercial forms of trust metrics in computer software were in applications like [eBay's Feedback Rating](#).

- b. Trust in social networks:** Searching trust in social networks, there are two main questions:
1. Who are user’s friends that he/she can trust most?
 2. How to measure trust between a user and his/her friends-of-a-friends

In our work, we are focusing on the problem – how to know our trusted friend e.g. problem one.

1.1. *Model for calculating Trustfriend:* Trust_System is using social activities set as a social interaction between two directly connected social network users (i.e., facebook friends). The weightage of each and every activity is calculated. Social activities by a user on facebook for our work are: tagging of ego user by an ego user’s friend on photo uploaded by the friend, likes, comments, share, poke, wallposts etc.

For calculating the trustfriend we are calculating the weightage of every activity set. Then we are using it for trust calculation

1. *Weightage calculation:* It can be calculated for an individual user’s (friend) activity with respect to the ego-user.

$M = A \text{ particular action} * \text{total of that action by all friends on ego-user's profile};$

$S = \text{Total of all activity set of all friends of ego-user with respect to ego-user};$

$W_{(weightage)} = M/S.$

2. *Trust calculation:* The trust rating of an individual with respect to a friend can be calculated as follows:

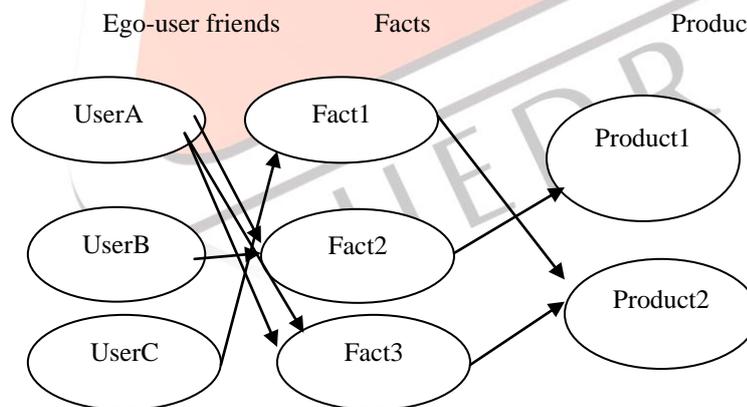
$S_a = \sum \text{of activity on users profile (In w weightage)}$

$S_p = \sum \text{of all activity of selected user (friend) on user's profile (In weightage)}$

Hence, the trust rating of particular friend = $S_p/S_a \times 100.$

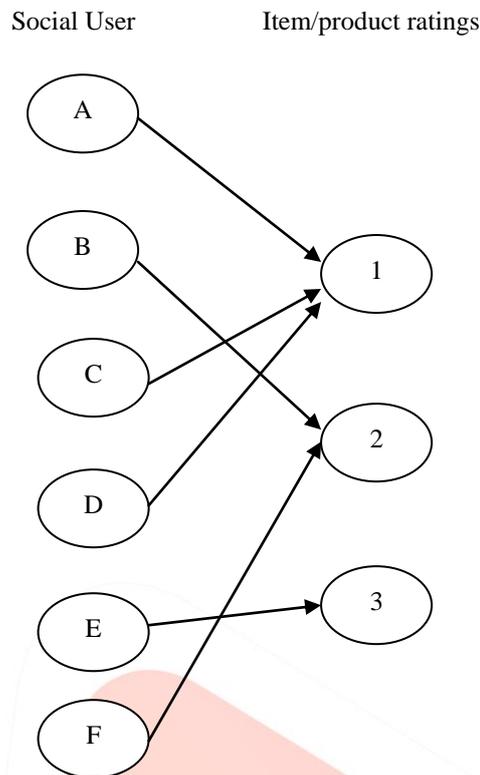
2. Calculation of trustworthiness of facts provided by friends (Trust_Recommender):

Trust_Recommender of Trust_System is being defined by the review ratings and likes for a given product within a social networking site Facebook.



