FOLLOW YOUR NEIGHBOURS AND ENGAGE IN A NEW CULTURE

by

Meshary Abdulrahman AlMeshary

Bachelor in Computer and Information Sciences

King Saud University, KSA, 2006

A thesis

presented to the Ryerson University

in partial fulfilment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada,2015

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Follow Your Neighbours and Engage in a New Culture

Master of Science 2015

Meshary Abdulrahman AlMeshary

Computer Science

Ryerson University

Abstract

Twitter is one of the popular social media websites. It has more than 400 million active users. They post a huge number of tweets daily to share their opinions and knowledge in different languages and locations. Twitter has been used to distribute news, politics and more. This thesis proposes an approach to recommend new followees to Twitter users who just moved to a new place where the local language is different. A recommender system is developed that provides Twitter users the ability to adjust and engage in a new culture and helps them adapt to a new environment. This recommender system finds users' interests from his historical tweets in his mother language and looks for followees who have the same interests in the local language. This proposed system uses Twitter APIs to fetch local tweets after finding the location of the user and recommends similar local followees to the system user.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Abdolreza Abhari, whose expertise, understanding, patience and encouragement guided my graduate study experience.

I take this opportunity to express gratitude to all of the Computer Science Department at Ryerson University for accepting me to their program. I also thank my father and my family for their unceasing encouragement and support. I am also grateful to my friend, Michael McGinn who supported me through this venture.

In conclusion, I recognize that this research would not have been possible without the financial assistance of The Ministry of Education for the Kingdom of Saudi Arabia and I would like to express my deep gratitude to those agencies.

Contents

List of Tables

List of Figures

List of Appendices

Chapter 1

Introduction

1.1 Background

Recommender Systems (RS) [1] have been used in different sectors such as education, travel, sales, marketing and in endless commercial and personal ways. Recommender systems help users to make clear decisions and suggest other options. For example, in Netflix, the actor, director, or genre of the previously watched movies is used to create a list of suggested new choices based on the users likes, interests and history [2].
With the overload of information from social media websites, recommender systems provide the service of organizing and funnelling information and data to a series of more user friendly choices.

1.2 Social Network

Networks get things done, whether it is sending a letter or lighting your home, networks make it happen. Cities are connected by highways, road signs and are supported by maps. To travel by car from Toronto to London, Ontario we need to follow these highways to get there. We see Toronto is connected to Hamilton, which is connected to London. People

networks can help us with finding jobs, meeting new people and finding partners. This is how it works in the real world. Your friend Bob knows Sally and Sally's friend Joe has a job for you. This is a social network highway of people.

Social Network [3] refers to the ways of interaction or relationship between families and friends. Also social networking (social media) can be defined as categorizing and organizing information and sharing it between two or more devices .

There are two types of social networking: On-line and Off-line. Off-line Social Network is a real time conversation or face to face interaction between people in the real world. For example, meeting your friend at a coffee shop and talking about his wedding over a cup of tea. On the other hand, On-line Social Network (known as Social Media) is an interaction between people (Users) in websites. For instance, you post your " Happy Birthday!!" best wishes on your friend's Facebook page. The scope of off-line social network is hidden and most connections between people in the real world are not seen. Your network may have huge potential but it is only as valuable as the people and the connections that you can see.

Social media websites (On-line Social Networks) have solved this problem. These websites help people to see connections that are hidden in the real world. Here is how it works, users sign up for free accounts and create personal profiles. This profile page typically includes a photo, some basic personal information (name, age, gender, location) and extra space for listing your favourite bands, books, TV shows, movies, hobbies and Web sites. Then users look for and add people they know to their connections. When these connections are created between the users and their friends then a web of knowledge, sharing personal data, photos and ideas is created between all who have joined into this linking. Now all linked users are no longer strangers and all can be contacted or accessed more easily. All are visible and hidden opportunities are now revealed in a way not possible with Off-line Social Networking.

Connection between users on social media websites are categorized under two main types.

Type one is a Two-way connection "Friendship": both parties have consented to be in the relationship. 'A' invites 'B' and 'B' needs to accept the invitation from 'A' in order to create a connection. Broadly speaking, 'A' then sees what 'B' posts and 'B' sees what 'A' posts. Naturally, if either 'A' or 'B' chooses to "un-friend", the entire relationship disintegrates and neither party can see each other's posts. This type of connection is used widely on Facebook, LinkedIn, and MySpace.

Type two is a One-way connection "Following": 'A' follows 'B' and sees what 'B' posts. There is no obligation for 'B' to follow 'A'. It is a simple, one-way relationship. If 'B' chooses to follow 'A', it becomes a loosely coupled pair of one-way connections. Loosely coupled in as much that 'A' could choose to un-follow (disengage) from 'B', but 'B' following remains intact with 'A'. This type is used in many social media websites such as Twitter, Tumblr, Instagram, and Flickr.

The increased usage of smart-phones, tablets, and the availability of the Internet have made the social media just a finger click away, being used any time and anywhere.

Social media gives people a chance to find a few ways to explore what is happening in the world, meet other people, research opportunities and share ideas. Ultimately, social media websites help people work together to achieve goals that they share in common. Besides using social media to be social, people use social media to find information about products/services, keeping up to date with real-world social events, and learning new things. Also people use social media to kill time by sharing information about themselves, by publishing updates, videos, pictures, and people even market their own personal brand or business.

These days, Facebook and Twitter are the two major kings of the social media[4, 5]. This thesis will focus on Twitter.

1.3 Twitter

Twitter is a form of micro-blogging network [6], which allows users to send and receive short public messages known as tweets. Tweets are limited to 140 characters or less, and can include links to another website pages, images, videos and blogs. These days, Twitter is considered one of the biggest social media websites. It launched in March 2006. Since its launch, Twitter has amassed a large user base and now has more than 400 million active users. These users send more than 500 million tweets per day.

Twitter is the best way to discover what is new in the world by answering the question "What's happening?" by Twitter users. 140 characters, it might not seem like much but it is more than enough for the weather channel to let its followers know about the weather. Twitter allows its users to post short status messages called tweets. Tweets can be posted from various sources which include the Twitter website, Twitter mobile applications as well as several third party applications or websites. In general, Twitter is more public compared to other social media websites such as Facebook. The relationship between Twitter users is by default based on following relationships, which means there is no need to ask for friend requests as in Facebook. On the other hand, Twitter provides the privacy feature for its users to control their tweets by making them public and visible to any one or keep them private, which restricts the access to only some users who obtain permission from the user.

The growing popularity of Twitter [7, 4, 5] inspires people to use it on a daily basis to share some information about themselves or seek for information that interests them. Twitter is considered one of the fastest ways to distribute news locally or globally, stay in touch with friends and family, and coordinate events, marketing, business and build new learning. Figure1.1 shows the usage of Twitter.

Twitter has its own terms [8] associated with functions that it provides. The following will review and define these functions.

Figure 1.1: *Twitter Participates in many Areas.*

- **Tweet** - Each Twitter post or update is known as a tweet. It is up to 140 characters. The content of these tweets usually is the answer of "What's happening?". Some tweets may include hashtag and URLs to another websites, pictures.
- **Retweet** - When Twitter users repost or tweet another Twitter users' tweet.
- **Hashtag** - it is a part of the tweet usually referring to the tweet topic. It is a number symbol (#) used before a keyword of your tweet. Using hashtag is optional and you can use it more than once.
- **Follower** - They are Twitter users who want to receive or read your tweets.
- **Followee** - They are Twitter users that you want to receive and read their tweets.

In this thesis we are going to use our own terms . We define them below:

- **Local Tweets** - They are tweets from Twitter users in your area (your physical location).

- **System's User**- Twitter user who tweets in Arabic and moved or willing to move to English countries.
- **Local Users** They are Twitter users who tweet in English at the system user's location.

1.4 Problem Statement

Twitter users could find valuable information from the followee who tweets in the same language. While adding the followees who tweet in different languages to follow them, would be more productive. Being a new to a culture can be solved by linking this new person to people in the same area who have the same interests. One of the advantages of using social media information is studying the culture.

This thesis will explore and explain how Twitter social media information can be used to solve the problem of being new to a culture. By collecting historical tweets from this newly immigrated Twitter user, and finding his/her interests, other twitter users who live in the same location can be recommended, who share similar interests, to follow.

People who use social media have different tastes and interests [9]. Some of these people use social media for communication with others and to stay in touch with friends and families. In business and marketing, companies and business owners use social media to get the word out to their customers and to connect with other owners, local and abroad, in the same industry. Some people use social media to seek for information about what is new with famous celebrities and musicians and also to look for events and entertainment. Different people use social media in different ways.

It is obvious that every culture is governed by the people who live and work within it. Of course different people in different locations have different cultures, and these cultures have an effect on social media. Let us take Twitter for example, when the Twitter user moves to somewhere in United States and logs in to his Twitter account he will see a

different trending list (Hash-tags and hot topics) than from when he was in Canada. This is because people in United States have a different culture to tweet about with different topics from people in Canada. Culture changes from country to county, city to city and even from neighbourhood to neighbourhood in the same city. Language can also effect a big change in culture. For instance, the top 10 videos on YouTube's homepage in Montreal are drastically different from YouTube's homepage in Toronto, despite the fact that they are both cities in Canada. This is because the main language spoken in Montreal (French) is not the same main language spoken in Toronto (English). This example proves that social media is a mirror for current culture. People migrate and move from place to place for many reasons including work or study. With this migration they will face some difficulties to engage with the new environment and adapt to the new culture. Social media is the key to ease and finesse this transition to the new culture.

1.5 Methodology

Twitter is used almost everywhere in the world and it supports many languages. This thesis proposes a way to help Arabic Twitter users who have just moved to a new place where English is the main language spoken to engage with their new environment and streamline their adaptation to the new culture by recommending local Twitter users who have the same interests to follow.

Our goal in this thesis is to build a cross-language recommender system for whom-to-follow in Twitter. As we mentioned before, we are considering only two languages: Arabic and English. We assume that the user who uses our recommender system is an Arabic Twitter user, tweeting mostly in his/her home language Arabic, with some rare tweets in English. Also we assume the system's user can read and speak English. Here is a summary of this proposed solution. First, we use Twitter REST API to fetch the last 200 tweets of the Arabic Twitter user, and find the keywords for each tweet by

using Part-of-Speech tagging method. Part-of-Speech tagging is one of Natural Language Processing methods; it is discussed in detail in Chapter 3. Second, we create the user's vector (list of interests) in Arabic by counting the Term Frequency for each keyword. Third, we use a Bing translator to translate the user's vector to local language (English). Fourth, we use cosine similarity between the system's user's vector and all local Twitter users' vectors. Finally, based on the similarity score between the system's user and local users, the proposed system recommends the top 10 local users to follow who have the high scores. Chapter 3 will have a full explanation.

1.6 Objectives

The main objective of this research study is to recommend local Twitter users (local people or agents) to our user. That would help our user to understand the new culture and make easy to adapt.
The plan is to meet the objective described above:

1. - Correct the misspelled Arabic words before translating them to English.
2. - Find our users location and then start collecting local tweets based on that location.
3. - All local tweets must only be in Arabic or English.

1.7 Contribution

There is a contribution of our research:
We propose a novel recommender system that uses some Natural Language Processing (NLP) techniques to generate recommendations for Twitter users. These recommendations are lists of Twitter users to follow, who share interests and who tweet in different

language of our system user.

