AUTOMATIC IDENTIFICATION OF USER INTEREST FROM SOCIAL MEDIA

by

Mathavan Kumar

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
March 2015

This work is dedicated to

My supervisor and well-wisher Prof. Srinivas Sampalli

and

My beloved parents Mr.Kumar and Mrs. Mala for their unconditional support

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Automatic identification of user interest from social media has gained much attention in the recent years. In Twitter, users could post tweets about a wide range of topics. These tweets could be analyzed to identify the user's interests, which could be used to personalize recommendations for that user. But the short length of these tweets poses a huge challenge in classifying the tweets using traditional classification algorithms. In this thesis, a hybrid approach has been proposed to overcome this challenge. All tweets containing URLs are grouped as sessions with session duration as 1 hour, which increases the text length considerably. These sessions are then classified into 8 pre-defined categories using logistic regression. Based on the categories which appeared frequently in these sessions, top 3 categories are identified as the interests of the user. Experiments show that the proposed approach is able to identify the user interest in a precise manner.

LIST OF ABBREVIATIONS USED

API	Application Programming Interface
URL	Uniform Resource Locator
WH	Wikipedia Hierarchy
WCG	Wikipedia Category Graph
HIG	Hierarchy Interest Graph
REST	Representational State Transfer
OAUTH	Open Standard for Authorization
JSON	JavaScript Object Notation
OSN	Online Social Networks
AMT	Amazon Mechanical Turks
HIT	Human Intelligence Tasks
LDA	Latent Dirichlet Allocation
pLSI	Probabilistic Latent Semantic Analysis
L-LDA	Labeled Latent Dirichlet Allocation
NLTK	Natural Language Toolkit

ACKNOWLEDGEMENTS

In the journey that led to creation of the work presented in this thesis, I have been guided, helped and supported by many. First and foremost I would like to thank my supervisor Prof. Srinivas Sampalli. His dedication and enthusiasm towards research is truly inspirational. He has been an excellent mentor and guide throughout my Master's program. I would also like to thank all the members of the MyTech Lab for providing valuable suggestions for my research work. I would like to thank my team member Mounika for her constant support throughout my Master's program.

This acknowledgement would be incomplete without thanking my friends Sathish, Pradeep and Naganathan and my sister Anitha. They have been helping me continously in all the hard times.

I cannot put into words the gratitude I owe to my parents. They have constantly encouraged me to excel in whatever I do. All that I am , I owe to my parents.

CHAPTER 1 INTRODUCTION

Online social networks provide new ways to generate and consume information. Traditionally, people get information from either online news portals or blogs. But after the advent of social networking sites like Facebook, Twitter, Google+, etc., people started to receive information from these social networking sites. These sites are also used by people to share the information they have across the network. They share the events that happened in their own lives or events which they heard from someone else [36]. They share this information in the form of text or multimedia content. These user generated information on the social networking sites could be used to identify the interests of that user and these interests could be used to personalize the user experience [2].

In this thesis, Twitter has been chosen as a medium to collect information from the users. Twitter has been chosen due to the fact that most of the Twitter contents are public [2] and tweets are good indicators of user interests [37, 38]. Twitter has 288 million monthly active users and 500 million tweets are sent per day in Twitter [39]. The topics of these tweets range from personal status to opinions or reviews on a product. Twitter has a combination of features of both micro-blogging and social networking [7]. As a micro-blogging service, Twitter allows user to post 140 character messages called tweets. This feature makes the Twitter to have a large dataset of user generated text. These tweets can be broadcasted publicly and a social network of Twitter users can be set up by following another user's tweet [7].

This large amount of user generated text in Twitter attracted researchers to understand the human behavior to improve the quality of life. Twitter data have been used for discovering breaking news, detecting natural disasters or for analyzing political campaigns [7]. In this thesis, Twitter has been used to identify the user interests. Twitter stores a lot of information about a single user. Sensitive information like user names, tweets posted by the users, geographic location of the tweets, list of followers/friends are stored in Twitter. Some of this information could be retrieved using APIs provided by Twitter. If this user information is tapped correctly, then this information could be used to personalize recommendations for that user [40].

There are many challenges involved in using the Twitter data to identify the user interests. Tweets are short in length. Only 140 characters are allowed in tweets. So classification with the sparse data becomes a huge challenge. Another problem is the informal nature of the tweets. Users use abbreviations to overcome the limitation of the tweet length. Sometimes users use slang words in the tweets. Usage of these informal languages makes the vocabulary very large. Another issue is the constant changing vocabulary. If some new events occur, then the words relating to those events could gain popularity. These new words need to be updated in the classification models. So, static classification models will not be suitable for classifying the tweets [41].

1.1 BRIEF INTRODUCTION OF THE PROPOSED APPROACH

In this thesis, a hybrid approach has been proposed to overcome the challenges posed by the short length of the tweets. One or more tweets containing URLs which are posted within 1 hour time duration are grouped into sessions [3] to increase the length of the tweets. This approach overcomes the issues caused due to the short length of the tweets. In this approach, tweets containing URLs which are posted within 1 hour duration are collected and grouped into sessions. These sessions are then preprocessed to remove useless data from them. This preprocessed data is represented in a vector format and then fed into to the logistic regression classifier [1]. This classifier classifies the sessions into 8 predefined categories (Technology, Politics, Sports, Movies, Music, Fashion, Food and Travel). As a result of classifying, categories are assigned for each session. Based on the frequency of the categories of all the sessions, top three categories are chosen and declared as the interests of that user.

1.2 OUTLINE OF THE THESIS

This thesis has been organized as follows. Chapter 2 gives an overview of the social media, text analytics, Twitter and its APIs. Chapter 3 discusses the existing approaches for identifying the user interests using tweets. Chapter 4 explains the proposed methodology in detail. The architectural framework of the proposed approach is explained in detail. Chapter 5 explains the implementation details of the proposed approach. Technical specifications of the implementation are explained in detail for all the modules in the framework. In this chapter, code snippets are also attached for each module. Chapter 6 contains the detailed descriptions of the experiments conducted to evaluate the proposed approach. This chapter also contains the comparative analysis of the experimental results. This thesis is concluded with limitations and future work in Chapter 7.

CHAPTER 2 BACKGROUND

2.1 SOCIAL MEDIA

Millions of people use social media to communicate with each other for a variety of reasons. Social media such as blogs, microblogs, forums and multimedia sharing sites helps users to communicate personal messages, share breaking news, discuss about global or local issues, etc. This wide range of possibilities makes social media a popular medium of communication for the masses.

Wikipedia defines Social media as:
"Social media are computer-mediated tools that allow people to create, share or exchange information, ideas, and pictures/videos in virtual communities and networks" [13]

Among the web applications, social media sites attract most internet traffic. Report from Alexa [11] shows that social media sites occupy 40% of top 10 sites (Italics font in Table 1).

Table 1 Top 10 sites ordered by internet traffic [12]

Rank	**Website**	**Rank**	**Website**
1	Google.com	6	*Wikipedia.org*
2	*Facebook.com*	7	Amazon.com
3	*Youtube.com*	8	*Twitter.com*
4	Yahoo.com	9	Taobao.com
5	Baidu.com	10	Qq.com

The data generated from social media contain huge volumes of information about human interaction and collective behavior [12] which attracts researchers from various disciplines.

2.2 TYPES OF SOCIAL MEDIA

There are different types of Social media sites generating different formats of data (Text, image and video). Table 2 lists the categories of the social media and representative sites for them. Most of the social media sites produce text data, while others combine text data with multimedia content like image and video to make an effective communication.

Table 2 Types of social media [12]

	Category	**Representative sites**
Text only content	Wiki	Wikipedia, Scholarpedia
	Blogging	Blogger, LiveJournal, WordPress
	Social News	Digg, Mixx, Slashdot
	Micro Blogging	Twitter, Google Buzz
	Opinion & Reviews	ePinions, Yelp
	Question Answering	Yahoo! Answers, Baidu Zhidao
Text + Multimedia content	Media Sharing	Flickr, YouTube
	Social Bookmarking	Delicious, CiteULike
	Social Networking	Facebook, LinkedIn, MySpace

2.3 TEXT ANALYTICS

Due to the extensive usage of social media, a large volume of textual data is being generated on a daily basis. For instance, 500 million tweets are generated on Twitter in one day [14]. Such a huge volume of user generated data had to be processed to utilize them effectively. These data could be used in a variety of applications to enhance human life. For processing such huge amount of textual data, more advanced algorithms are required to learn the hidden patterns in the data. Text analytics is the method to process this huge corpus of unstructured text to get high quality data.

Text Analytics is defined in Wikipedia as follows:

"Text Analytics describes a set of linguistic, statistical, and machine learning techniques that model and structure the information content of textual sources for business intelligence, exploratory data analysis, research, or investigation." [15]

A text analytics framework consists of three stages. Figure 1 shows the basic framework of text analytics.

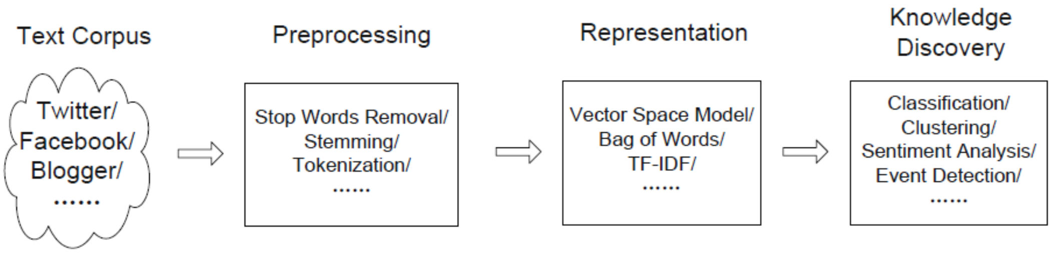


Figure 1 Basic framework of text analytics [12]

The three stages of text analytics are explained briefly in the following subsections.