{fig.2: graph showing the calculation of facts}

This graph is showing how Trust_System is calculating the facts provided by the different friends on different product.



{Fig.3 graph showing the product with higher ratings and being recommended. }

By the presented graph we can evaluate the accuracy of facts, as item1 is trusted by more users then item2 and then item3. Trustworthiness of the facts for Trust_Recommender may be defined in two ways.

a. Confidence of facts of ego user :

The confidence of a fact F_{acc} (denoted by $c(f)$) is the probability of f being correct: it is determined according to the past history of ego user and ego user preferences.

b. Trustworthiness of fact provided by ego user's friend:

The trustworthiness of fact F_t (denoted by $t(f)$) is the expected facts provided by ego user's friends by their opinions like, reviews ratings.

Considering the ego user's past history about products, his preferences and activities sets are collected from ego user's friends. For Trust_Recommender activities sets can be defined as the ratings of products, reviews of products, opinion about the product from the ego user's friend.

Suppose following assumption:

1. If a fact is calculated by getting the activities sets from many friends of ego user's, it is likely to be true;
2. If a fact is conflicting with the facts which are calculated by getting activities sets of ego user's friends, it is unlikely to be true.
3. On the other hand, a fact is trustworthy if it's calculated on the basis of data sets collected by many trusted friends.
4. We can see that whether the fact is trustworthy, can be determined by friends review rating and we can use an iterative method to compute both.
5. Because true facts are more consistent than false facts, it is likely that we can distinguish true facts from false ones at the end.

As mentioned above, the trustworthiness of a fact can be defined as the fact which is getting calculated by the data sets based on ego user's friend's activities. For a particular fact, we compute its trustworthiness $t(f)$ by calculating the average confidence of facts provided by trusted friends.

$$t(f) = \frac{\sum_{f \in F(t)} c(f)}{F(t)};$$

Where $F(t)$ is the set of facts provided by trusted friend.

IV. RELATED WORK ON TRUST_SYSTEM

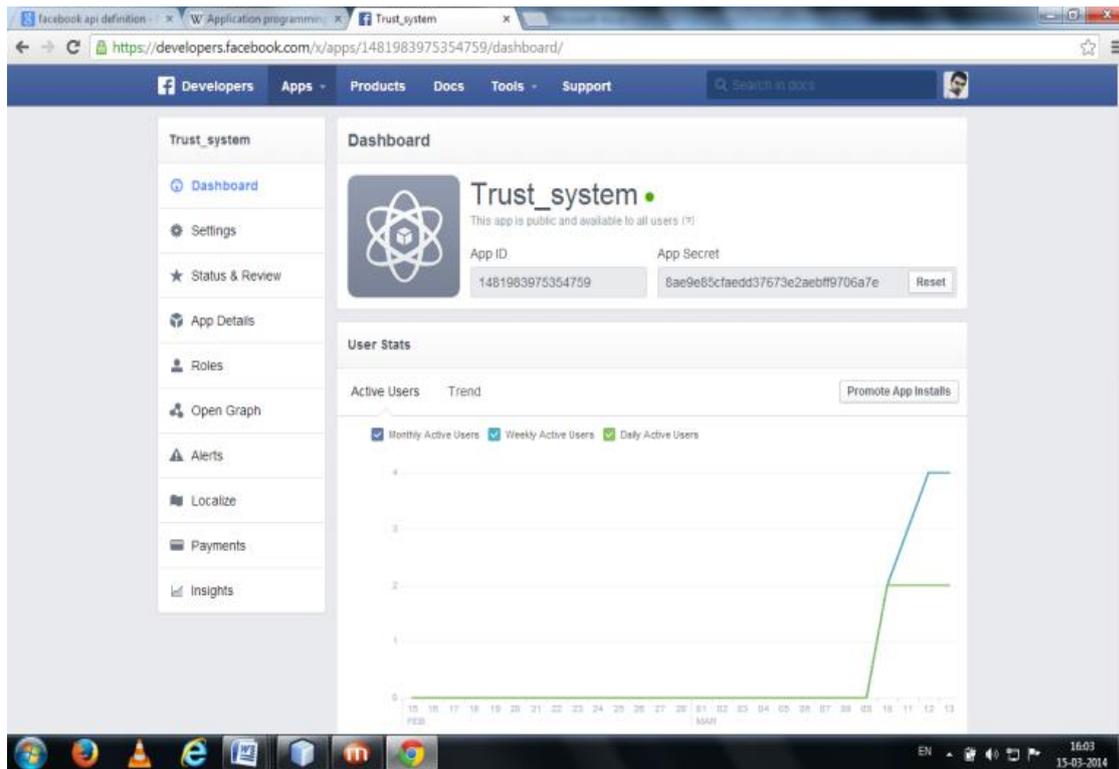
1. MAKING FACE BOOK APPLICATION:

There are many APIs provided by facebook for the developing third party facebook applications. Among them, most popular is Facebook graph API used for developing the games for the facebook. Facebook also provides some of SDKs to facebook developers e.g. php SDK, JavaScript SDK

1. Registering for application on Facebook:

There are simple steps to registering our application on facebook.

1. Registering as a facebook developer.
2. Register a new application and get verified.
3. Setting up application.
4. Getting App_ID and APP_secret.



{Fig.4 Registering Trust_system and getting App_ID and APP_secret}

2. Getting access token and Authentication for application: Further there are few steps for getting Authentication. There are two ways to get verified by facebook on using application.

1. Automatically: using php and JavaScript it is quite simple.
2. Manually getting access token for application:
 1. Go to the given URL and get authentication token for application.
https://graph.facebook.com/oauth/authorize?client_id=1481983975354759&redirect_uri=http://www.facebook.com/connect/login_success.html&scope=basic_info,email,publish_actions,publish_stream,read_friendlists
 2. https://graph.facebook.com/oauth/access_token?client_id=1481983975354759&redirect_uri=http://www.facebook.com/connect/login_success.html&client_secret=8ae9e85cfaedd37673e2aebff9706a7e&code=MY_VERIFICATION_CODE
 Now go to this and get access token, which is being used in application.

2. ACCESSING FACE BOOK THROUGH JAVA

As facebook is providing its own SDK for application development for the third parties like PHP SDK Graph API there and JavaScript SDK, there is no core library or SDK in java for developing facebook application except android API for developing applications. But it can be done using different libraries like facebook4j for developing a facebook application using java.

Anyhow there are many works from scholar and are available online to access facebook by java. Some of them are <https://code.google.com/p/facebook-java-api/>, <http://restfb.com/>, <http://spring.io/guides/gs/maven/>; <http://facebook4j.org/en/index.html>. Trust_system is using their references and code example for being developed. The important ones among them are facebook4j and restAPIs.

- 1) *Facebook4j*: Facebook4J is an unofficial library for integrating java applications with facebook API. It is open-sourced, mavenized Java library.
- 2) *Java API for face book*: It is java library for integrating facebook graph API. Provided by a Google community.
- 3) *Restfb*: Restfb is a simple and flexible Facebook Graph API and Old REST API client written in Java. It is open source software released under the terms of the MIT License.
- 4) *Snapshots of working model*

V. SNAPSHOTS OF WORKING MODEL

1. CODE SAMPLES WITHOUT PUTS:

C. Fetching data from facebook.

a. Fetching user information and posting on facebook:

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package trust_system;

import facebook4j.Facebook;
import facebook4j.FacebookFactory;
import facebook4j.Friendlist;
import facebook4j.Post;
import facebook4j.ResponseList;
import facebook4j.User;
import facebook4j.auth.AccessToken;