1.8 Thesis Outline

The remainder of the thesis is organized as follows:

Chapter 2 surveys the work already done by researchers in this area of interest. We review recommender system in general and discuss the filtering methods that have been used in recommender systems. It shows how recommender system is used in social media in general and Twitter in specific.

Chapter 3 explains the system architecture, the way of collecting Twitter data and the algorithms of filtering data and builds the recommendations.

Chapter 4 explains the experiment and how the proposed system will be evaluated.

Finally, Chapter 5 concludes our thesis work by summarizing our experiment results and the proposals for potential future research.

Chapter 2

Related Work

2.1 Introduction

In this Chapter, we would like to review some related works such as recommender system filtering approaches, recommender systems on social media in general, and recommender systems for Twitter.

2.2 Recommender System

Recommender system (RS) collects information about its users' preferences for a set of items such as songs, books, websites, applications, movies, and posts. RS uses the information to provide predictions and recommendations on items for its users. The information in RS comes from different sources. It could be the user's profile, basic information like age, gender, nationality, language, or it could also be information coming from the user's rating on a specific item. Another way to collect the information is monitoring user's behaviour, such as movies watched, books read, and websites visited. RS in Web 2.0 uses social information like followers, likes, posts as source of information [10].

There are two types of recommender systems [11], Personalized and Non-Personalized. Non-Personalized recommender systems do not use the users' preferences to generate recommendations. For example, recommending the top ten songs or app in iTunes. Personalized recommender systems consider the users preferences to be recommended. An example of this type of recommender systems can be found being used in Twitter, Facebook, Xbox and more. This thesis work will focus on the Personalized type of recommender systems.

The main function of any RS is to generate recommendations and these recommendations are based on a combination of the following factors:

- The filtering methods used in the RS may be an individual or a combination of approaches (e.g., Collaborative, Content-based, and hybrid).
- Data is required for input into the RS (e.g., user profile information, rating, tagging, user location, social relationships, and user followee).

All core functions for recommender systems are categorized by their filtering methods. The most widely used filtering methods are [12, 13, 14] (i) collaborative filtering, (ii) demographic filtering, (iii) content-based filtering and (iv) hybrid filtering.

2.2.1 Collaborative Filtering

Collaborative filtering [12, 15, 16] is a method of making automatic predictions about the interests of a user by collecting preference or taste information from many other users. In other words, using what you like in the past to predict what you will like in the future based on what people with similar interests liked.

There are two main types of Collaborative Filtering approaches:

1. **User_based:** Given an active user 'A' and an item 'I' not yet seen by 'A'. Find a set of users (nearest neighbours/peers) who liked the same items as 'A' in the

past and who have rated item 'I'. Then using the average of their ratings to predict if 'A' will like item 'I' or not. Do this for all items that have not been seen and recommend the best rated one for active user 'A'.

2. **Item_based:** Use item-to-item similarity based on users' likes and rating. For example, user 'A' liked item 'I1' and at the same time, item 'I1' was liked by users 'C','B', and 'D'. On the other hand, users 'C','B', and 'D' like item 'I2' so most likely user 'A' will like item 'I2'as well.

2.2.2 Demographic Filtering

It uses user's basic information such as gender, age, location, and language to make recommendations. YouTube uses the user location to recommend some videos[17, 18, 16].

2.2.3 Content-based Filtering

Collecting the user selections made in the past will help predict and generate recommendations for the future. For instance, in Netflix if the user watched a comedy movie in the past the Netflix recommender system will possibly recommend a recent comedy movie that the user has not seen. The item content can be analyzed to help find the similarity between two or more items. Again, Netflix categorizes all their movies by genera, actor, year, director and content and uses a system like this to help promote and sell their movies [19, 20, 21].

2.2.4 Hybrid Filtering

Also known of Hybrid Collaborative Filtering, this method can merge or combine together two processes, for example, Collaborative Filtering with the Demographic Filtering, Collaborative Filtering with Content-based filtering, or Demographic Filtering with Content-based filtering. Amazon recommender system is the best example for hybrid

collaborative filtering. If a customer wants to buy an item like 'A', Amazon recommends another item like 'B' to buy at the same time. The recommender system makes the connection that most people who buy 'A' usually buy 'B' with it at the same time because 'A' and 'B' have the similar content [22, 16].

2.2.5 Recommender Systems Limitations

There are some limitations with using recommender systems. The two most important problems are found with new users and new items. A new user with few ratings is not going to get much help to find the new user interests, which makes the recommender system almost not able to predict a recommendation for this new user [22]. Equally a new item with no or few ratings cannot be clearly predicted by the recommender systems.

To implement content-based recommendation, the system requires the list of significant keywords related to an item. To find this list, item contents needs to be signified in a format that is automatically analyzed by computers as texts or added as keywords manually to an item [12]. Keyword extraction methods such as information retrieval are used in recommender systems. However, these methods cannot be functional on another data types such as audio, video, or graphics, because they cause limitation on content-based recommender systems. Additional problems happen when two items are given the same list of keywords, which makes these items identical for the content-based system, which relies on keywords to generate recommendations. Using the same set of keywords will cause inaccurate results as the systems will be unable to see the difference between a well-written book, for example and a poorly written book[23].

To get accurate recommendation, the user is required to rate a significant number of items to build a platform for content-based recommendations. It will be hard for the system to predict good recommendations if the user has not used the system before or previously rated few items. In order for the system to generate accurate recommendations, the system needs first to collect and know the user's interests based on the

ratings he/she gave. This problem has been solved by using the hybrid filtering method (combining content-based and collaborative filtering).

2.3 RS in Social Media Websites

People making life and purchase decisions commonly ask and depend on suggestions and advice from family, friends or acquaintances when selecting the best items to buy. These days, people use the Internet to seek for more information and opinions to help them to make selections. We know that the Internet is full of products and services, but the Internet itself is not able to provide the users with ample recommendations for their needs. So social media is the key to providing recommendations, when combining recommender systems with social media we get a new set of options and observations that traditional recommender systems will not be able to achieve. These new options and observations will be discussed and reviewed under the following headings: 1 - Users relationship; 2 - Enhanced recommendation for unrated items; 3 - User content-based as a recommendation source.

- **Users Relationships (Social Influence)**

  Traditional recommender systems do not consider the on and off line social relationships (Social Influence) within have a significant impact [24]. Friends recommend products to each other because they trust each other and have similar likes, experiences and interests. Some businesses take the advantage of this relationship between people by channeling their recommender systems to achieve a huge success. For example, Hotmail used social influence to get more than 10 million users just in year and half with an advertising budget of US $50,000. Hotmail became popular all over the world and in some countries such as Sweden and India [25], Hotmail did not spend any money on advertising. This Hotmail case shows that user relationships are powerful when making decisions on buying items [24].

- **Enhanced Recommendation for Unrated Items**

  By combining recommender systems with social media, the recommender system will rely on the user's friends' preferences to recommend unrated items to the user [23].

- **User Content-Based as Recommendation Source**

  The main two sources for any recommender system are free text and rating [26]. In e-commerce websites they use customers reviews and experience with a specific item and share it with others customers to increase revenue [27, 28]. However, these comments can be inaccurate, which leads to ratings that will be extremely high or extremely low [29] To solve this issue, some researchers have proposed the use of social media as a source of data. They use computerized text mining methods and customer satisfaction reviews from their trusted contacts on social media

2.4 RS for Twitter and Cross-language

There is a need for recommender systems in social media websites to help the users make friendships with others. There are many academic studies that discuss the recommendation of users to follow or make friendships. Most of these academic studies used Facebook or Twitter because they are the most popular social media website platforms today.

In [30] authors proposed Twittomender, which is a system that suggests Twitter users to each other, content-based filtering is used to analyze the tweets content and collaborative filtering to study the relationship between followees and followers of a Twitter user.

Authors in this paper [31] built a recommender system that uses the users' tweets or retweets as a source of input to their system. The system starts by deleting stop words then extracting keywords, hashtags, and urls of each tweet and Retweet. It uses TF-IDF

to find the most important keywords, and hashtags. Finally, it looks for the Twitter users who have similar taste as the system user. The system gives the similarity scores between the user and Twitter users in five different dimensions: Socialness, Feedness, HashtagUsage, Retweeted, and TermVariation. Socialness gives a score of how the connectivity between the system user and the others. Feedness is the total number of urls in the user tweets or Retweets. HashtagUsage measures the frequency of using and adapting of hashtags. Retweeted calculates how many retweets are from the user. TermVariation measures the variation of words in the content. This thesis work will focus on keywords and hashtags.

Armentano et al. [32] created a system that recommends a list of followees candidates to follow. Their algorithm is based on the user neighborhood to look for followees who share common followees. They fetch and list the target user's followees' followees and followers and the user's followers' followees and followers. Then they sorting the candidates list based on these factors: the relative between the number of followers and followees, the rate of frequency of each candidate in the ultimate list, the rate of mutual friends. In our research we use almost the same methodology to find the local twitter user who has the most local followers.

Zangerle et al. [33] proposed a system to recommend the hashtags for the twitter user based on the content of the tweet. The system does not consider the users interests. Authors assume that the hashtag is the key or the topic to categorize the tweets. When the user types the tweet the system starts to fetch tweets that are similar to the new tweet. The system calculates the similarity score between the tweets by using TF-IDF schema. It then finds the top N highest similarity scores and extract the hashtags from these tweets and recommends these hashtags.

Authors of [34] have proposed a bilingual personalized recommender model for digital libraries. The model uses the web-log data of the digital libraries users to recommend academic articles in more than one language. The model builds a profile for every user.

This profile has information about the article's languages read or viewed by the user beside the basic information such as name, language and location. The model is built to support English and Chinese languages only. The first step is feature extraction for the user; which has keywords extracted from user queries sentences, brows the secondary sources and download full text of the articles. The feature vector, after translation, is divided into two vectors: Native language vector and Target language vector. The second step is using the user's feature vectors in different languages to recommend articles in more than one language.

2.5 Summary

In this chapter we reviewed the recommender system in general. Moreover, recommender systems filtering methods are defined. Also we gave an example for each method.
Several recommender systems have been proposed to help Twitter users perform information sharing and social interaction more easily. In addition, we showed some related works used recommender system in social media in general and in Twitter in specific.

Chapter 3

Methodology

In this chapter, we explain in depth our approach for implementing a recommender system for Twitter. The system is helping a Twitter user who has just moved to a new location where the main language spoken at the new location is different from his/her home language. This methodology will address the primary issue of how to make recommendations based on the "new to the area" type of user's interests to adapt and adjust to his/her new culture.

3.1 System Architecture

Figure 3.1 illustrates the components of the proposed system, which was used to recommend followees (local Twitter users) to the users. The main three components are: User Interests List, Local-Users Interests Lists, and Similarity Measurement. The system gets required data from Twitter APIs to start to generate recommendations. All data that is used in this system is public data or with each users permission.

To recommend local Twitter users to follow we need first to determine our user's new location. Google Maps API [1] provides a free service called Geolocation. Geolocation uses

[1] https://developers.google.com/maps/

the GPS technology in the user's devices, or at the user IP address to provide the user location coordinate (latitude and longitude) [35]. In order to fetch tweets from a specific geographical location by using Twitter APIs these coordinates need to be pinpointed. Twitter APIs make use of the bounding box locations, which are the latitude and the longitude for both the south-west and the north-east points of an area. To calculate the bounding box locations we need the user's location (latitude and longitude) and radius. As mentioned before, Google Maps API locator function provides the user location after it receives the user's permission to share their location. The radius is set by default as 30 Km. Algorithm 1 shows how to calculate the bounding box locations.