2.3.1 Text preprocessing

Textual data that are produced by social media sites could not be analyzed directly because these are raw input texts. Preprocessing makes the text more consistent to facilitate the text representation. Stop word removal and stemming are the most commonly used preprocessing technique [8]. Stop words are a list of common and meaningless words. These words do not contribute much to the text analyzing. So these noisy words are removed from the raw text. Stemming is a process of reducing the derived words to their root word. For example "Go", "Going", "Gone" represent the root word "Go". Derived words are replaced by root words in the input text.

2.3.2 Text representation

After preprocessing the input text, only significant words are present in the text. These words need to be represented as numeric vectors to make the analyzing easier. There are two common approaches used in text representation. They are bag of words and string of words [8]. Generally Bag of words (BOW) approach is used to represent the preprocessed text. In Bag of words approach, the text is divided into words. This process is called as tokenization. The structure of the text is not maintained in this approach. Each word is represented as one single variable with different numeric weights. TF-IDF (Term frequency/Inverse Document frequency) is commonly used as the weighing mechanism. In string of words approach, sequence of the words is maintained. In most applications, Bag of words is used due to its simplicity [8].

2.3.3 Knowledge discovery

Once the textual data are transformed into numeric vectors, machine learning or data mining algorithms could be used to identify hidden patterns in the text. The most common approaches followed are classification and clustering [8]. Clustering fall under the category of unsupervised learning and classification falls under the category of supervised learning. In unsupervised learning, training data are not required. The documents which contain the textual data are segmented into different partitions such that each partition belongs to a single topic. This process is termed as clustering. In supervised learning, training data are required to make a machine learning method to learn a classifier to classify unseen data. Classification is used in various applications like news filtering, document organization and retrieval, opinion mining, email classification and spam filtering [8].

2.4 Distinct nature of the social media text

Latent and useful information in the text document could be obtained by applying the above procedures. These methods were originally designed to analyze traditional text data. Using these approaches to analyze social media text presents new challenges due to the distinct nature of the text content in the social media. The distinct features in social media, which pose these challenges are discussed in the following subsection.

2.4.1 Time sensitivity

Users on social networking sites post real time updates about movies, games, political campaigns, etc., which opens a large arena to know about the user's interest. These user interest could be used to build targeted advertising and recommendation to cater the needs of the users in real time. Traditional data are independent and identically distributed (i.i.d) [12]. But in the case of social media, users are connected and their friends can influence the content of the post the user updates. For example, users may not be interested in a movie which was released several months ago, but due to a friend’s recommendation, users may be interested in a movie irrespective of the release date [12]. Time sensitivity in social media data poses these new issues.

2.4.2 Short length

Some social media sites restrict the length of the content posted by the users on these sites. For instance, Twitter allows user to post only content of length 140 characters. These restrictions on social media text play an important role in the applications of social media [12]. Processing these short texts poses a new challenge to the available text analytics method. Because these methods require lots of words to perform statistical analysis, which is missing in the case of social media text. Many methods were proposed to solve this problem. Significant among them is using external knowledge sources [12] to bridge the semantic gap in text representation. It is difficult to use BOW based models in the social media text as there are not many common words in a short text [12].

2.4.3 Unstructured phrases

Social media texts are unstructured compared to the traditional text [12]. Text which is posted by users in a social media site can be of high quality as well as of low quality. The quality depends on the user who posts the content. Text can be ungrammatical and cannot be fitted into the traditional semantics of text. Users can coin new terms or abbreviations which are not present in traditional text documents [12]. For example “How r u?” These words do not produce any meaning because they are not real words. But users can use these types of new terms in social media. The unstructured nature of these texts can pose challenges to analyze and understand them. Sometimes text can be noisy. Irrelevant contexts are written in one single text which makes it difficult to identify the domain of the content.

2.4.4 Abundant information

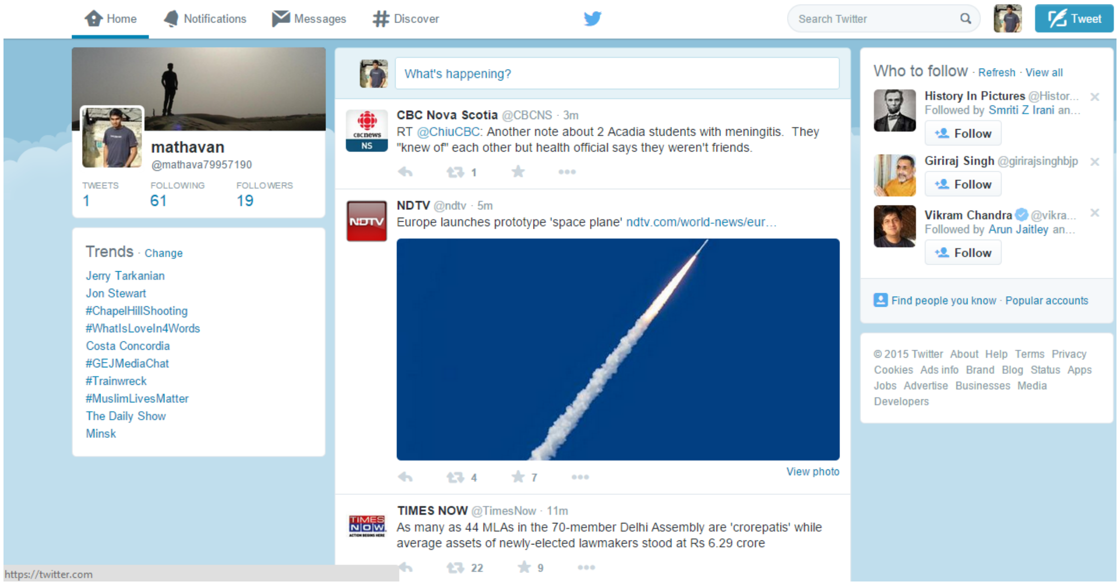
Textual content is not the only source of information on social media sites. Plethora of options has been given by social media sites to express user thoughts. Twitter allows users to use hashtags in the posts. These hashtags are used to represent the topics or keywords in the tweets. Many social media sites provide options to attach multimedia content to the posts. There are also abundant information sources like user location, URLs, tags and timestamp attached to the text data relating to user profiles [12]. All these additional information provided by the social media sites assists the traditional text analytics tasks to analyze the short text. Many approaches have been proposed to combine these additional information to short text to identify the hidden patterns in the text in a much efficient way.

2.5 TWITTER

Twitter is a social networking service that helps people to connect together by sending short messages of length 140-character. These messages are called as “Tweets”. “A Tweet is an expression of a moment or idea. It can contain text, photos, and videos” [19]. Twitter is also called as a microblogging site because it allows users to microblog about various topics. Microblogging is defined as a form of blogging that allows users to write short texts and send them to friends via internet or SMS [17]. Tweets are short and focused due to the restriction of the tweet length. Twitter allows registered user to write as well read tweets. But unregistered users can only read tweets. Initially, Twitter has set limits to the tweet length as 140 to support SMS compatibility. This has brought the SMS slang into Twitter.

Users started to use abbreviations to make the tweet as short as possible to fit within the allowed limit.

Twitter allows users to include URLs in the tweets. This is an optional feature, but still many people use it to link to an external resource. Due to the restriction in the tweet length, short URLs were used instead of the long URLs. URL shortening services like goo.gl, tinyurl.com, bit.ly can generate human readable URLs [10]. Figure 2 shows the home page of a Twitter account. “What’s happening?” is the place where users type their tweets and post it on their account. Users can either follow someone or allow someone to follow them. When a user posts a tweet, his followers would be able to see it. If they like that tweet they can retweet it. This is just like forwarding an email. Tweets from all the followers of the user will be displayed on the home page of the user.



CBC Nova Scotia @CBCNS · 3m
RT @ChiuCBC: Another note about 2 Acadia students with meningitis. They "knew of" each other but health official says they weren't friends.

NDTV @ndtv · 5m
Europe launches prototype 'space plane' ndtv.com/world-news/eur...

TIMES NOW @TimesNow · 11m
As many as 44 MLAs in the 70-member Delhi Assembly are 'crorepatis' while average assets of newly-elected lawmakers stood at Rs 6.29 crore

Figure 2 Home page of a Twitter account

Users can post tweets with hashtags in it. Hashtags are words prefixed with the symbol “#”. These are used to represent the topics or contents of the tweet. These could be used to group posts on a particular topic. Trending topics are generated using these tags. A tag or word or phrase that has been used in greater numbers than the other tags are called as trending topics. These trending topics help people as well Twitter to understand the events happening across the world. Users could also reply to someone by mentioning their names in the tweets by prefixing “@” before the username. This is referred as “mentions”

There are some interesting statistical information about Twitter users and their tweets. This information shows the importance of analyzing the Twitter data. The statistical data along with their chart representations are listed below.

- Twitter has more female users than male users [16]. Figure 3 shows the gender distribution on Twitter.