/**
 *
 * @author hemant
 */
public class Userinfo {
    public static void main(String args[]){
        try{
            Facebook facebook = new FacebookFactory().getInstance();
            facebook.setOAuthAppId("1481983975354759","8ae9e85cfaedd37673e2aebff9706a7e");
            facebook.setOAuthPermissions("basic_info,email,publish_actions,publish_stream");
            facebook.setOAuthAccessToken(new
            AccessToken("CAAVD20hp8YcBAAkMWBNSk2ZAymUpkzfmPu8G6111mh20E7MKx0WqOy0EIBkMINSbrvwL5
            Vceln9n2bSCGnPUMJVAL6lADR9wyjpnZANO7DeVofQAsvMoFDCq0uNMmJDSUcaKbVCuwqSamS5Ma5Bd04
            0d9TYLfs5zkSkvOm2v4ZC0kMu"));
            User user=facebook.getMe();
            facebook.postStatusMessage("QUOTES from :"+user.getName());
            facebook.postStatusMessage(" :"+user.getQuotes());
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

Output:



{Fig.5 output which showing the status post on facebook by Trust_System}

b. Fetching friend list:

```
import java.util.Map;
import facebook4j.*;
public class Goodabletogetallfriendsname {
    public static void main(String[] args) throws JSONException, FacebookException {
        try
        {
            Facebook fb=new FacebookFactory().getInstance();
            fb.setOAuthAppId("1481983975354759","8ae9e85cfaedd37673e2aebff9706a7e");
            fb.setOAuthPermissions("basic_info,email,publish_actions,publish_stream,read_friendlists");
            fb.setOAuthAccessToken(new AccessToken("CAAVD20hp8YcBAHSxbz0ZCWRRjYVOn3ss34"));
            Map<String, String> queries = new HashMap<String, String>();
            queries.put("all friends", "SELECT uid2 FROM friend WHERE uid1=me()");
            queries.put("my name", "SELECT name FROM user WHERE uid=me()");
            Map<String, JSONArray> result = fb.executeMultiFQL(queries);
            JSONArray allFriendsJSONArray = result.get("all friends");
            for (int i = 0; i < allFriendsJSONArray.length(); i++) {
                JSONObject jsonObject = allFriendsJSONArray.getJSONObject(i);
                User user =fb.getUser(jsonObject.getString("uid2"));
                System.out.println(":"+user.getName());
            }
        } catch(Exception e){
            e.printStackTrace();
        }
    }
}
OUTPUT:
```

```

Output - Fb_Access (run)
:Anjali Singh
:Shriya Shantacharjee
:Salim Raja
:Deependra Pratap
:Walshay Dicit
:Utkay Singh
:Saty Jain
:Shalish Verma
:Shrigda Talukdar
:Shatrughan Verma
:Shubham Saur
:Sabali Singh
:Sandeep Singh
:Satya Prakash Saini
:Anitha Anamath
:Shamshap Dey
:Prashant Singh
:Raj Kishor Yadav
:Anand Chaudhary
:Manish Singh Krishna
:Mahesh Pandey
:Viney Chatterjee
:Deepyoti Saini
:Saana Akash
:Shikant Yamamandra
FacebookException(statusCode=400, response=HttpResponse(statusCode=400, responseAsString={"error":{"message":"Unsupported GET request..."}, "type":"NotSupportedException", "code":100})
, isHttps=true, protocol=http, HttpURLConnection=HttpURLConnection@295559, streamConsumed=true), errorType=RequestMethodException, errorM
essage=Unsupported GET request., statusCode=100, statusCode=1}
at facebook$.internal.http.HttpClientWrapper.request(HttpClientWrapper.java:189)
at facebook$.internal.http.HttpClientWrapper.request(HttpClientWrapper.java:65)
at facebook$.internal.http.HttpClientWrapper.get(HttpClientWrapper.java:95)
at facebook$.FacebookImpl.get(FacebookImpl.java:2491)
at facebook$.FacebookImpl.getUser(FacebookImpl.java:114)
at facebook$.FacebookImpl.getOpen(FacebookImpl.java:110)
at trycode.Goodabletogetallfrindename.main(Goodabletogetallfrindename.java:38)
BUILD SUCCESSFUL (total time: 3 minutes 55 seconds)

```

{ FIG.6 OUTPUT OF FETCH FRIEND LIST FROM FACEBOOK. }

VI. FUTURE WORK

Improving Trust_Friend algorithm by implementing global trust metric: As Trust_System finds the closeness of two friends who are already connected so till now friend of friends is beyond its scope. In future the Trust_System can use the information provided by friends of friend. This would make the Trust_Recommender System of Trust_system more relevant and efficient. The review rating functionality can be improved by getting the activities sets of users who are not directly connected to the ego user but they are mutual friends and may be friend's friend. For this new trust metric can be purposed and implemented to help the ego user whom he could trust if he is not directly connected to them.