Algorithm 1: Set the Location

Input: User's latitude and longitude ,Radius
Output: The area points (South-west and North-east)

```
1 boundingBoxes = array()
2 EarthRadiusKm = 6371
  /* maxLon and maxLat refer to longitude and latitude for North-east
  point.  minLon and minLat refer to longitude and latitude for
  South-west point.  */
3 maxLon = round(lat + rad2deg(30 / EarthRadiusKm), 5)
4 minLon = round(lat - rad2deg(30 / EarthRadiusKm), 5)
5 maxLat = round(lon + rad2deg(30 / EarthRadiusKm /cos(deg2rad(lat))), 5)
6 minLat = round(lon - rad2deg(30 / EarthRadiusKm /cos(deg2rad(lat))),5)
7 boundingBoxes[] = array(minLon, minLat, maxLon, maxLat)
8 return boundingBoxes[]
```

3.2 Twitter APIs

Twitters users post more than 500 million Tweets on a daily basis. Some of these Tweets are accessible to scientists and developers through open APIs at no expense. In this section we will show how to use Twitter APIs, what kinds of APIs, and what kind of

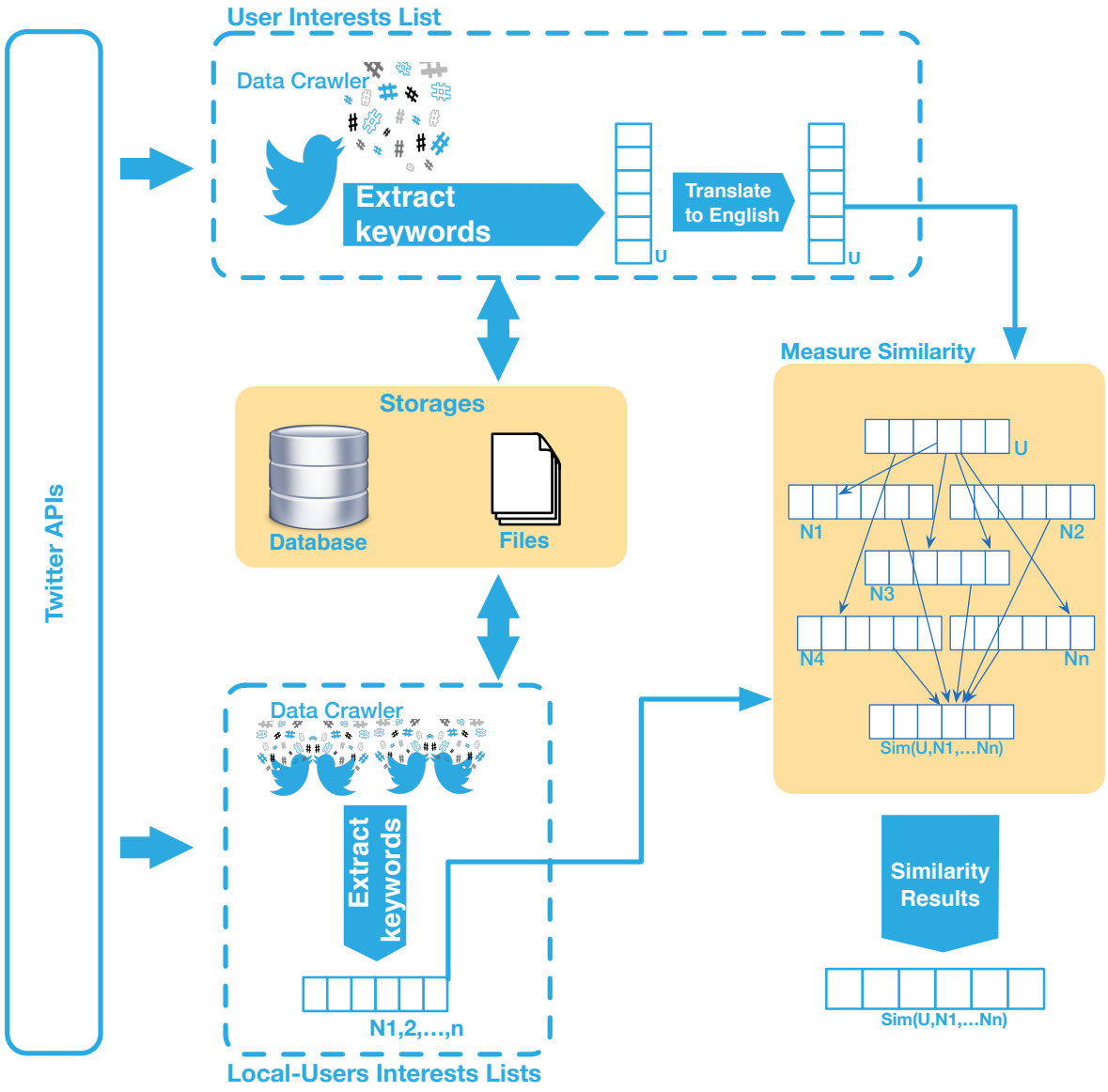


Figure 3.1: *Architecture of our recommender system.*

data we can get by calling these APIs.

To retrieve Tweets by using Twitter APIs we need an authorization. Twitter uses Open Authentication[2] (OAuth) to provide a secure access to its protected data. The only way to get OAuth (Access Token and Access Secret) is by creating an application. The application must be registered itself with Twitter. Authentication parameters are needed to make APIs calls. See table 3.1.

Consumer_Key	*"1x0RwyLVZNVdZ6py3F7gA"*
Consumer_Secret	*"R870vIXEOApLTsDjlGAVkCSjlw4wp2tuGgpCZHHrF5o"*
Access_Token	*"382306901-mDHVSq7641CsMSDJi7xnt8ShYqDo1gfT0M1SJhIk"*
Access_Token_Secret	*"475RPZk2SarkCtlU06eFunebwTV0TKff3qPIyXNNkQ"*

Table 3.1: Authentication Parameters

Twitter offers two types of APIs to interact with its own data. Each API type has its own functions and limitations. In our proposed system we call both APIs to get our required data to start to generating recommendations. The two types of APIs are:

1. **REST API**[3] gives a secure access to peruse and compose Twitter data such as write or post a new tweet or pull Twitter user profile information and followers. We use REST API in our system to retrieve the last 200 tweets for a specific user. REST API has limitations per request or call[4] and these limitations depend on OAuth parameters. To avoid some of these limitations we use Stream API.

2. **Streaming API**[5] provides continuous of streaming tweets. No limitations per call. We use Streaming API in our proposed system to retrieve tweets in specific location and specific languages.

[2]http://oauth.net/
[3]https://dev.twitter.com/rest/public
[4]https://dev.twitter.com/rest/public/rate-limiting
[5]https://dev.twitter.com/streaming/overview

Twitter APIs respond to requests with data in JSON format. We can get many kinds of data from calling Twitter APIs, including:

1. **User's Information**: basic information that appears in user profile such as name, location and description.

2. **User's Network**: user followers and followees.

3. **User's Tweets**: all information about the tweets themselves such as text, language, date, location information and more.

Figure 3.2 shows how we get data when we call Twitter APIs. We need to decode JSON format and save it in our Database.


```
{
  "id": 569967182609653760,
  "text": "#مايكروسوفت تسمح للطلاب والمدرسين حول العالم باستخدام #اوفيس مجانا\
          #أوفيس# تكنولوجيا #تقنية تعليم #Office\n\nhttp://t.co/IDEs5T09bt"",

  "source": "<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>",
  "user": {
    "id": 382647359,
    "id_str": "382647359",
    "name": "Ahmad Helal",
    "screen_name": "ahmad_helal",
    "location": "Jeddah",
    "description": "Director of @Flat6labsJeddah, Head of @GEWSaudi, Engineer, MBA @INSEAD,
                    "@Cornell Grad, PMP & CMA Certified اللهم ارزقني الحكمة وفصل الخطاب"",
    "url": "http://t.co/5zZJEp2xcE",
    "followers_count": 920,
    "friends_count": 1019,
    "geo_enabled": true,
  },
  "retweet_count": 23,
  "favorite_count": 21,
  "lang": "ar"
}
```


Figure 3.2: *Example of respond call for Twitter API in JSON format*

3.3 User Interests List

In this section we are going to discuss our methodologies that have been used to create the user interests list in figure 3.3.

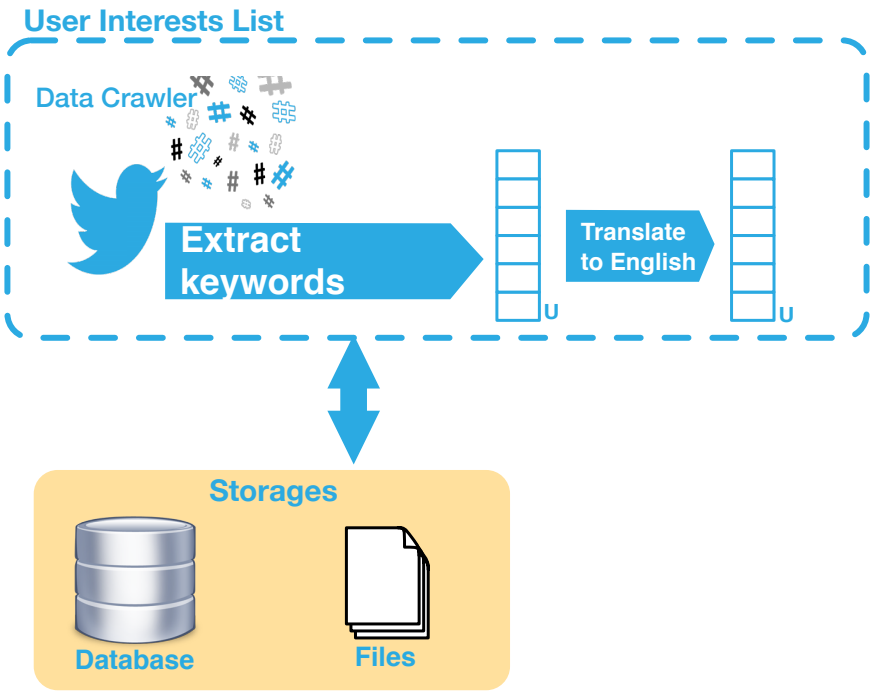


Figure 3.3: *User Interests List Subsystem*

To find the user interests list there are some sub-processes we need to go through. These sub processes are Data Crawler, Extract keywords, Translation and Interests List creation.

3.3.1 Data Crawler

The main purpose of our proposed system is to recommend local Twitter followees to our user who just moved to the new location where the language spoken is different. We

assume that our user only posted tweets in Arabic and the main language spoken at the new location is English.

Calling REST API to retrieve the last 200 Arabic tweets for our user. We get the data as raw tweets (See figure3.2) in JSON format then we decode these raw tweets and save them in our database. Algorithm 2 shows the calling process and the process of inserting data into the database.The time complexity of algorithm 2 is based on the number of tweets which is equal: $T(n) = O(n)$. Where n represents the number of the tweets where are saved in the database. Figure3.4 shows an example of UserTweets table in Database.