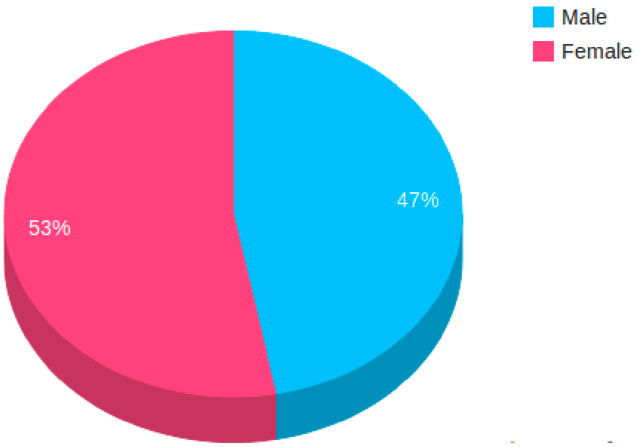


Figure 3 Gender distribution on Twitter [16]

- 74% of Twitter users are in the age limit of 15 to 25 [16]. Figure 4 shows the age distribution on Twitter.

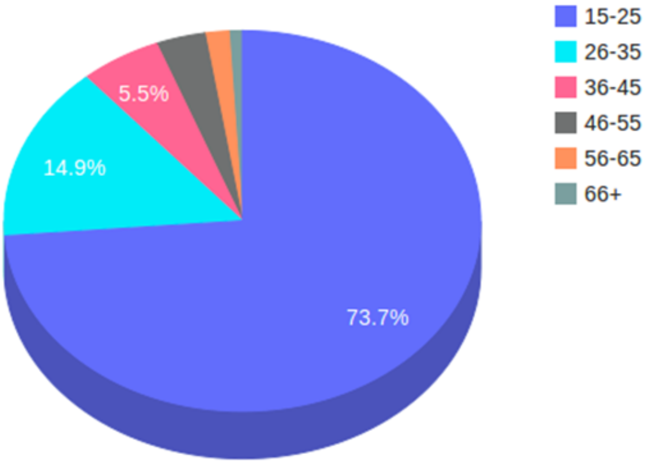


Figure 4 Self-disclosed age distribution on Twitter [16]

- United States of America has the highest number of Twitter users followed by United Kingdom [16]. Figure 5 shows the geographical distribution of Twitter users.

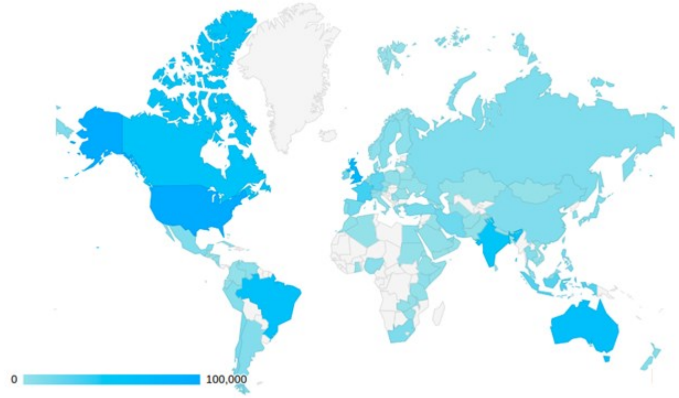


Figure 5 Geographical distribution of Twitter users [16]

- Figure 6 shows the most popular categories of Twitter users

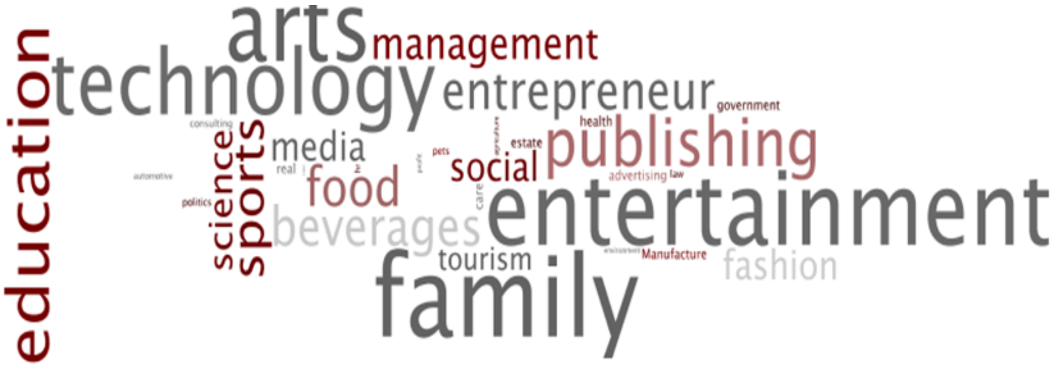


Figure 6 Most popular categories of Twitter users [16]

- Female users tweet more about family and fashion. Male users tweet more about Technology, sports and entrepreneurship [16]. Figure 7 shows the gender distribution by top 10 categories on Twitter.

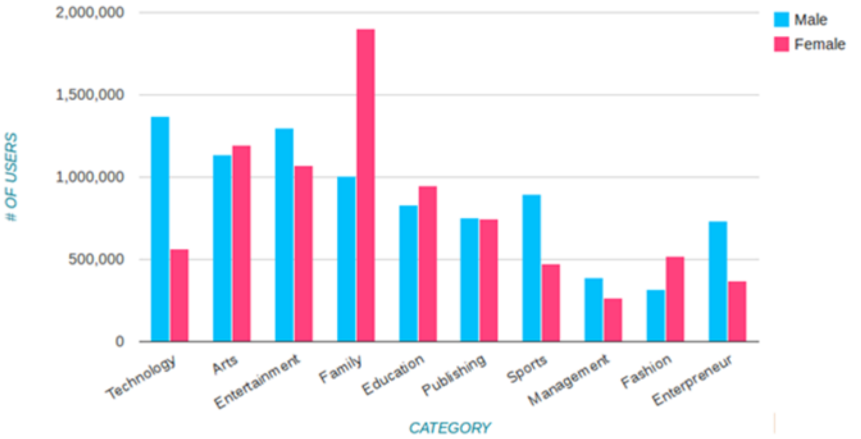


Figure 7 Global distribution by top 10 categories on Twitter [16]

- Twitter users among the age group 26-55 tweet more about family [16]

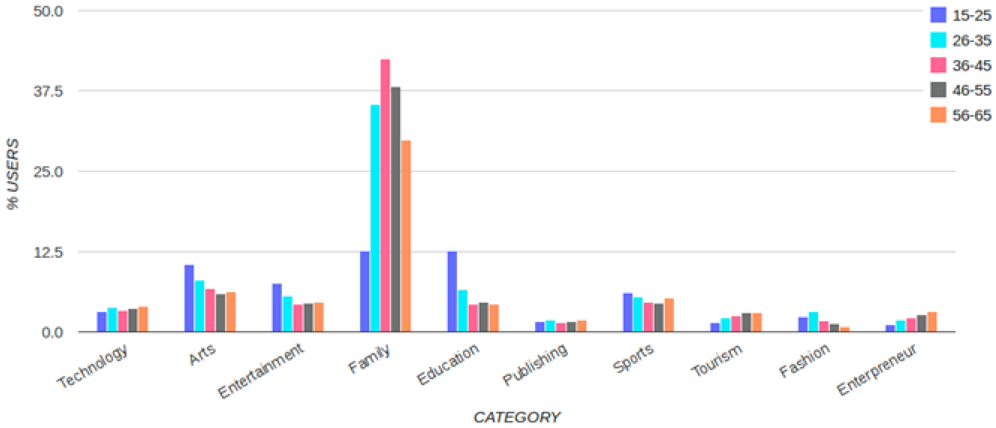


Figure 8 Top 10 categories by age group on Twitter [16]

2.6 TWITTER APIS

Twitter provides APIs to access the public tweets. Developers who are willing to develop applications to analyze the Twitter had to create a developer account. This account creation gives the credentials required to access the public tweets using APIs provided by Twitter. Two types of APIs are provided by Twitter to developers. They are REST API and streaming API.

REST API is used to read or write tweet data. OAUTH is used to authenticate the users and Twitter applications. When an application requests for a tweet data, Twitter sends the response in JSON format [20]. There is a rate limit window of 15 minutes. Only 15 requests can be made in this 15 minute window.

Streaming API gives developers a real time access to Twitter's global stream of data. Streaming API requires a constant HTTP connection between the client and server [21]. While implementing application with Streaming API, two separate process needs to be created. One for handling HTTP requests and the other for streaming connection. Streaming connection process collects tweet data from Twitter server and stores it in a data storage. After getting requests from client, "HTTP server process" reads the data from data storage and responds it back to the client [21].

The difference between REST API and Streaming API is that the REST API has rate limits, but streaming API does not have such limits [21]. Clients have to maintain a persistent connection with the server in the case of Streaming API. It is not so in the case of the REST API. Only when requesting the tweet data, the connection needs to be established in the case of the REST API.

Response data that is received from Twitter contains tweet text, list of hashtags, list of mentions, retweet count, URLs, time zone, description about the user, geolocation, etc., Using these data from Twitter, various types of analysis could be made which could provide deep insights about the user.

CHAPTER 3 RELATED WORK

Given a user with a list of tweets and its corresponding information (like hashtags, followers, retweets, URLs, Mentions, language), the proposed model is interested in finding the interest of the user. There are many existing research works that are currently going on in this domain. Each one has followed some unique ways of identifying user interests from the tweets and its associated information. Each tweet contains only 140 characters. Since text content is very little, it is difficult to classify the tweets based on just the textual information. Several methods have been developed to eliminate the data sparseness problem by using information associated with the tweets [1]. Some methods have used external sources like Wikipedia to enhance the tweet content [2] [7].

Tweet topics are identified in [1] by using textual information of the tweet combined with other associated information like hashtags, URLs, Twitter lists, replies, clicks and favorites. User Interests for each of the individual features are identified and then a scored voting mechanism is used to identify the topic of the tweet. This approach uses supervised learning to classify tweets into a defined taxonomy. This taxonomy contains 300 topics with a maximum level of 6 [1]. Existing taxonomies like ODP and freebase are used to construct this taxonomy.