1. For making Trust_Recommender more efficient the inference theory may be used: Proposing four basic ideas that serve as the basis of Trust_Recommender future computational model.
 1. One True fact ($T_{(x)}$): Usually there is only one true fact about given product which can be calculated on the basis of different activities sets of an ego user's friends. It's based on their opinion for a property of an object and based on the ego user's opinion and preferences.
E.g. If we consider about a fact about a movie, there are several like and dislike from the social network user. But if considering the past experience of the ego user he may be like the particular movie from the particular studio, particular types, with the role play of his favourite actor.
 2. Similarities of facts ($S_{(x)}$): This true fact may be the same or similar to those calculated by ego user's friends activities.
 3. False facts have smallest probability to being same ($F_{(x)}$): The false facts calculated by getting activities sets from different ego user's friends are less likely to be the same or similar.
E.g. In our case of movie reviews the rating and the opinion will differ for a movie which is not good enough.
 4. Surety about the true fact ($St_{(x)}$): The fact which is calculated by getting activities sets from some friends who are trusted by the user and have the same preferences have the higher impact and surety.

And the above assumptions may be calculated by the mathematical logics.

VII. CONCLUSION

This paper presented a general model for the calculation of trust among social users on social networking sites, as well as described how a general model of trust is useful to make a recommender system for a social user. For that a Trust_System is being deployed using a specific social network as a medium (i.e., Facebook). We propose set of two algorithms – first one for calculating the trust among the users Trust_Friend_Algo for Trust_Friend module and another for recommender, and we are using them to define our Trust_System model. And from here we have introduced two other problems, one how to calculate a trust rating for a friend who is not directly connected to ego user, and another how to make Trust_Recommender more efficient if there are conflicting facts. This paper has purposed the future scope by defining and using the new trust metric and resolving conflicting facts provided by friends on social network. This paper also purposed four basic assumptions which can be derived by mathematical logic for finding the true facts. And then Trust_System can use both of them for more accuracy.

ACKNOWLEDGMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Mrs. J. Jeysree for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her time to time shall carry me a long way in the journey of life on which I am about to embark.

I am also thankful to my friend Vinav Chatree, for the valuable help provided by him. At last I am very thankful to my parents encouraging me for my studies and projects.

REFERENCES

- [1] <https://developers.facebook.com/>
- [2] <http://restfb.com/>
- [3] <https://groups.google.com/forum/#!forum/restfb>
- [4] <https://github.com/revetkn/restfb/issues>
- [5] <http://facebook4j.org/en/index.html>
- [6] <http://en.wikipedia.org/>
- [7] Java for programmers, Detale Developers Series
- [8] Vedran Podobnik, Darko Striga, Ana Jandras, Ignac Lovrek "How to Calculate Trust between Social Network Users?" University of Zagreb Faculty of Electrical Engineering and Computing Zagreb, Croatia
- [9] Thomas DuBois, Jennifer Golbeck, and Aravind Srinivasan "Predicting Trust and Distrust in Social Networks" 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing.
- [10] Sai T. Moturu, Jian Yang, Huan Liu "Quantifying Utility and Trustworthiness for Advice Shared on Online Social Media" 2009 International Conference on Computational Science and Engineering.
- [11] Xiaoxin Yin, Jiawei Han Senior Member IEEE, and Philip S. Yu Fellow IEEE "Truth Discovery with Multiple Conflicting Information Providers on the Web" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 20, NO. 6, JUNE 2008.