Algorithm 2: Retrieve the Last 200 Arabic Tweets for Specific Twitter User

Input: Twitter user id,Language code
Output: Insert 200 Tweets in Database

```
1 Tweets = array()
    /* ISO Language Codes is used and the code for Arabic is "ar"  */
        /* Call REST API and fetch last 200 tweets in Arabic */
  Tweets = twitter -> get('https://api.twitter.com/1.1/statuses
   user_timeline.json?screen_name=OurUser&count=200&lang='ar');

   for(i=0; i< Tweets.count;i++)
   {
     singleTweet=json_decode(Tweets[i]);

     insertIntoDatabase(singleTweet);
   }
```

3.3.2 Extract Keywords and Hashtags

In this process we manipulate tweet text and collect keywords and hashtags. Because we are dealing with Arabic text it is not an easy process to find the nouns (subjects or objects). We use Parts of Speech tagging technique (POS) [36]. Parts of speech tagging is one of the Natural Language Processing (NLP) techniques. It is the method of checking

Field Name	Value
Tweet_id	569967182609653760
User_id	382647359
Tweet_text	مايكروسوفت تسمح للطلاب والمدرسين حول العالم باستخدام #اوفيس مجانا,#تعليم #تقنية# #Office #تكنولوجيا #أوفيس
Lang	ar
Screen_name	ahmad_helal
Location	Jeddah
followers_count	920
followees_count	1019

Figure 3.4: *Example of user tweets table in Our Database*

a word inside a text and it assigns that word to a specific part of speech, depending on both its meaning, and its context such as noun, verb, adjective, etc.

The tool we use to split Arabic text based on POS is called Arabic Parser [37]. It was created by Stanford Natural Language Processing Group. It is free to use and licensed under the GNU General Public License [6]. Arabic Parser tool uses Penn Arabic Treebank [38] for tokenization of the text. We save all the Arabic tweets for one user in one text file. Then we run Arabic Parser tool for that text file and we collect all words that have

[6]http://www.gnu.org/licenses/gpl-2.0.html

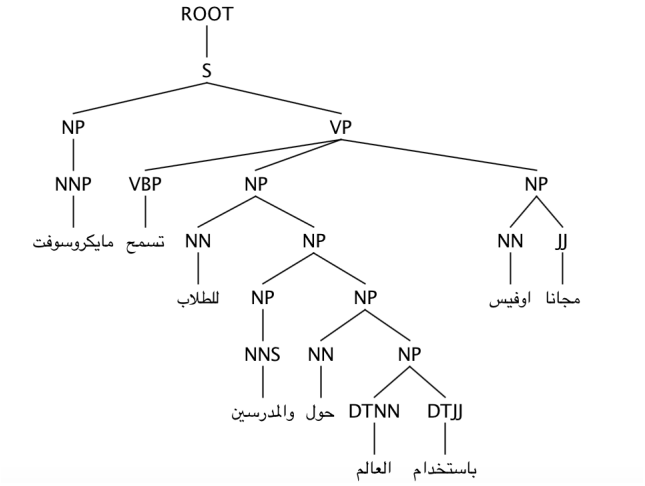


Figure 3.5: *Example of Part of Speech tagging*

POS tag	Description	Example
NP	noun phrase	a person, place, or thing
NN	noun, singular or mass	door
NNS	noun plural	doors
JJ	adjective	big
JJR	adjective, comparative	bigger
NNP	proper noun, singular	John
DT	determiner	the

Table 3.2: POS tagging meaning

POS tag that appear in table 3.2 or a combination of these tags and we ignore all stop words. See figure 3.5 for example of using Arabic Parser tool on Arabic tweet text.

3.3.3 Translation

After manipulating the tweet text and collect all keywords and hashtags we use Bing translator API to translate these list of words from Arabic to English. Before we translate we need to make sure that there is no spelling mistake.

We created our own subsystem to correct the most common misspelled words in Arabic. The subsystem has two lists. First list contains the correct form for each Arabic word. Second list has all the potential misspellings for each Arabic word. These lists are gathered by a collection of misspelled words from personal compilation, Arabic websites and Wikipedia[7]. The subsystem preforms this task by taking the Arabic word and checking to see if it exists in the misspelled list and then the subsystem provides the correct form for it. These lists are updated regularly by adding more words.

For example (See table 3.3) some people type Office in Arabic like this " اوفيس " or " إوفيس " which is not correct and that leads to inaccurate translation. We correct the spelling mistake by using our spelling check and the correct spelling for "Office" is" أوفيس ".

None_Correct_Form	Correct_Form
ايفون	آيفون
أيفون	آيفون
إوفيس	أوفيس
اوفيس	أوفيس

Table 3.3: Example of Arabic None Correct Spelling word and Correct Spelling word

3.3.4 Create the user's Interests List

Interest list is a vector that has all keywords and hashtags appear in up to last 200 Arabic tweets for specific user (our proposed system's user). We use this vector to find the similarity between our user and local users. The average size of the user's interests vector is around 1400 keywords. Table 3.4 shows an example of one Arabic tweet keywords (nouns

[7]https://ar.wikipedia.org/wiki/

and hashtags).

user_id	tweet_id	keywords	BingTraslate
382647359	569967182609653760	مايكروسوفت	Microsoft
382647359	569967182609653760	اوفيس	Office
382647359	569967182609653760	تعليم	Education
382647359	569967182609653760	تقنية	Technique
382647359	569967182609653760	تكنولوجيا	Technology
382647359	569967182609653760	أوفيس	Office
382647359	569967182609653760	Office	Office
382647359	569967182609653760	طلاب	students
382647359	569967182609653760	مدرسين	teachers
382647359	569967182609653760	عالم	world

Table 3.4: Example of one Tweet from User Interest List

3.4 Local-Users Interests Lists

Local users refer to Twitter users who are tweeting at the new location of our user (neighbours). In this thesis we focus on and retrieve only local users who tweet in English. See figure 3.6.

3.4.1 Data Crawler

Based on the user location for whom the system wants to provide recommendations (we use Google Maps to show the user's location see figure 3.7) and we call on the Twitter Streaming API to fetch some local tweets. These tweets have some information such as Twitter username, location, tweet text, and language. From these local tweets we compile a list of local Twitter users. For each local Twitter user, we retrieve the last

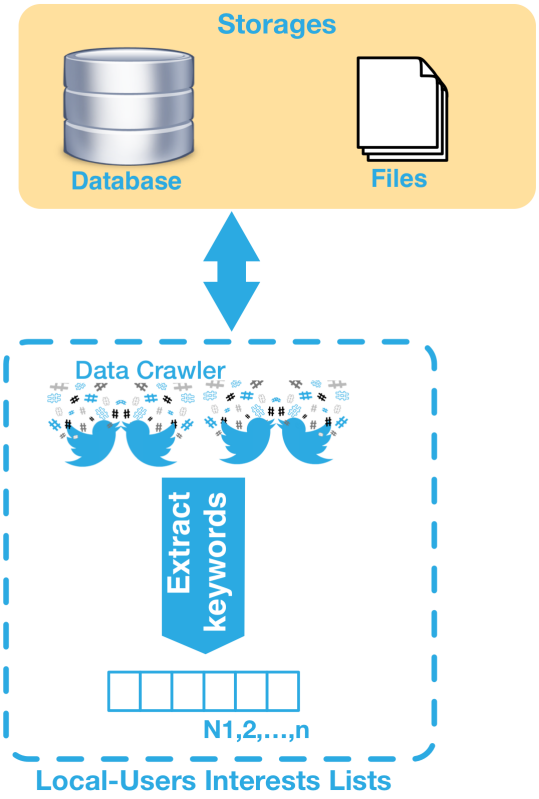


Figure 3.6: *Local-Users Interest Lists*

50 English tweets by calling Twitter REST API and then receive a response in JSON format. This collection sample is kept small to not exceed the limitation of the API calling. We retrieve only English tweets that because the main language at the new location is English, where the system user moved. This data is decoded to become local tweets and these local tweets are stored in a Database.

3.4.2 Extract Keywords and Hashtags

Local tweets that are only in English language are retrieved. Figure 3.8 shows an example of local tweet in JSON format and it also shows the tweet's text and hashtags and how

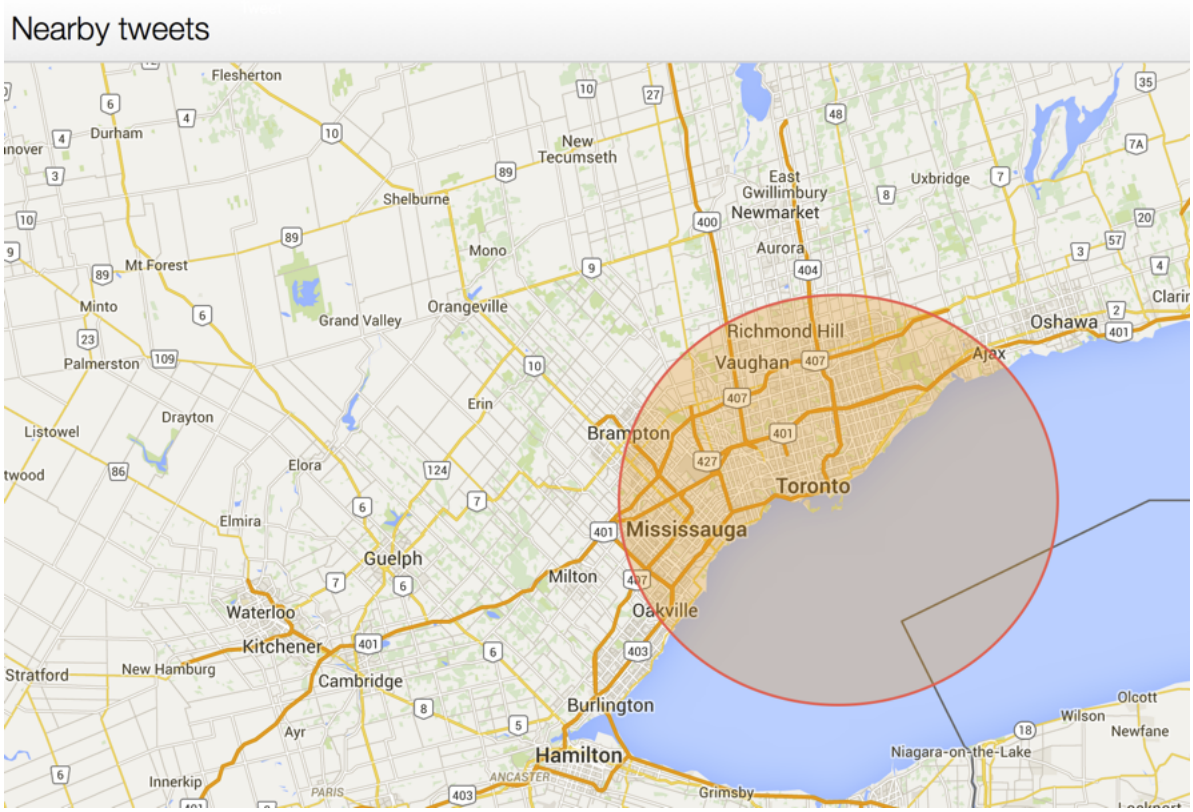


Figure 3.7: *User location Where the Radius is Equal 30 Km*

they are presented. By using algorithm 3, a filter is applied to the tweet text to get keywords by deleting stop words from a pre-compiled list and housed in the database. When called on, Twitter APIs provide hashtags for each tweet if they exist. Algorithm 4 extracts the tweet hashtags by decoding the JSON format.