This model [1] employs filtering techniques to filter the tweets during data collection for training the classifiers. User level, entity level and URL level filters are used to filter the tweets for a particular topic. After collecting the tweets, features are extracted for each tweet by using a Byte4Gram feature extractor. Regularized logistic regression is used to classify the tweets. Crowdsourcing and feedback from users are used to evaluate the identified topics. This result is fed back to the classifier to fine tune the classifier based on the feedback. Figure 9 shows the flow diagram of this approach.

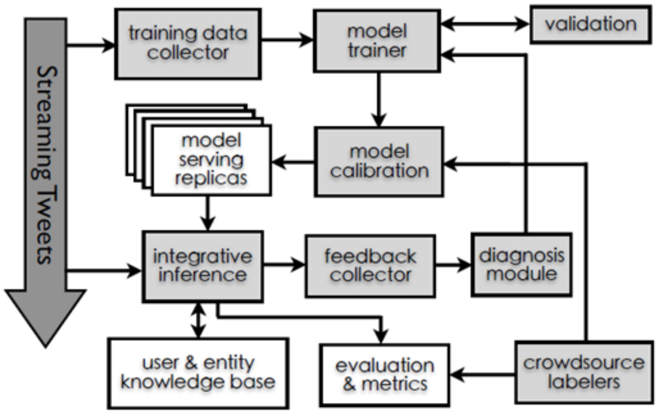


Figure 9 Flow diagram of an integrated topic inference system [1]

Tweets are short and using term frequency methods to identify the topics of the tweet would not produce precise results. Because same word might have different meaning in different context. So for disambiguating these words, some external knowledge base is required to identify the context of the word. Wikipedia is used as the external source to identify user interests from the tweets [7] [2]. In [1] identifying the context of the tweet is solely based on the training data. If the training data does not contain enough information to identify the topics, then there is a risk of identifying irrelevant topics.

Entity recognition and interest identification are the two stages involved while using Wikipedia as a knowledge base to identify the user interests [7]. Entities in the tweets are recognized by removing the stop words and hashtags. These entities are then disambiguated using Wikipedia. The categories of the disambiguated entities are identified using folksonomy (Wikipedia's user defined category) [7]. A subtree is generated with category as the root for each entity. This subtree is then processed to analyze the frequency of the categories to identify top-k interests of the user. This approach claims that frequency based topic models may perform better only for term-level topic modelling, but not for identifying high level topics. This method also claims that hashtag could not be used in identifying user interests as they are also identifying interests at the term level and they produce topics which are too specific. Figure 10 explains the two stage process of this approach.

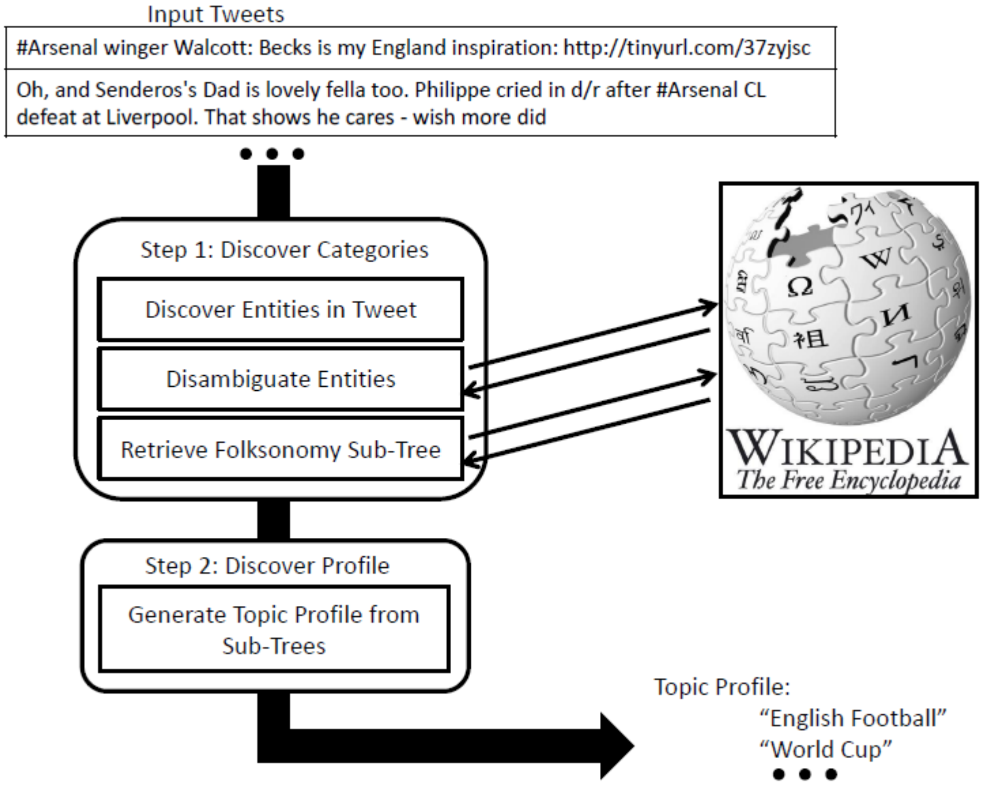

Input Tweets
#Arsenal winger Walcott: Becks is my England inspiration: http://tinyurl.com/37zyjsc
Oh, and Senderos's Dad is lovely fella too. Philippe cried in d/r after #Arsenal CL defeat at Liverpool. That shows he cares - wish more did
Step 1: Discover Categories
Discover Entities in Tweet
Disambiguate Entities
Retrieve Folksonomy Sub-Tree
WIKIPEDIA
The Free Encyclopedia
Step 2: Discover Profile
Generate Topic Profile from Sub-Trees
Topic Profile:
"English Football"
"World Cup"

Figure 10 Entity based topic profiles using Wikipedia [7]

Wikipedia has also been used to identify hierarchical interests of the user [2] compared to a list of interests as specified in [7]. In this approach [2] Wikipedia category graph has been transformed into a hierarchy. Hierarchy preprocessor is used to generate Wikipedia Hierarchy (WH) from the Wikipedia Category Graph (WCG). Zemanta is used for entity recognition from the tweets. Zemanta uses Wikipedia to generate the entities. Once the entities are recognized, the Hierarchy Interest Generator (HIG) is used to map the entities to the Wikipedia hierarchy to generate hierarchy interest graph for the user. Even though this method is able to generate hierarchical interests, this approach is not suitable for real-time applications as querying Wikipedia consumes much time to get the results. The architecture of this hierarchical interest identification is explained in figure 11.

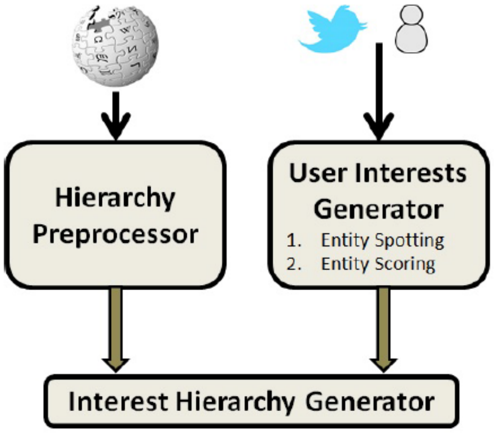


Figure 11 Architecture of hierarchical interest identification using Wikipedia [2]

Users have accounts on multiple social media services (Facebook, Twitter, Pinterest, Google+, etc.,). User interests across multiple online social networks (OSN) were studied in [3] considering users having accounts in both Twitter and Pinterest. User interest topics were identified from the Twitter using tweets. In Pinterest, user interests are identified by the images a user pins and repins. Each of these pins is tagged with a category. Tweets are collected and grouped into sessions. Since tweets contain limited information to identify the user interest, this method combines tweets which are posted within a certain time duration. Thereby increasing the chances of finding the topic more precisely. Each session contains tweets that are posted by the user within 2 hours. This time duration is considered because 2 hours are the minimum unit of time that could be measured in Pinterest.

Dataset from both Twitter and Pinterest are collected. Categories for each session is identified by analyzing the tweets that are present in that session. In the case of Pinterest, the most frequent category of pins in each session is identified as category for that session. Based on the identified categories in both Twitter and Pinterest, user behavior across Pinterest and Twitter are studied. Figure 12 explains the basic flow diagram of categorizing the session using pins and tweets.

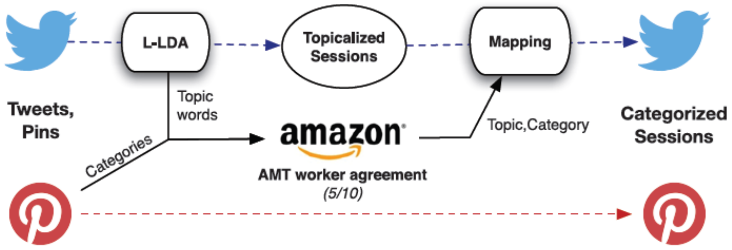


Figure 12 Flow diagram of a tweet categorization method using pins and tweets

User tweets about a particular topic, mostly when it is in the "public's eye". So user interest could be identified by knowing the tweet times of the user [6]. This approach maps tweet times of the user to the external world events to identify the interests of the user. Thereby minimizing the computation cost incurred in analyzing the tweets. But this method could not perform well when it is used as a standalone tool. Because during a particular time, many events would be happening across the world. Filtering the location of the user and trying to map the external world events to the tweets of the user is cumbersome. This method could be combined with the tweet analyzer to disambiguate the entities in the tweets. The context of the tweet could be identified easily by mapping it to the external world event.