To find the time complexity of algorithms 3 and 4, we calculate the total number of operations for each one. Since, the both algorithms are implemented to look for a term inside a tweets text, which means, we are using a for-loop to go through the whole tweets text. The worst time complexity for each one is:$T(n) = O(n)$. Where n is the length of the tweet's text.

Figure 3.8: *Example of Local Tweet in JSON Format and Shows Tweet Text and Hashtags list*

Algorithm 3: Delete Stop Words from Local Tweet Text

Input: Local Tweet Text
Output: Keywords

```
\\Retrieve the tweet's text and tokenize the text
      localTweetText= Tweets->text;
      wordsList[]= str_word_count(tweetText,1);

\\Retrieve all stopWords from Database and put them in a list
     query = " SELECT stopWord FROM StopWord ";
     stopWordList[] = oDB -> select(query);}

\\Remove any word in wordlist[] that is exist
   in stopWordList[]

    for(i=0;i<count(wordsList[]);i++)
          {
              if(in_array(wordsList[i], stopWordList))
                   wordsList[i]="";
       }

   return wordsList[];
```

Algorithm 4: Extract Hashtags and Insert them in Database

Input: Local Tweet Text in JSON format
Output: Extract Hashtags

```
// here this code below is about to find out
    the hashtag and insert them into a user table.

    foreach (entities->hashtags as hashtag) {

            hashtag = hashtag->text; \\decode JSON format.

        \\Insert the hashtag in system database
          insert(tweet_id,user_id,hashtag)
         }
```

3.4.3 Create Local Twitter Users Interest Lists

Each local twitter user has their own interest list, which is formed by gathering all of the keywords (nouns,and hashtags) from their recent 50 English tweets. Table 3.5 shows an example of one tweet for a specific local user 400419866. Also it shows the keywords that are retrieved by using algorithms 3 and 4.

To clarify the above processes, consider the following sample tweet:

"*I was going to print out this #3D loaf of #bread for dinner but scissors not large enough to do slicing. #technology*"

After the tokenization and removal of stop words, the following set of keywords are extracted from the above shown tweet:

{'#3D', 'loaf ', '#bread', 'dinner', 'scissors', '#technology' }

user_id	tweet_id	IsItHashtag	Keywords
400419867	565001228481212417	1	BREAD
400419867	565001228481212417	1	TECHNOLOGY
400419867	565001228481212417	1	3D
400419867	565001228481212417	0	LOAF
400419867	565001228481212417	0	SCISSORS
400419867	565001228481212417	0	DINNER

Table 3.5: Example of Creating Local-User Interest List

3.5 Measuring Similarity

Measuring similarity (Figure 3.9) is a sub process that works to calculate a match between the system's user "U" and each of the local Twitter users "N1,N2,N3,...Nn". This thesis uses a commonly used text mining and information retrieval process called Cosine Similarity as a method to find how our user is similar to local users.

For all users; system's user and local users; words vector is created from their interests'

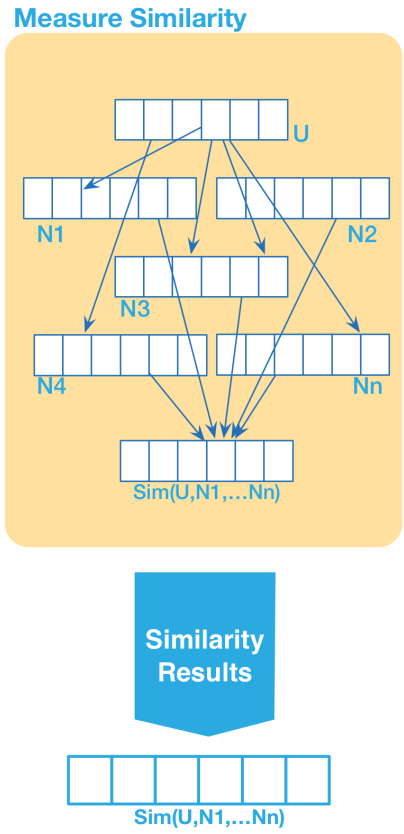


Figure 3.9: *Measure the Similarity between the System User and Local Twitter Users*

lists. To create the words vector, we assign to each keyword in the interest list a weight for that keyword. That depends on the number of occurrences of the keyword in the interest list. This weighting scheme is referred to as Term Frequency (TF).

Equation 3.1 calculates the cosine similarity. It measures the cosine of the angle between these two vectors. "A" represents the system user words vector. "B" represents words vector for a local user.

$$SimScore = cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{3.1}$$

This cosine score will always be between 0 and 1. A high cosine score value indicates that vector "A" and "B" are similar to each other. Algorithm 5 shows how the similarity measurement is preformed.It is obvious in algorithm 5 that there are two loops with iterations for each. Also, there is a calculate similarity score function. Since the calculate similarity score function is built to calculate cosine similarity between two vectors then the time complexity is $O(mn)$ [39]. Where n and m are represent the size of the two vectors. Therefore, the time complexity of algorithm 5 is $O(n(n+m))$.

Algorithm 5: Calculate-Similarity

```
LocalUsers = Get total number of local users in the dataset

for ( i = 0; i < LocalUsers; i++) {
 tweets = Get tweets of localUser[i]
 Compose localUsersInterestlist[i] using tweets of localUser[i]
 Create localUsersVector[i] of localUsersInterestslist[i]
}

\\ For the system user, whom want to recommend
UserTweets= Get tweets;
Compose UserInterestslist[] using UserTweets
Create UserVector[] of UserInterestslist[]


for (j = 0; j < LocalUsers; j++)
  {
  similarity-score = Calculate Similarity Score(UserVector[],
   localUsersVector[j]) \\using Equation 3.1
  }

save similarity-score in Database
```

System-User	Local-User	Score
172730866	43487238	0.345540399
528380701	57359038	0.399737696
446905360	65446278	0.313523986
332700437	2786329706	0.385846172
409651107	2909321087	0.659111926
382647359	624905423	0.338764752
110648233	43487238	0.643896536

Table 3.6: Snapshot of Personal Recommendations

3.6 Personal Recommendations

Based on the user taste, the system recommends the top 10 or 20 similarity scores between the user and local users. Table 3.6 shows an example of personal results.

Figure 3.10: *Community Interests*

3.7 General Recommendations

Extracting keywords for all local users and using Alchemy API[8] to classify these words helps the system to show the community interests. Cloud of words[9] is used to visualize the result.

3.8 Summary

In this chapter we explained the methodologies of our work in depth. We gave a small introduction of our recommender system architecture and describe its components. We explained how was the data collected and the filtering algorithms. Also how was the recommendations generated.

[8]http://www.alchemyapi.com
[9]http://www.wordle.net/

Chapter 4

Experiment

This chapter explains how the proposed system is implemented. Moreover, the experiment performed shows the precision of the proposed recommender system. We start by showing the experiment design, followed by the explanation of how the data was gathered and how the recommender system was built and executed. The chapter finishes with a demonstration of the experiment results and an examination and explanation of the final results.

4.1 Experiment Design

The main purpose of our recommender system is to recommend Twitter followees (local Twitter users) to our users, who just moved to a new place where the language and culture are different. We implemented the experiment to prove the precision of our proposed recommender system.

The proposed recommender system in this thesis is divided to several modules. To start generating recommendations the system needs input data. The input data is provided by Data Crawler module. The first step is to implement the Data Crawler.

Data Crawler downloads all required data from Twitter and stores it in our local

machine database. This Input data contains our users and local Twitter users tweets and other information such as followers, followees, user profile basic information, hashtags and urls. Input data is collected directly from Twitter APIs [40]. As Twitter APIs have some limitations for each call and the number of calling we can do per session, we run Data Crawler module for 2 weeks from January 1st to 15th, 2015.

Other modules, which are mentioned in chapter 3 such as extract keywords and hashtags, translation and measure similarity, are implemented as well. Finally, the proposed system gives some recommendations (local followees to follow) to our users based on his/her interests.

4.2 Dataset

In order to evaluate our proposed recommender system, it was important to use a proper data set. Since our system generates recommendations to Twitter users and these recommendations are followees (Friends) to follow, required data is collected from Twitter servers by using Twitter APIs in our Data Crawler module. As explained in chapter 3, users' physical location is needed to start collecting the dataset that is needed to recommend local followees who share almost the same interests to our users. In this thesis report it is assumed that system's users moved from the Middle East (Saudi Arabia); where the language spoken there is Arabic, to North America (Canada) where the language spoken is English.

The dataset used in this experiment is divided to three sub datasets:

1- Users' Dataset.

2- Local Twitter Users Dataset.

4.2.1 Users' Dataset

The Users' Dataset has data of Twitter users who tweets most of the time in Arabic and have moved or will be moving to Canada, (new location) to live and adapt to a new culture. This dataset contains 16 Twitter users, has 16 documents, one for each user. These Twitter users were selected from a group of Arabic speaking people found in our social circle. Each document represents the last 200 Arabic tweets for each user. Table 4.1 shows example of Users' dataset properties.

User_id	**Tweets**	**Keywords**
172730866	200	1241
711269720	200	1870
110648233	200	2520

Table 4.1: Users' Dataset Snapshot

4.2.2 Local Twitter Users Dataset

The Local Twitter Users Dataset has 1800 Twitter users who were selected randomly, and they are located in and tweet from Toronto, Canada, where system user has moved to or will relocate to in the future. This is explained in further detail in the implementation section. This thesis report will refer to these Twitter users as Local users or Local Twitter users. For each local user a document is created which has their last 50 English tweets and keywords are extracted from these tweets. See table 4.2 for an example of Local Twitter Users Dataset.

User_id	Tweets	Keywords
680703	50	457
940241	50	406
1979461	50	374
5966412	50	445
6475812	50	496
7076142	50	65
7290652	50	488
7380532	50	252
7477172	50	489
7590112	50	385

Table 4.2: Snapshot of 10 Local Users from Local Twitter Users Dataset

4.3 Implementation

This section discusses how our proposed system is implemented. The system has four main modules: Data Crawler, Extract Keywords, Translation, and Measure Similarity. PHP is used to implement and write the code of our system's modules. PHP is a powerful programming language and it has a variety of built in functions, which makes coding an easy job. Beside the power of PHP, Twitter, Google Maps, and Bing Translator APIs provide their libraries and services in PHP. The MAMP [41] free software bundle (see table 4.3) is used to run our system. Also we used AptanaStudio3 [1] free editor to write our code.

Operating system	Mac OS X
Web server	Apache
Database management system	MySQL
Web development	PHP

Table 4.3: MAMP Software Bundle

The first module we implemented was Data Crawler. Data Crawler has two sub modules,

[1] http://www.aptana.com/

one for system's User and the other for the Local users. Data Crawler user collects system's users Twitter data and it is implemented to call Twitter REST API and retrieve the last 200 tweets for each user.

Data Crawler Local collects local user Twitter data [42]. To start collecting local users' tweets the location information (Latitude and Longitude of system user's location) is needed; which is provided by algorithm 1 in Chapter 3. In this stage, Data Crawler Local calls Twitter Stream API to know who is tweeting in English at that location. For each tweet received from calling Stream API, Data Crawler Local finds the owner for that tweet and calls Twitter REST API to retrieve the last 50 tweets for that owner.

Data Crawler prepares the data and stores them in MySQL database. MySQL Community Edition is an open source free database and it is provided under the GPL License [2]. The system's database has different tables to store system users' and local users social network data. Here are the most useful tables in the thesis: MyUsers, LocalUser, MyUsers tweets, Tweets urls, Tweet mentions, and Tweet tag.

As soon as enough Twitter data is available in our local machine database for both our users and local users, the Extract Keywords module is activated. In this module we also have two sub modules that divide and process the tweets into the two languages, Arabic and English. Again, we refer to people who moved from Middle East to Canada and they tweet only in Arabic and will be called "System User". People who tweet in English from Toronto, Canada will be called "Local User".

Arabic Extract Keywords (AEK) is implemented to find all nouns for our users tweets. AEK uses Stanford's Arabic Parser (SAP) [37] tool to tokenize Arabic text based on parts of speech tagging technique. SAP is implemented in Java. To use this tool inside AEK we need to retrieve the tweets text from our system database and store them in a text

[2]http://www.mysql.com/about/legal/licensing/oem/

file. Following this, the file is sent as text file to SAP, then SAP is run and results are compiled into a coded text file as seen in figure 4.1. AEK reads the coded text file and extracts the tagged words from table 4.4. AEK finds the keywords for each Arabic tweet and stores them in the database.

Figure 4.1: *Snapshot for Arabic Text Coded by Stanford's Arabic Parser*

POS tag	Description	Example
NN	noun, singular or mass	door
NNS	noun plural	doors
JJ	adjective	big
JJR	adjective, comparative	bigger
NNP	proper noun, singular	John
DT	determiner	the

Table 4.4: Part of Speech Tagging Code Meaning

English Extract Keywords (EEK) is implemented to find all keywords for local users tweets. EEK takes each tweet for local users and deletes white-spaces, urls, and stop words. In our database we have a list of most common stop words in English stored in a table and it has 710 words. EEK stores local users tweets keywords in the database.

After AEK extracts all keyword for each Arabic tweet it stores them in a database. The Translation module retrieves all Arabic keywords from this system database and translates them to English by using Bing translator API. Before the translation function calls Bing translator API, the spell check function is used to correct misspelled Arabic

words. We create our own updated list of most common misspelled words in Arabic. Then we call Bing translator API to translate the words. After getting the words meaning in English we delete the white-spaces and stop words then store the English words in the database. After AEK and EEK, interest lists are created for both local users and system's users by counting the term frequency for each keyword.

The proposed recommender system is built to recommend a list of local Twitter users to be followed by the people who moved to a new location, where the language spoken at the new location is different. These previously created interest lists are used to find the similarity between the system's users and local users. Based on the similarity score the proposed system generates a recommendation list.

We implemented this system to gather a list of tweets contents for each Twitter user (local users and system user) and consider it as a document. This document is filtered by AEK or EEK as mentioned above. Then we generate the words vector [43] for each document, which has the term frequency for each keyword. Finally, using cosine similarity to measure the similarity between these Twitter users (local users and system user). Based on the similarity score between the user and all local users, the proposed system sorts the local users descending and recommends the top 10 local users to follow.

The configuration of the system that we used to implement our proposed system is shown in 4.5.

CPU	2.6 GHz Intel Core i7
RAM	16 GB
Operating System	OSX version 10.10.4
Programming Languages	PHP and Java
DataBase	MySQL 5.5

Table 4.5: System Configuration

4.4 Runtime

The total runtime of the system depends on multiple factors including:

1. Twitter API calling and responding.
2. Bing Translator API calling and responding.
3. String Processing (Parts of Speech Tagging for Arabic tweets and Count Term Frequency TF).
4. Similarity Measurement.

The amount of delays produced by the above factors is dependent on the number of tweets requested and processed. Table 4.6 shows the average run-time to find the similarity between user and local users.

Average time of retrieving 200 tweets for specific Twitter user by calling Twitter API	3.343 seconds
Average time of Parts of Speech Tagging for one Arabic tweet	0.4 seconds
Average time of Extracting Keywords for one tweet	1.462 seconds
Average time of Translating one word by using Bing translator API	0.3023 seconds
Average time of Counting Term Frequency (TF) for one user who has 200 tweets	5.0067901611E-6 seconds
Average time of measuring the similarity between a system user and a local user	6.3896179199E-5 seconds
Total	5.50741 seconds

Table 4.6: Average Time of Performing finding the similarity between 2 Twitter Users

4.5 Result and Analysis

Recommender systems generate a list of candidate recommendations and these recommendations are sorted and ranked by special algorithms as mentioned in chapter 2. So

far, we have presented our method of recommending local Twitter users to follow. In this section, we aim to describe the technique we exploit to assess this recommendation approach. As started earlier in this Chapter, the recommend function is applied to compare every system's user against all the existing local users in the repository and sort the existing local users based on their similarity to the system's user. Since the list of candidates is sorted in descending order based on the similarity measure, the most similar local users are expected to be ranked higher in the resulting list. Table 4.7 shows the top 10 recommended local users to follow for system's user "110648233".

Top 10 Local-Users	**Similarity Score**
43487238	0.643896536
244706026	0.642127294
1979461	0.60468874
65446278	0.590830644
62322761	0.504725807
127235443	0.475754352
57359038	0.466995084
81518458	0.450044457
2822596970	0.446523167
2502391231	0.413901234

Table 4.7: Top 10 Recommendations for "110648233"

To evaluate the ranked list of recommendations, we have applied the Mean Average Precision (MAP) measure [44]. MAP is a single-figure measure of ranked retrieval results independent from the size of the top list. It is designed for general ranked retrieval problem, where a query can have multiple relevant answers. In our proposed recommender system, each system's user would get a sorted list of recommended local users to follow. In this experiment, we focus only on the top 10 candidate recommendations that are given by the system. The MAP measure is calculated as follows:

$$MAP = \frac{\sum_{q=1}^{Q} AvgP(q)}{Q} \tag{4.1}$$

$$AvgP(q) = \frac{\sum_{k=1}^{n} P(k) \times (Rel(k))}{number - of - relevant - recommendations} \tag{4.2}$$

In the above functions, Q is a set of queries which represents the system's users and $AvgP(q)$ is the Average Precision for query q. Relevant-recommendations are the actual local users selected by the system's user to follow; $Rel(k)$= 1 when the recommendation is hit by the system's user and $Rel(k)$= 0 otherwise; n is 10 which represents the Top 10 candidate recommendations generated by the system ; and $P(k)$ is the precision at the cut-off k. The precision function is provided below:

$$Precision = \frac{|\{Relevant - Recommendations\} \bigcap \{Retrieved - Recommendations\}|}{\{Relevant - Recommendations\}} \tag{4.3}$$

In our experiments we need to evaluate our list of candidate recommendations and we need to know the relevant-recommendations for each system user. Therefore, every system's user is asked to select relevant local users from the top 10 list that is generated by the proposed system for that user. In our experiment, we have 16 system's users in the users dataset and 1800 local users in the local Twitter users dataset, as showed in table 4.8. The system generates 1800 candidate recommendations for each system's user

in descending order based on similarity score between the system's user and the local users. Based on the surveys' feedback, we determined the relevant-recommendations for each user. Table 4.9 shows the local users are selected to be followed by the system's user "110648233".

Dataset	**Description**	**Number of Twitter Users**
Users' Dataset	System users who moved to a new location	16
Local Users Dataset	Local Twitter users at the new location	1800

Table 4.8: Snapshot of our Datasets

Local Users	**Selected**
43487238	NO
244706026	YES
65446278	NO
62322761	NO
127235443	YES
57359038	NO
81518458	YES
2822596970	NO
2502391231	NO
371944107	NO

Table 4.9: Selection of Local Users to Follow by System's User "110648233"

To calculate the MAP, we need to have $AvgP(q)$ for each query. Since we have 16 system's users for this experiment and they present Q at Equation 4.1, we calculated $AvgP(q)$ for each one. Thus, according to Equation 4.2, the resulting average precision for system's user "110648233" and "1170925825" are as follows:

$$AvgP(\text{"110648233"}) = \frac{\sum_{k=1}^{10} P(k) \times (Rel(K))}{3} = \frac{\frac{1}{2} + \frac{2}{5} + \frac{3}{7}}{3} = 0.445$$

$$AvgP(\text{"1170925825"}) = \frac{\sum_{k=1}^{10} P(k) \times (Rel(K))}{5} = \frac{\frac{1}{1} + \frac{2}{3} + \frac{3}{5} + \frac{4}{8} + \frac{5}{10}}{5} = 0.654$$

In the above functions, we divided the summation of precisions for the system's user "110648233" by 3, because "110648233" selected 3 out of 10 when he did the survey as showed in table 4.9. While system's user "1170925825"'s the summation of precisions was divided by 5 because he selected 5 out 10 in his survey feedback. Table 4.10 shows "1170925825" selection.

Local Users	**Selected**
107270466	Yes
81518458	No
247477005	YES
558290910	NO
67327726	YES
32731483	NO
1052065470	NO
565648025	YES
2340652894	NO
511331825	YES

Table 4.10: Selection of Local Users to Follow by System's User "1170925825"

In the same way, we calculated $AvgP(q)$ for other system's users. Figure 4.2 shows the average precision for all system's users in the users dataset.

Consequently, the MAP value is calculated for the proposed recommender system by applying Equation 4.1 and it is 0.52242 which is the mean of $AvgPs$. MAP returns values from 0 to 1. Higher values for the MAP imply the better performance of the proposed recommender system.

$$MAP = \frac{8.3587}{16} = 0.52242$$

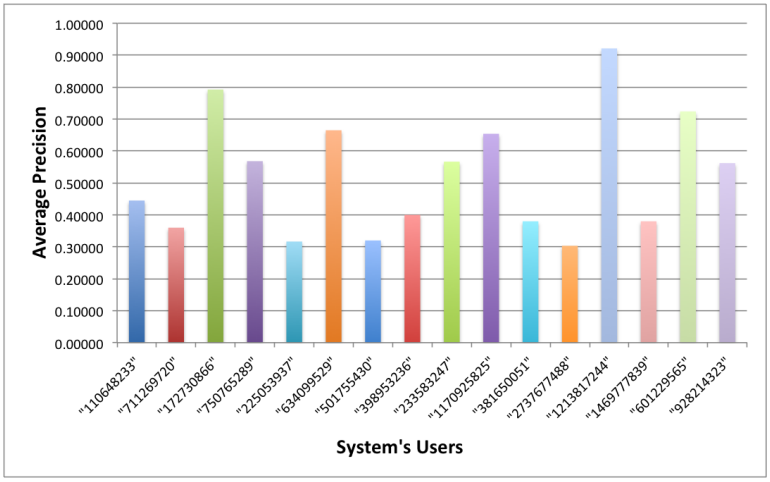


Figure 4.2: *Average Precisions for all System's users*

4.6 Summary

In this chapter, we provided a brief introduction about this experiment. We also explained how this experiment was designed, how data was used and how it was divided into two datasets, how the data explained and implemented with details about how the system was created and how it works. Finally, the results were evaluated by comparing the outcomes between our user and his/her friends.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Recommender Systems have been used in many sectors and in many applications. The main idea of recommender system is to generate recommendations, which help the user to make decisions. These recommendations are created and ranked by filtering techniques.

This thesis report has discussed and explained how the recommender system was implemented to generate ranked recommendations for Twitter users who moved to a new location. Moreover, this report evaluated these recommendations and showed the effectiveness of the system. These recommendations are a list of local Twitter users to follow.