Some users post tweets mostly on certain topics. If a user follows another user who posts tweets only on certain topics, then the user is interested in that particular topic [4]. In this approach, social annotations (Twitter Lists feature) have been used to identify the user interests. In Twitter lists, users group the experts on a particular topic which is of interest to them. So if a user has been grouped on several lists belonging to the same topic, then that user is said to be an expert on that topic. If a user U follows such user V who is an expert on a particular topic, then user U is interested in that topic. This approach has been used by [1] also. But in this method [4], user interest is identified only by using social annotations. In [1] Tweets combined with social annotations and other features have been used to identify the topics of the tweets. This method [4] also finds user interest using the tweets posted by the user and the posts received by the user. This approach suggests that Twitter lists provide precise interests comparing to self and received tweets.

Number of tweets posted by the user can also be used to identify the user interests. If the user posts too many tweets compared to other users, then that user is said to be active. This activeness level has been combined with the tweets posted by the user to identify the user interest [5]. This method also uses the retweet frequency of the user to find the user interest. Activeness feature has been introduced in this approach to identify the user interests of less active users. Because in the case of less active users, the tweets would be considerably less. So some scoring mechanism is needed to make sure that the precise user interests derived from active as well as passive users. Retweet feature is also used in this approach to identify the susceptibility nature of the user. Retweeting a certain posts indicate that the user is interested in that tweet. This approach uses the tweet session method discussed in [3] but activity level and retweet frequency has been introduced in this approach [5] to identify user interests for less active and less susceptible users.

3.1 TEXT MINING TECHNIQUES USED IN THE EXISTING APPROACH

In all the above approaches, different techniques were used in all the stages of text mining to solve various problems. The following section discusses all of those techniques used by the above approaches for user interest identification in each stage of text analyzing.

3.1.1 Data collection

Tweets are collected for training the classifier. This stage is referred as “Training data acquisition”. Twitter provides a REST API to get public data from Twitter users. All the tweets from the users are not useful for us. A filtering mechanism is needed to filter the tweets as well as users. The following are the filtering techniques used for “Training data acquisition”.

1. User level filter

   There are users who tweet mostly about a single topic. For example "@ESPN" tweets mostly about sports. So for training a sports classifier, this filter is used to collect tweets related to sports [1].

2. Entity level filter

   Some hashtags (or entities) represent a single topic unambiguously. For example #NBA (hashtag) is related to basketball. These unambiguous hashtags (or entities) need to be stored with their topics to filter the tweets for that topic [1]. There are many hashtags, which represent different meanings in different contexts. These hashtags could not be used in these filters.

3. URL level filter

   Users post tweets containing URLs. These URLs could be used to filter the tweets. URL strings usually contain topical information of the webpage it refers to [1]. If the topic is present in the URL string, then it would be same as the topic of the tweet.

   For example:

   Consider the following tweet from @ibnlive

   "Nasa selects tiny research satellites for future missions http://timesofindia.indiatimes.com/home/science/Nasa-selects-tiny-research-satellites-for-future-missions/articleshow/46172991.cms …

   The URL string present in this tweet gives the topic of the webpage as "Science".

4. Social annotations

   Twitter List feature could be used to filter users, who tweet mostly about a particular topic [1]. Twitter users create a list of certain topics of their interest. These lists contain users who tweet mostly on those topics. This

Twitter list could be used to identify users who predominantly post on a certain topic.

5. Language filter

   Language filters could be used to filter tweets based on the language of the tweet. For example, only tweets which are posted in English language could be used for training the classifiers. Translation option is also available on Twitter. Tweets, which are posted in another language are converted to the English language. This feature could also be used rather than avoiding the tweets from other languages.

6. Chatter detection:

   Tweets which contain emotion, feelings or tweets related to personal status updates are considered as chatter [1]. Because these tweets do not contribute much to the user interest identification. These tweets could not classified into any topics. So these tweets could be removed from the training data.

3.1.2 Preprocessing

The process of removing unwanted words, symbols and retaining meaningful words and entities is referred to as preprocessing. There are several techniques used to preprocess the text. Some common approaches are listed below:

1. Stop words removal:

   Common words (e.g. the, is, of, why, on) that are present in the text which does not provide much information for classifying the text are listed as stop words. These words are removed from the tweet to get only meaningful data from the tweets [8].

2. Stemming:

   Single word could be represented in different forms (For e.g. run, running, ran). Identifying the root word for these different forms is referred to as stemming [8].

There are many other techniques which are followed in tweet preprocessing. Hashtags, mentions and URLs are removed to get only words which contribute to the process of identifying the user interest.

3.1.3 Tweet representation

For classifying the tweets using the classifier algorithms, representation of tweets in some format is necessary. There are many representation techniques that are available to represent a text. Some of them are discussed below:

1. Bag of words:

   Tweets are represented as a set of words together with the frequency of that word [8]. Sequence of the words are not considered in this technique.

2. Bag of concepts:

   In this approach, tweets are represented a set of concepts. Semantic knowledge bases (Wikipedia, ODP, WordNet, etc.,) are used to identify the concepts [2]. These techniques have been used in [2] and [7] to represent the tweets. Since hierarchical interests are identified in [2], bag of concepts approach has been chosen to represent the tweets ([2] uses Wikipedia to generate the interests).

3. Topic model:

   A text document, which is about a certain topic contains words related to that topic in more frequency than other words. For example, a document about dogs contains more terms related to dogs rather than cats [9]. This intuition has been used to represent the tweets that are posted on a certain topic. Topic models have been used in [1] to represent the tweets.

“Bag of concepts” representation perform better in the case of tweets than the topic models and bag of words [2]. But for identifying the interests in real-time, Bag of concepts approach could not be used. Because it consumes too much time to query the external knowledge base (e.g. Wikipedia) [10].

3.1.4 Named Entity Recognition

“A named entity is a sequence of words that designates some real world entity” [8]. The process of identifying entities in a text is called “Named entity recognition” [7]. Entity recognition in tweets is not simple because of the informal nature and ungrammatical language of tweets [2]. Different approaches have been followed for identifying the entities. Capitalized and non-stopwords are used as entities in [7]. There are web services available for entity recognition. Zemanta is one of the popular entity recognition tools. This tool has been used by [2] to identify the entities in the tweets.

3.1.5 Classification

Machine learning algorithms are required to predict the user interests from the tweets. Some common algorithms used for this purpose are listed below.

1. Unsupervised clustering algorithms (k-means, pLSI and LDA) [1]
2. Information filtering approaches [1]
3. Weakly supervised models (Labeled Latent Dirichlet Allocation (L-LDA) which is a supervised version of LDA) [1]

L-LDA has been used to identify the interests in [3] and [4]. This method takes considerable latency for classifying and does not produce precise results [1].

3.1.6 Evaluation

After identifying the user interests from the tweets, these results had to be validated to ensure its quality. Many methods have been used to evaluate the results produced by the classifiers. Some of the prominent methods are discussed below:

1. Crowdsourcing

   For evaluating the results in large scale, crowdsourcing option is used. Amazon mechanical Turk (AMT) is one of the popular platforms used for crowdsourcing [5]. “Mechanical Turk is an online labor market where workers are recruited for the execution of tasks (called HITs, acronym for Human Intelligence Tasks) in exchange for a wage”[5]. Interests identified by the classifier are given to these workers along with the tweets corresponding to that interest. The crowdsource worker has to mark binary answers (yes/no) for this tweet-interest pair. Based on the answer received from the crowdsource workers. The precision and coverage of the classifier are evaluated. This approach has been used by [1] and [5] to evaluate their classifiers. In spite of the large scale evaluation of the tweets, this method has a disadvantage. Humans find it difficult to identify latent interests in a tweet made by some other person [5]

2. Feedback from users

   User feedback could also be used to evaluate the interests from the classifier. In this method, the interests identified by the classifier is sent directly to the user itself. So the user checks if the interests generated by the classifier match with his/her own interests.

   The feedback given by the user is collected and analyzed to evaluate the precision and recall. Crowdsourcing and feedback from users are combined together to produce maximum result [1].

3.2 MOTIVATION AND RESEARCH PROBLEM

Existing approaches tried to solve the problem of the short length of the tweets using external knowledge sources like Wikipedia [7] [2]. This approach overcame the problem of sparseness of the data. But these approaches could not be used to identify user interests in real time as querying external sources might take considerable time [10]. Another approach [3] overcame the problem of data sparseness by grouping tweets that are posted within a time duration of 2 hours. They used this approach to study the behavior of user interests across Twitter and Pinterest. The data sparseness problem was considerably reduced using this approach. This approach gave the motivation to use a group of tweets instead of single tweets for identifying the user interest. L-LDA has been used by this approach [3] to classify the sessions. It has been reported in [1] that L-LDA takes considerable latency for classifying. Logistic regression has been used by [1] and this classification algorithm gave a more precise real-time user interest. But this approach [1] used only single tweet to identify the topics of that tweet.

Considering all the pitfalls in the above solutions, a hybrid approach has been proposed to identify the user interest using a group of tweets containing URLs that are posted within 1 hour time duration. Tweets are filtered based on the URLs. This is based on the assumption that the tweets with URLs represent user interest more than the tweets without URLs. Tweets with URLs will have some concrete information which is of interest to the user. Chatter tweets or personal chats can be avoided in this case. So tweets containing URLs which are posted within 1 hour are grouped as one session. For grouping tweets, 2 hours session duration was used in [3]. But 2 hours is a long time. There is a very high possibility that users may post tweets of various context within 2 hour time. So to avoid this, 1 hour session duration has been used in the proposed approach. These sessions are then classified using logistic regression to identify the categories of the session. Top 3 categories with the maximum frequency of all the sessions are considered as the top 3 interests of the user.

CHAPTER 4 METHODOLOGY

4.1 PROPOSED APPROACH

Twitter users post millions of tweets about topics ranging from movie reviews, product reviews, news updates, natural calamities, political campaign updates, etc. Since tweets represent the interest of the user, the topics hidden in the tweets could be used to identify the interests of the user. There are many methods available to find the user interests from the tweets. These methods have been explained briefly in chapter 3. Most of the methods use tweets to identify the user interest. Since tweets are short in length, they do not provide a sufficient amount of information to the text analytics algorithm to analyze them precisely.

"Tweet session" approach has been proposed in [3]. A session is defined as a set of tweets that are generated within a time duration of two hours. Tweets that are posted within this time duration are supposed to be in the same topic [3]. But this time duration is too large for a person who tweets actively. Tweets could be from different context when the time duration is too large. There are many chances of getting noisy data by combining tweets from different context. Ottoni et al. [3] used L-LDA to classify the tweets. But inference with L-LDA has low latency as well as low precision [1]. The proposed approach combines the advantages in both of the approaches [1] and [3] and eliminates the limitations in these approaches.

Our new approach considers all the tweets containing URLs which are posted within one hour as one session and regularized logistic regression has been used to classify these sessions. By this approach, sufficient text is being provided to the text classifier to classify the tweets more precisely which helps in solving the data sparseness problem and the classifier infers the user interest in much less latency. Only tweets containing URLs are chosen, based on the assumption that these tweets represent the user interest more than the tweets without URLs. This approach identifies the top three user interests from the tweets.

4.2 ARCHITECTURE OF THE PROPOSED APPROACH

User tweets are classified into 8 predefined categories. Politics, Technology, Movies, Music, Sports, Fashion, Food and Travel are chosen as the categories. The most popular categories of Twitter users [16] are chosen as the categories for this approach. For identifying these categories from the tweets, the classifier had to be trained with labeled tweets. Labeled tweets are acquired from Twitter using Twitter APIs. In this approach, only English tweets are acquired for training as well as testing. Tweets belonging to other languages are ignored. Acquired tweets cannot be directly used. Tweets contain many unwanted data which had to be removed before using the tweets to train the classifier. So extra symbols, meaningless words and other useless data are removed from the tweets. After collecting the meaningful words from the tweets, these words are represented in a vector form. These vectors are used as inputs by the classifier to train the classifier models.

After training the models for each category, tweets from particular user are collected to identify the interests of that user. After collecting the tweets, unwanted data had to be removed from the tweets. Meaningful words are represented in a vector format. These vectors are then fed into the above trained models. These models predict categories for each tweet. After getting the categories of all the tweets. Top three user interest is identified from this list of tweet-category based on the frequency of the categories.

The Architectural design for the proposed approach consists of the following stages:

- Data acquisition
- Preprocessing
- Representation of data
- Generating classifier models
- Classification
- User interest identification

The basic framework for this approach is shown in Figure 13.

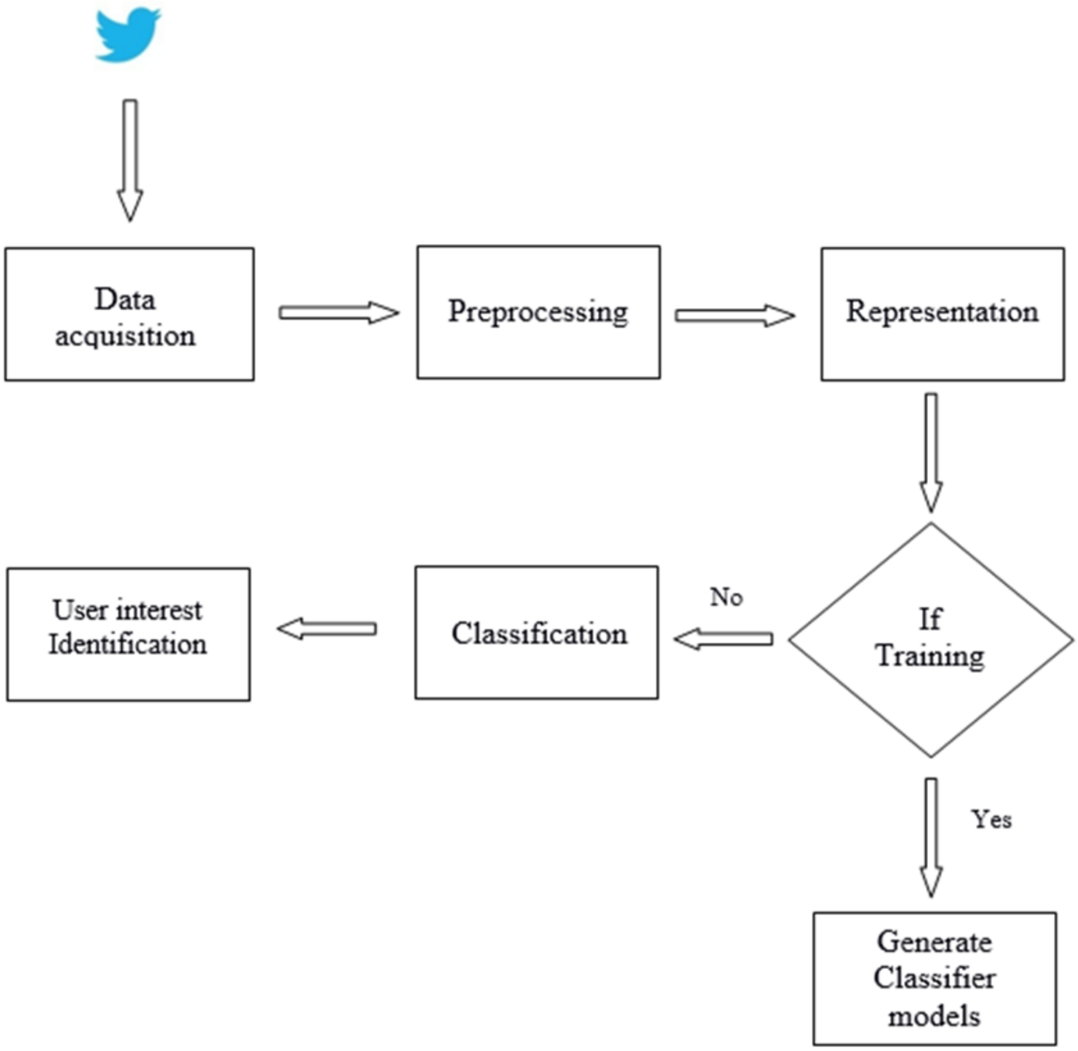


Figure 13 Basic framework of the proposed approach

4.2.1 Data acquisition

Tweets are collected for training as well as testing. There are subtle differences in acquiring the tweets in both these stages.

Training:

Labeled tweets are needed for training the classifier. Twitter provides "who to follow" option (https://twitter.com/who_to_follow/interests) which lists the user names based on their category [22]. Figure 14 shows the webpage of Twitter showing the list of users in a particular category. In our case, 8 categories of interest have been predefined. Screen names of the users are collected for each category using the "who to follow" option on Twitter. Using these screen names, tweets are collected from the users and stored in a file. For each category, one file is allocated. All the tweets belonging to that category are stored in a single file. Each line in the file contains single tweet.

Twitter provides many REST APIs to acquire data from Twitter using screen name. Using the "user_timeline" API [24], tweets from a particular user is acquired. This API takes many parameters as input. Screen name and number of tweets are two important parameters required to acquire a certain number of tweets from a particular user. Using this API, tweets for all the users in all the categories are acquired and stored in a file. In the proposed approach, only English tweets are collected. The remaining tweets are ignored. These collected tweets are used to train the classifier models.

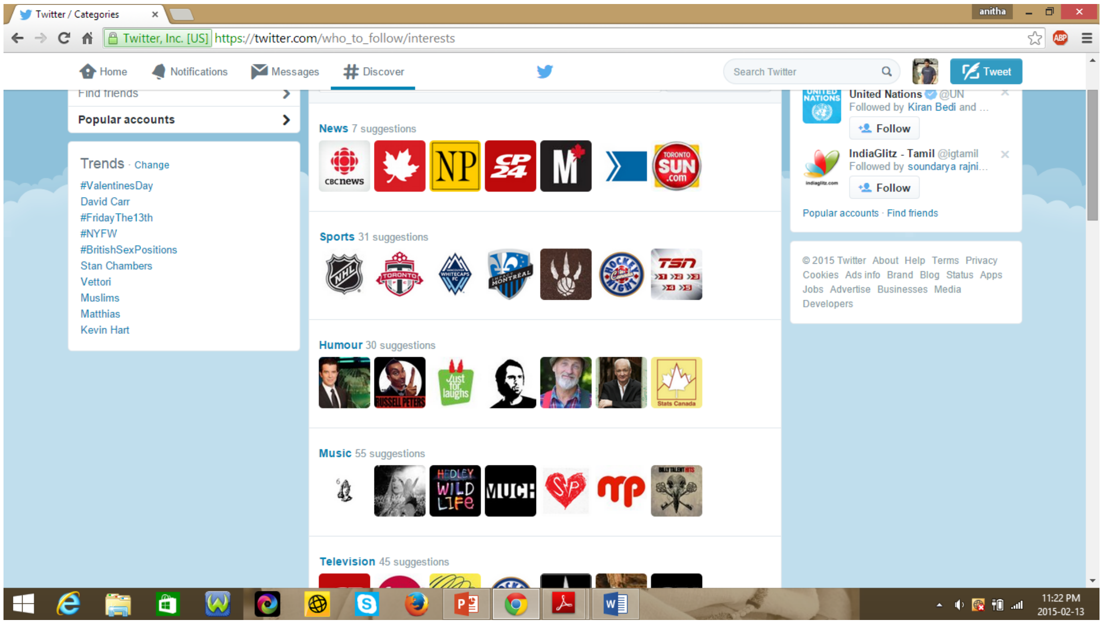


Figure 14 “Who to follow“- List of users in a particular category

Testing:

For collecting the testing data, only valid screen name and number of tweets to be acquired from the user is needed. Using the screen name, tweets are collected using the Twitter API (user_timeline). Only the tweets containing URLs are filtered for collecting the tweets for testing. After acquiring the first tweet from the test user, posted time of this tweet is stored. Using this posted time, all the tweets that are posted within 1 hour from the first tweet are grouped as one session (i.e. one single tweet). This process is continued for the remaining tweets of the test user. These sessions are stored in a separate file. These sessions are classified using the trained classifier models and the user interests are identified.