The recommender system has two data inputs. First, our user's data, which is the last 200 tweets in Arabic. Second, local users' data, which is the last 50 tweets in English for each local user. For each user, local or our user, the interest list is found and compiled individually by extracting keywords from his/her tweets. For our user only, the interests list is translated to English.

TF was used to create the word vectors for each user. In this stage, the system has our user's word vector and the word vector for each local user. The cosine similarity

measurement was applied to calculate the similarity between our user's word vector and the local users' word vectors separately. Based on the similarity scores the recommender system finds the top 10 similarity scores and recommends these local Twitter users to follow.

To find the effectiveness of the recommender system, we gathered 16 twitter users who tweet in Arabic in a group. For each one of these 16 we run the proposed recommender system and generate a top 10 candidate recommendations list. These lists are sent out by email to each system's user and they select the ones they want to follow. Based on their selection we calculate the average precision for each system's user and use this to find the mean average precision (MAP) for the proposed recommender system.

5.2 Future Work

We would like to keep working on our recommender system to improve its proficiency and productivity of recommending local Twitter followees.

First, we would like to improve our way of extracting keywords from tweets in both languages Arabic and English. Currently we use Stanford Arabic Parser tool to extract Arabic keywords, we would like to create our own tool for that and not only for Arabic but for English as well.

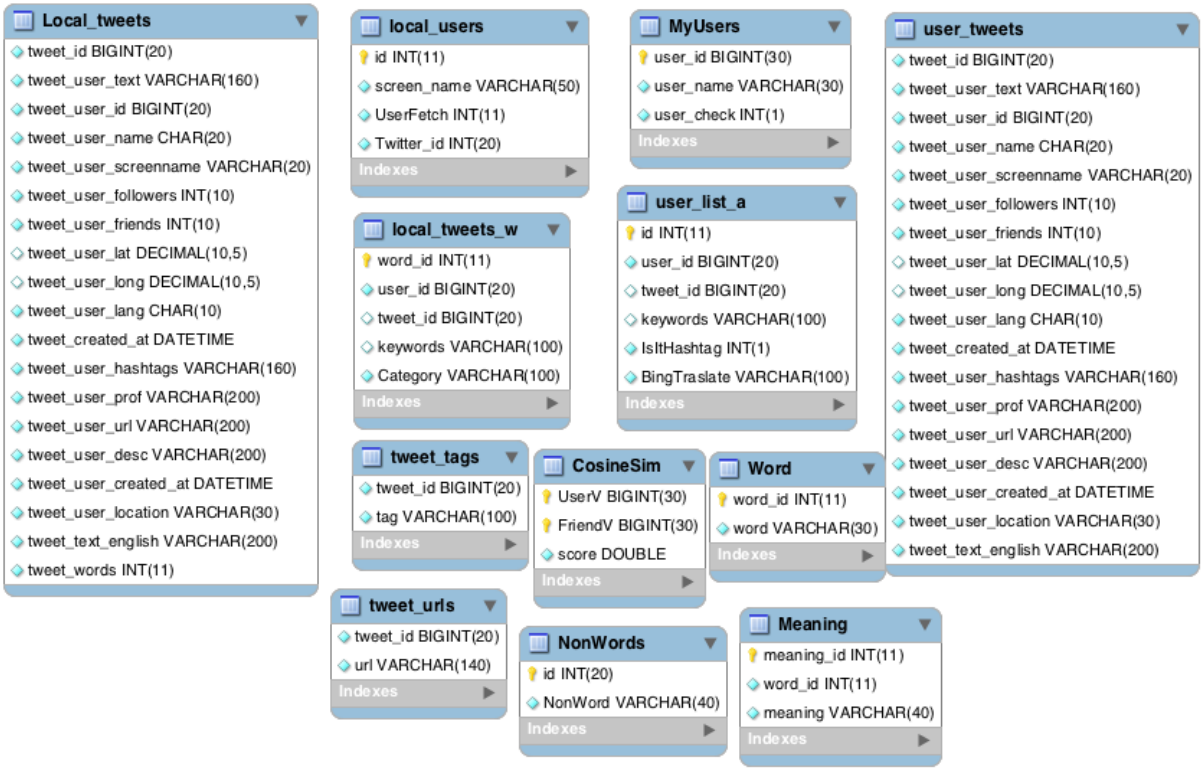
Second, we would like to improve our spell checking for Arabic by creating a tool to predict the correction if we could not find the correction in our dictionary.

Third, we would like to use more translators to find the best translation. In Addition, we would like to find the related words too. Twitter now provides translation for tweets. We checked Twitter translation for some tweets in Arabic and the translation was not efficient yet.

Finally, We would like to categorize the keywords and use the category to find the similarity between tweets.

Appendix A

List of Database Tables



System Database Tables

Appendix B

List of Stop-Words

a	beings	face	Higher	many	opening	see	though
about	best	faces	Highest	may	Opens	seem	thought
above	better	fact	Him	me	Or	seemed	thoughts
across	between	facts	Himself	member	Order	seeming	three
after	big	far	His	members	ordered	seems	through
again	both	felt	How	men	ordering	sees	thus
against	but	few	however	might	orders	several	to
all	by	find	i	more	Other	shall	today
almost	c	finds	if	most	Others	she	together
alone	came	first	important	mostly	Our	should	too
along	can	for	in	mr	Out	show	took
already	cannot	four	interest	mrs	Over	showed	toward
also	case	from	interested	much	P	showing	turn
although	cases	full	interesting	must	Part	shows	turned
always	certain	fully	interests	my	Parted	side	turning
among	certainly	further	into	myself	parting	sides	turns
an	clear	furthered	is	n	Parts	since	two
and	clearly	furthering	it	necessary	per	small	u
another	come	furthers	its	need	perhaps	smaller	under
any	could	g	itself	needed	place	smallest	until
anybody	d	Gave	j	needing	places	so	up
anyone	did	general	just	needs	point	some	upon

Table B.1: List of Stop-words

anything	differ	generally	k	never	pointed	somebody	us
anywhere	different	get	keep	new	pointing	someone	use
are	differently	gets	keeps	newer	points	something	used
area	do	give	kind	newest	possible	somewhere	uses
areas	does	given	knew	next	present	state	v
around	done	gives	know	no	presented	states	very
as	down	go	known	nobody	presenting	still	w
ask	downed	going	knows	non	presents	such	want
asked	downing	good	l	noone	problem	sure	wanted
asking	downs	goods	large	not	problems	t	wanting
asks	during	got	largely	nothing	put	take	wants
at	e	great	last	Now	puts	taken	was
away	each	Greater	later	nowhere	q	than	way
b	early	Greatest	latest	number	quite	that	ways
back	either	group	least	numbers	r	the	we
backed	ended	grouped	less	o	rather	their	well
backing	ending	grouping	let	of	really	them	wells
backs	ends	groups	lets	off	right	then	went
be	enough	h	like	often	room	there	were
became	even	had	likely	old	rooms	therefore	what
because	evenly	has	long	older	s	these	when
become	ever	have	longer	oldest	said	they	where
becomes	every	having	longest	on	same	thing	whether
been	everybody	he	m	once	saw	things	which
before	everyone	her	made	one	say	think	while
began	everything	here	make	only	says	thinks	who
behind	everywhere	herself	making	open	Second	this	whole
being	F	high	man	opened	Seconds	those	whose
why	within	worked	would	year	year	you	youngest
will	without	working	x	would	years	young	your
with	work	works	y	z	yet	younger	yours

Table B.2: List of Stop-words

Appendix C

List of None Correct Arabic Words

ايفون	ابراهيم	أرتفاع	اطباء	انذاك	احيانا	أعتقاد	الازهر
أيفون	ابنائه	أستثمار	اطفال	انسان	أخراج	أعتماد	الاسلام
جلبريك	ابوه	أستخلاص	اعلانات	أنسان	اخراج	أعتناء	الاشعة
تيونز	ابي	أستدراج	اعوام	أنصرف	اخرى	أغتيال	الاف
إتهام	اتاح	إستراد	اقامة	أنطلاق	اخرين	أفتتاح	الاصدار
أتهام	اثارة	أسترداد	افلام	إنطلاق	ادب	افريقيا	الاعمال
أستخدام	اثناء	أستفتاء	افراح	أنظر	اذا	أكتشاف	الافران
إستخدام	اجباري	أستقبال	اقسام	أهتمام	أذا	إلتحاق	القران
اسعاف	اجبر	أستقرار	اقطار	أبنته	اربع	أمتداد	المنيوم
سكندرية	اجبرت	أستمرار	اكسيد	أبنه	اربعة	أمتلاك	الواح
سكندرية	اجزاء	أستنكار	الإعتراض	أتجاه	ارشيف	أنتظام	الوان
أسم	اجل	اسرائيلية	الألات	أتصال	ارقام	أنتقاد	امبراطور
اسيا	اجنبية	أضافة	الامكان	أتفاق	اساس	أنحلال	اميرة
اسيوي	احد	أعتبر	الإتصال	أجتماع	اساليب	أنخفاض	اموال
أشتهر	احداث	أعتداء	الأستيطان	أحتل	اسامه	أندثر	انتاجها
اعلان	احدى	إعترض	الأشتراكي	أحتياط	اسلامية	أنسحاب	انترنت
أعلان	احزاب	أعتراض	الأعتداء	أحتياطات	اصحاب	أنصهار	اهرامات
الهة	احفاد	أعتراف	الأنشاء	أختصار	اصلي	أنضم	ربحي
امريكي	احمد	أعتزاز	الاجنبي	أختيار	اصول	أنفصال	اكسبريا
أنتقل	احياء	أعتزال	الارضي	أدارة	اضرحة	إنقلاب	وتساب
ائمة	اوفيس	إنهيار	واتس اب	أهتمامات	إوفيس	إنكمش	واتساب

Appendix D

List of Arabic Correct Words

آيفون	اهتمام	استقبال	امتداد	أبوه	أخرى	أفكار	الإسلام
جيلبريك	ابنته	استقرار	امتلاك	أبي	آخرين	إقامة	الأشعة
لآي تيونز	ابنه	استمرار	انتظام	أتاح	أدب	أفلام	آلاف
اتهام	اتجاه	استنكار	انتقاد	إثارة	إذا	أفراح	الإصدار
استخدام	اتصال	إسرائيلية	انتهاك	أثناء	أربع	أقسام	الأعمال
إسعاف	اتفاق	اضافة	انحلال	إجباري	أربعة	أقطار	الأفران
سكندرية	اجتماع	اعتبر	انخفاض	أجبر	أرشيف	أكسيد	القرآن
اسم	احتل	اعتداء	اندثر	أجبرت	أرقام	الاعتراض	ألمنيوم
آسيا	احتياط	اعتراض	انسحاب	أجزاء	أساس	الآلات	ألواح
آسيوي	احتياطات	اعتراف	انصهار	أجل	أساليب	الإمكان	ألوان
اشتهر	اختصار	اعتزاز	انضم	أجنبية	أسامه	الإنجازات	إمبراطور
إعلان	اختيار	اعتزال	انفصال	أحد	إسلامية	الاتصال	أميرة
آلهة	إدارة	اعتقاد	انقلاب	أحداث	أصحاب	الاستيطان	أموال
أمريكي	ارتفاع	اعتماد	انكمش	إحدى	أصلي	الاشتراكي	إنتاجها
انتقل	استثمار	اعتناء	انهيار	أحزاب	أصول	الاعتداء	إنترنت
آنذاك	استخلاص	اغتيال	اهتمامات	أحفاد	أضرحة	الانشاء	أهرامات
إنسان	استدراج	افتتاح	أئمة	أحمد	أطباء	الأجنبي	ربح
انصرف	استراد	أفريقيا	أبحاث	أحياء	أطفال	الأرضي	اكسبيريا
انطلاق	استرداد	اكتشاف	إبراهيم	أحيانا	إعلانات	الأرقام	واتسآب
انظر	استفتاء	التحاق	أبنائه	إخراج	أعوام	الأزهر	أوفيس