4.2.2 Preprocessing

Tweets (or sessions), which are collected in the previous step cannot be used to train the classifier directly. Because these textual data contain some unwanted text. In preprocessing, these unwanted symbols and meaningless words are removed from the original tweet. So that only meaningful words which contribute to the classification are retained. Most Tweets contain URLs, links, special symbols, abbreviations, hashtags, mentions and incorrect spellings [22]. The following rules have been followed to preprocess the tweets:

Rule 1: Remove all special characters except “#” and “@”

Tweets express emotions. So people use special characters to express their emotions. These special characters do not contribute much to our classification. So all these special characters are replaced with null characters. “Hashtags” are the keywords in the tweets followed by the “#” symbol (e.g. #SuperBowl). Many users would be using the same hashtag for a particular event. So these hashtags are retained in the tweets. @ Symbol is used to specify the username. This is being handled by Rule 2.

Rule 2: Remove all URLs and @mentions

Shortened URLs are used in tweets. These URLs do not provide much information on classifying the tweet. For example, consider this shortened URL “bit. ly/12Jkw6U”. These URL strings do not contain much text to predict the category. So these shortened URLs are removed from the text during preprocessing. “@” symbol is used to specify a screen name of a user in the tweets (e.g. @BarackObama). The words prefixed with “@” symbol is called as “mentions”. These words cannot be used to predict the category of the tweet. Because these words usually contain only user name.

Rule 3: Remove stop words

Stop words are a list of common and meaningless words. A list of stop words is maintained and these stop words are removed from the tweets. This stop word list is compiled using [25] and [26]. Some extra stop words have also been added into this list to make it complete. This list is compiled based on the common words used in Twitter. 735 stop words are present in the final stop word list. This list could be updated with new stop words.

Sample list of stop words in the stop word list:

- if
- don't
- dont
- now
- now
- retweet
- about
- above
- across
- actually
- an
- and
- are
- being

Rule 4: Apply Porter stemming algorithm to reduce the derived word to the root word

Derived words produce much complexity in classifying the tweets. Training data cannot contain all the derived words for a root word. So all the derived words are reduced to their root word to make the process much simpler. Porter’s algorithm has been used to stem the words.

Rule 5: Convert all words to lower case

Tweets are written in an inconsistent format. All the characters in the tweets can be either capital or small or mixed. To make the training data more efficient, all the words in the tweets are converted to lower cases.

Using the above rules, tweets are preprocessed and unigrams as well as bigrams for each tweet are stored in a file with the word Id. Another file contains, Tweet Id, category Id and word Id for word in the tweet. While training, Category Id has to be assigned to each tweet. List of categories with their category Id are stored in a category file. The contents of the category file are shown below:

Contents of Category file:

(Category Id)	(Category)
1	Sports
2	Music
3	Movies
4	Politics
5	Technology
6	Fashion
7	Food
8	Travel

Preprocessing of a single tweet from @BarackObama is explained in the below example:

"LIVE: President Obama is speaking at the White House **#CyberSummit** at **~~@~~Stanford**.
http://ofa.bo/i2rm "

Applying Rule 1 (special characters except # and @):
LIVE President Obama is speaking at the White House #CyberSummit at ~~@~~Stanford
http://ofa.bo/i2rm

Applying Rule 2 (URLs and mentions):
LIVE President Obama is speaking at the White House #CyberSummit at

Applying Rule 3 (Stop words):
LIVE President Obama speaking White House #CyberSummit

Applying Rule 4 (Stemming):
LIVE President Obama speak White House #CyberSummit

Applying Rule 5 (Lower case)
live president obama speak white house ~~#c~~ybersummit

After preprocessing the tweets, unigrams and bigrams are stored in the word list with a word Id assigned to each word.

Sample contents of Word list file:

(Word ID)	(Word)
1	live
2	president
3	live president
4	obama
5	president obama
6	speak
7	obama speak
8	white
9	house
10	speak white
11	white house
12	house ~~#c~~ybersummit
13	~~#c~~ybersummit

Tweet Id for this tweet is 1. "President Obama" is being labelled under the category of "Politics". So category Id for this tweet is 4. Tweet Id, Category Id, Word Id is written in a tweet-category-word-list file for preparing the training data for the classifier.

All the above discussed process for preprocessing is same for both training and testing data. Except the category Id. Because in testing data, category has to be identified. So "0" is assigned as the category Id in tweet-category-word-list file.

Sample contents of tweet-category-word-list file:

(Tweet ID)	(Category ID)	(Word ID)
1	4	1
1	4	2
1	4	3
1	4	4
1	4	5
1	4	6
1	4	7
1	4	8
1	4	9
1	4	10
1	4	11
1	4	12
1	4	13

4.2.3 Representation of data

LIBLINEAR library [27] is an open source library. This library has been used for large-scale linear classification. This library supports logistic regression and linear support vector machine algorithms. LIBLINEAR library has been used in this approach to classify the tweets. Binary method has been used to generate document features. Unigrams and bigrams which were collected in the preprocessing step had to be represented in LIBLINEAR format to give it as an input to the LIBLINEAR library [27].

The training (or testing) data for the proposed approach were created in the following LIBLINEAR format:

[Category] [Word Index1]: [value] [word Index2]: [value]............ [Word IndexN]: [value]
[Category] [Word Index1]: [value] [word Index2]: [value]............ [Word IndexN]: [value]
[Category] [Word Index1]: [value] [word Index2]: [value]............ [Word IndexN]: [value]
.
.
.
.
.
[Category] [Word Index1]: [value] [word Index2]: [value]............ [Word IndexN]: [value]

Training and testing data are stored in different files. 8 training files are generated for the 8 categories. Each line in the training files represents one tweet. All the tweets belonging to all the 8 categories are represented in all the 8 training files. In the case of testing, only 1 testing file is generated. All tweets belonging to the test user are represented in that file. Each field in the training (or testing) file is explained below.

Category:

Category field specifies the category Id of the tweet. Training file contains valid category Id of the tweet or "-1" in this field. But testing file contains "0" as the category Id in this field for all the tweets. For instance, in the case of training data, "4" is assigned as the category Id for the tweets belonging to "Politics" category that are present in the "Politics" training file and "-1" is assigned as the category Id for the remaining tweets. In the case of testing data, "0" is assigned as the category Id for all the tweets in the testing file.

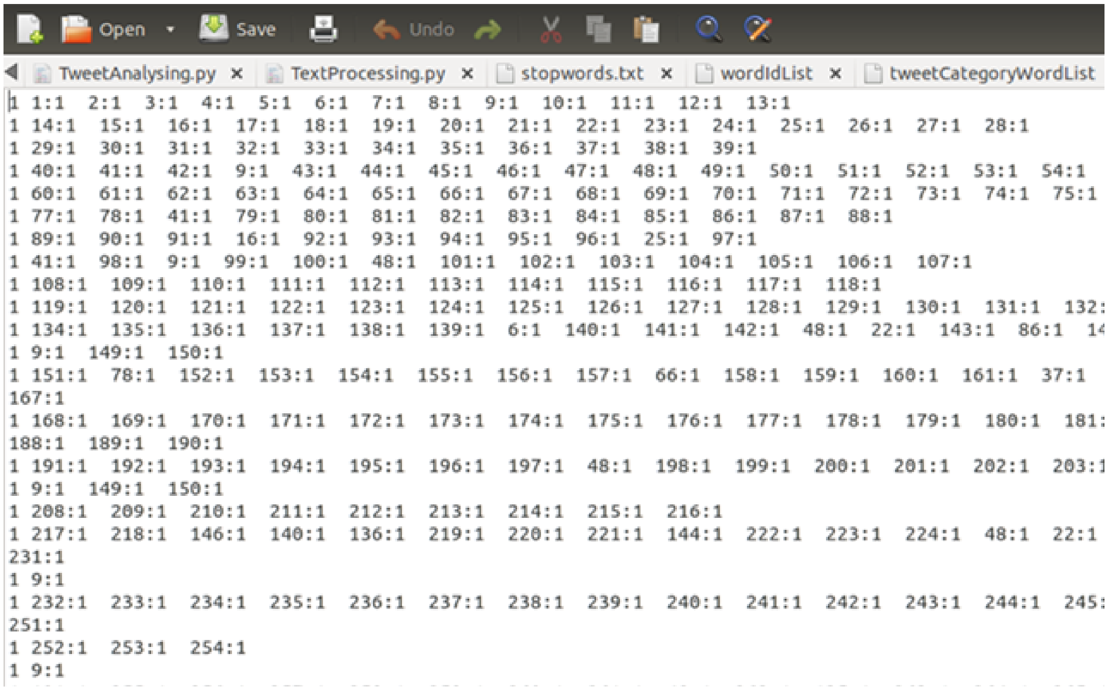
Word Index:

This field contains the word index of the i^{th} word in the tweet. This index is retrieved from tweet-category-word-list file which was generated in the preprocessing stage.