Appendix E

System's Users Survey Feedback

System's User	Local Users	Selected
172730866	43487238	YES
	81518458	YES
	244706026	NO
	65446278	NO
	300430734	YES
	47539078	NO
	374610201	YES
	393786012	NO
	39356875	NO
	2502391231	NO
711269720	244706026	NO
	65446278	YES
	62322761	NO
	43487238	NO
	1104395136	NO
	393568758	NO
	149962301	NO
	248104981	YES
	680703	NO
	37367449	NO

Table E.1: Selections of Local Users to Follow by System's Users "172730866" and "711269720"

System's Users	Local Users	Selected
750765289	2993090626	YES
	200090279	NO
	57359038	NO
	2797897450	NO
	127235443	NO
	244706026	YES
	65446278	YES
	102536324	YES
	295352197	YES
	38294150	YES
225053937	107270466	NO
	431899935	NO
	2797897450	NO
	1038681468	YES
	65446278	YES
	1610021376	NO
	85939459	NO
	329379159	NO
	1052065470	NO
	2909318705	YES
634099529	71933941	YES
	2147872199	NO
	2907334617	NO
	57359038	NO
	140881741	NO
	107270466	YES
	69499632	NO
	423126635	NO
	2336049481	NO
	81518458	NO

Table E.2: Selections of Local Users to Follow by System's Users "750765289","225053937" and "634099529"

System's Users	Local Users	Selected
501755430	65446278	NO
	244706026	NO
	270138861	NO
	223013289	NO
	2736677813	YES
	429158373	YES
	92433007	YES
	2993090626	NO
	57359038	NO
	143467917	NO
398953236	145686535	NO
	102536324	YES
	225553381	YES
	335672978	NO
	212211701	NO
	329379159	NO
	414239929	NO
	145746917	NO
	576915389	NO
	322332216	YES
233583247	750418266	NO
	1052065470	YES
	107270466	YES
	144301261	NO
	30900299	YES
	139280902	NO
	132715398	YES
	2278254764	NO
	196374488	YES
	863989141	YES

Table E.3: Selections of Local Users to Follow by System's Users "501755430","398953236" and "233583247"

System's Users	Local Users	Selected
381650051	414239929	NO
	11270742	NO
	2336049481	NO
	322332216	NO
	306246406	YES
	764331150	YES
	35695680	YES
	2781923903	NO
	69499632	YES
	81518458	YES
2737677488	270138861	NO
	244706026	NO
	300430734	NO
	65446278	YES
	2502391231	NO
	62322761	YES
	383450629	NO
	57359038	NO
	81518458	YES
	127235443	NO
1213817244	863989141	YES
	93776474	YES
	750418266	YES
	107270466	YES
	15384497	NO
	30900299	YES
	270689079	YES
	357231302	YES
	491079459	NO
	67327726	YES

Table E.4: Selections of Local Users to Follow by System's Users "381650051","2737677488" and "1213817244"

System's Users	Local Users	Selected
1469777839	107270466	NO
	81518458	YES
	2993090626	NO
	132715398	NO
	300430734	NO
	109337879	NO
	2336049481	YES
	198660882	NO
	32731483	YES
	92433007	YES
601229565	15384497	YES
	107270466	YES
	225553381	NO
	38294150	YES
	81518458	NO
	325907296	NO
	15238697	YES
	79051698	NO
	2909318705	YES
	244706026	NO
928214323	107270466	NO
	144301261	NO
	65446278	YES
	81518458	YES
	67327726	YES
	2784431343	YES
	15384497	YES
	431899935	NO
	208798544	NO
	30900299	NO

Table E.5: Selections of Local Users to Follow by System's Users "1469777839","601229565" and "928214323"

Appendix F

Publications

- Mubarak Alrashoud, M. AlMeshary and A. Abhari,"Automatic Validation for Multi Criteria Decision Making Models in Simulation Environments", Proceedings of 18th Communications and Networking Simulation Symposium (CNS15/ACM), Alexandria, VA, US., April 2015.

- Meshary AlMeshary and A. Abhari, "A Recommendation System for Twitter Users in The Same Neighborhood", Proceedings of 16th Communications and Networking Simulation Symposium (CNS13/ACM), San Diego, California, US., April 2013.

- Poornima Prabhu, M. AlMeshary and A. Abhari, "Information Retrieval in Web 2.0- Role of Tagging and Folksonomies", Proceedings of 16th Communications and Networking Simulation Symposium (CNS13/ACM), San Diego, California, US., April 2013.

- Shakira Kaleel, M. AlMeshary and A. Abhari, "Event Detection and Trending in Multiple Social Networking Sites ", Proceedings of 16th Communications and Networking Simulation Symposium (CNS13/ACM), San Diego, California, US., April 2013.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.

[2] S. Berry, S. Fazzio, Y. Zhou, B. Scott, and L. Francisco-Revilla, "Netflix recommendations for groups," *Proceedings of the American Society for Information Science and Technology*, vol. 47, no. 1, pp. 1–3, 2010.

[3] C. C. Aggarwal, ed., *Social Network Data Analytics*, ch. An Introduction to Social Network Data Analytics, pp. 1–15. Hawthorne, NY 10532, USA: Springer, 2011.

[4] D. J. Hughes, M. Rowe, M. Batey, and A. Lee, "A tale of two sites: Twitter vs. facebook and the personality predictors of social media usage," *Computers in Human Behavior*, vol. 28, no. 2, pp. 561–569, 2012.

[5] Nielsen-Wire, "Led by facebook, twitter, global time spent on social media sites up 82% year over year, http://blog. nielsen. com/nielsenwire/global/led-by-facebook-twitter-globaltime-spent-onsocial-media-sites-up-82-year-over-year," Last visited: April 15, 2015.

[6] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pp. 56–65, ACM, 2007.

[7] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?," in *Proceedings of the 19th international conference on World wide web*, pp. 591–600, ACM, 2010.

[8] Twitter, "The twitter glossary, https://support.twitter.com/articles/166337-the-twitter-glossary," Last visited:April 15 2015.

[9] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy, "Make new friends, but keep the old: recommending people on social networking sites," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 201–210, ACM, 2009.

[10] B. Furht, *Handbook of Social Network Technologies and Applications*. No. 115-146, New York, NY, USA: Springer Science & Business Media, 1st ed., 2010.

[11] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[12] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[13] L. Candillier, F. Meyer, and M. Boullé, "Comparing state-of-the-art collaborative filtering systems," in *Machine Learning and Data Mining in Pattern Recognition*, pp. 548–562, Springer, 2007.

[14] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web*, pp. 291–324, Springer, 2007.

[15] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[16] C. Porcel, A. Tejeda-Lorente, M. Martínez, and E. Herrera-Viedma, "A hybrid recommender system for the selective dissemination of research resources in a technology transfer office," *Information Sciences*, vol. 184, no. 1, pp. 1–19, 2012.

[17] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[18] B. Krulwich, "Lifestyle finder: Intelligent user profiling using large-scale demographic data," *AI magazine*, vol. 18, no. 2, p. 37, 1997.

[19] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the 12th international conference on machine learning*, pp. 331–339, 1995.

[20] J. Salter and N. Antonopoulos, "Cinemascreen recommender agent: combining collaborative and content-based filtering," *Intelligent Systems, IEEE*, vol. 21, no. 1, pp. 35–41, 2006.

[21] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, pp. 47–56, 2000.

[22] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[23] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating "word of mouth"," pp. 210–217, 1995.

[24] W. W. C. Jianming He, ed., *Data Mining for Social Network Data*, vol. 12, ch. A Social Network-Based Recommender System (SNRS), pp. 47–74. Springer, 2010.

[25] S. Jurvetson, "What exactly is viral marketing," *Red Herring*, vol. 78, pp. 110–112, 2000.

[26] M. Bank and J. Franke, "Social networks as data source for recommendation systems," in *E-Commerce and Web Technologies*, pp. 49–60, Springer, 2010.

[27] N. Hu, L. Liu, and J. J. Zhang, "Do online reviews affect product sales? the role of reviewer characteristics and temporal effects," *Information Technology and Management*, vol. 9, no. 3, pp. 201–214, 2008.

[28] A. Ghose and P. G. Ipeirotis, "Designing novel review ranking systems: predicting the usefulness and impact of reviews," in *Proceedings of the ninth international conference on Electronic commerce*, pp. 303–310, ACM, 2007.

[29] N. Hu, P. A. Pavlou, and J. Zhang, "Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of online word-of-mouth communication," in *Proceedings of the 7th ACM conference on Electronic commerce*, pp. 324–330, ACM, 2006.

[30] J. Hannon, M. Bennett, and B. Smyth, "Recommending twitter users to follow using content and collaborative filtering approaches," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 199–206, ACM, 2010.

[31] H. B. Celebi and S. Uskudarli, "Content based microblogger recommendation," in *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pp. 605–610, IEEE, 2012.

[32] M. G. Armentano, D. L. Godoy, and A. A. Amandi, "A topology-based approach for followees recommendation in twitter," in *9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, p. 22, 2011.

[33] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in twitter," in *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, vol. 730, pp. 67–78, 2011.

[34] Y. Lai and J. Zeng, "A cross-language personalized recommendation model in digital libraries," *The Electronic Library*, vol. 31, no. 3, pp. 264–277, 2013.

[35] Google, "Goggle maps api, https://developers.google.com/maps/," Last visited: April 15, 2015.

[36] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 173–180, Association for Computational Linguistics, 2003.

[37] S. Green and C. D. Manning, "Better arabic parsing: Baselines, evaluations, and analysis," in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 394–402, Association for Computational Linguistics, 2010.

[38] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, "The penn arabic treebank: Building a large-scale annotated arabic corpus," in *NEMLAR conference on Arabic language resources and tools*, pp. 102–109, 2004.

[39] J. Y. Chung, B. Park, Y. J. Won, J. Strassner, and J. W. Hong, "An effective similarity metric for application traffic classification," in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pp. 286–292, IEEE, 2010.

[40] Twitter, "Twitter apis, https://dev.twitter.com/overview/documentation," Last visited: April 15, 2015.

[41] MAMP, "Mamp bundle download, https://www.mamp.info/en/," Last visited: April 15, 2015.

[42] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis, "Discovering geographical topics in the twitter stream," in *Proceedings of the 21st international conference on World Wide Web*, pp. 769–778, ACM, 2012.

[43] C. D. Manning, P. Raghavan, and H. Schütze, "Scoring, term weighting and the vector space model," *Introduction to Information Retrieval*, vol. 100, 2008.

[44] S. Teufel, "An overview of evaluation methods in trec ad hoc information retrieval and trec question answering," in *Evaluation of text and speech systems*, pp. 163–186, Springer, 2007.