Value:

This field contains binary value. Either 1 or 0 is used. 1 represents the presence of this word in the tweet and 0 represents the absence of this word in the tweet. Since in our case, the word is always present in the tweet. 1 is used in this field.

Figure 15 shows the sample contents of “Sports” training data.



```
1 1:1   2:1   3:1   4:1   5:1   6:1   7:1   8:1   9:1   10:1   11:1   12:1   13:1
1 14:1   15:1   16:1   17:1   18:1   19:1   20:1   21:1   22:1   23:1   24:1   25:1   26:1   27:1   28:1
1 29:1   30:1   31:1   32:1   33:1   34:1   35:1   36:1   37:1   38:1   39:1
1 40:1   41:1   42:1   9:1   43:1   44:1   45:1   46:1   47:1   48:1   49:1   50:1   51:1   52:1   53:1   54:1
1 60:1   61:1   62:1   63:1   64:1   65:1   66:1   67:1   68:1   69:1   70:1   71:1   72:1   73:1   74:1   75:1
1 77:1   78:1   41:1   79:1   80:1   81:1   82:1   83:1   84:1   85:1   86:1   87:1   88:1
1 89:1   90:1   91:1   16:1   92:1   93:1   94:1   95:1   96:1   25:1   97:1
1 41:1   98:1   9:1   99:1   100:1   48:1   101:1   102:1   103:1   104:1   105:1   106:1   107:1
1 108:1   109:1   110:1   111:1   112:1   113:1   114:1   115:1   116:1   117:1   118:1
1 119:1   120:1   121:1   122:1   123:1   124:1   125:1   126:1   127:1   128:1   129:1   130:1   131:1   132:
1 134:1   135:1   136:1   137:1   138:1   139:1   6:1   140:1   141:1   142:1   48:1   22:1   143:1   86:1   14
1 9:1   149:1   150:1
1 151:1   78:1   152:1   153:1   154:1   155:1   156:1   157:1   66:1   158:1   159:1   160:1   161:1   37:1
167:1
1 168:1   169:1   170:1   171:1   172:1   173:1   174:1   175:1   176:1   177:1   178:1   179:1   180:1   181:
188:1   189:1   190:1
1 191:1   192:1   193:1   194:1   195:1   196:1   197:1   48:1   198:1   199:1   200:1   201:1   202:1   203:1
1 9:1   149:1   150:1
1 208:1   209:1   210:1   211:1   212:1   213:1   214:1   215:1   216:1
1 217:1   218:1   146:1   140:1   136:1   219:1   220:1   221:1   144:1   222:1   223:1   224:1   48:1   22:1
231:1
1 9:1
1 232:1   233:1   234:1   235:1   236:1   237:1   238:1   239:1   240:1   241:1   242:1   243:1   244:1   245:
251:1
1 252:1   253:1   254:1
1 9:1
```

Figure 15   Sample contents of the “Sports” Training data

4.2.4 Generating classifier models

Classifier models are generated for each category using the training data generated in the above step.

LIBLINEAR provides a Train API to train the classifier models using training data.

Arguments for Train API:

Train [options] training_set_file

Options:

There are many options available in the Train API. But the following three is being used in this approach.

-s type: This option takes the type of classifier from the list of classifiers.

Default value is set to 1.

For multi-class classification

0 -- L2-regularized logistic regression (primal)

1 -- L2-regularized L2-loss support vector classification (dual)

2 -- L2-regularized L2-loss support vector classification (primal)

3 -- L2-regularized L1-loss support vector classification (dual)

4 -- Support vector classification by Crammer and Singer

5 -- L1-regularized L2-loss support vector classification

6 -- L1-regularized logistic regression

7 -- L2-regularized logistic regression (dual)

For regression

11 -- L2-regularized L2-loss support vector regression (primal)

12 -- L2-regularized L2-loss support vector regression (dual)

13 -- L2-regularized L1-loss support vector regression (dual)

-c cost: This option is used to set the parameter C in the classifier algorithm. The default value is set to 1.

-q : This option is used to prevent the output from the classifier to be printed in the output screen.

Training set file:

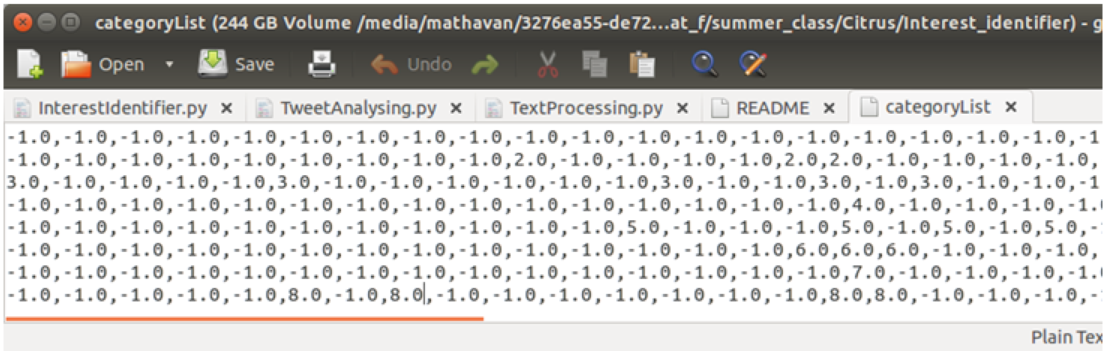
Training data for a particular category is given as input to this parameter. This parameter takes the training data that is represented in subsection 4.2.3.

L2 - regularized logistic regression has been chosen for this approach and the cost is assigned as 5 [22]. Using the train API, classifier models have been generated for each category.

4.2.5 Classification

Tweets are collected from the user for whom the interests had to be identified (as explained in subsection 4.2.1). These tweets are stored as sessions. These sessions are preprocessed (as explained in subsection 4.2.2) and the test data is generated (as explained in subsection 4.2.3). Using this test data, Predict API is used to classify the tweets. Each classifier model is used to predict the category for the sessions. The categories generated by each classifier are stored in a file. Predict API takes the test data file and the classifier models as the input. Categories for each session is generated by all the classifiers and stored in a file. If a classifier identifies that a session belongs to the category of the classifier, then the corresponding category Id of the classifier is assigned for that session. Else "-1" is assigned as the category Id for that session.

Figure 16 shows the sample contents of a category file. Each line in the category file shows the category predicted by each classifier for all the sessions. 1st line is the output produced by the "Sports" classifier, 2nd line is the output produced by the "Movies" classifier.


categoryList (244 GB Volume /media/mathavan/3276ea55-de72...at_f/summer_class/Citrus/Interest_identifier) - g
Open
Save
Undo
InterestIdentifier.py
TweetAnalysing.py
TextProcessing.py
README
categoryList
Plain Tex

```
-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1
-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,2.0,-1.0,-1.0,-1.0,-1.0,2.0,2.0,-1.0,-1.0,-1.0,-1.0,
3.0,-1.0,-1.0,-1.0,-1.0,3.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,3.0,-1.0,-1.0,3.0,-1.0,3.0,-1.0,-1.0,-1
-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,4.0,-1.0,-1.0,-1.0,-1.
-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,5.0,-1.0,-1.0,-1.0,5.0,-1.0,5.0,-1.0,5.0,-
-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,6.0,6.0,6.0,-1.0,-1.0,-1.0,
-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,7.0,-1.0,-1.0,-1.0,-1.
-1.0,-1.0,-1.0,-1.0,-1.0,8.0,-1.0,8.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,8.0,8.0,-1.0,-1.0,-1.0,-
```

Figure 16 Sample contents of a category file

4.2.6 User Interest Identification

Each classifier predicts the categories for each session. So for a particular session, there will be 8 categories. In most of the sessions, there will be only one classifier which predicts the exact category of that session. Other classifiers would be predicting "Others" as the category for that session. If a single session contains multiple categories, then this prediction will not be considered for the final result.

Also, if all the classifiers give "Others" category to a session, then that prediction is also not considered for the final calculation. The list of categories satisfying the above conditions is chosen. Then the frequency of each category in this list is identified. Top 3 categories which have maximum frequency are listed as the interests of that user.

CHAPTER 5 IMPLEMENTATION

5.1 DEVELOPMENT ENVIRONMENT AND LIBRARIES USED

The proposed methodology is implemented in Python and run under Linux platform. Table 3 shows the development environment used for implementing the proposed approach.

Table 3 Development environment for the proposed approach

Development Platform	Ubuntu 14.04.1
Scripting Language	Python 2.7.6
Programming Language	Java 1.7.0_55

Open source libraries have been used to implement the text mining algorithms, text preprocessing methods and tweet acquisition from Twitter. The libraries used for the above purpose are listed in the Table 4.

Table 4 Libraries used

Classification algorithms library	Liblinear 1.94
Natural Language Processing libraries	NLTK
Python library for Twitter API	Tweepy

5.2 IMPLEMENTATION DETAILS OF EACH PROCESS IN THE PROPOSED APPROACH

As mentioned in the chapter 4, the proposed approach consists of 6 stages. This section explains the implementation details of those 6 stages.

5.2.1 Data acquisition

The application has to be authenticated by Twitter before collecting data from Twitter. This is the initial step to be performed before collecting data from Twitter. When the application is registered in Twitter, the following parameters are provided to access the Twitter to collect tweets: Consumer token, consumer secret key, access token and access secret key. Tweepy is a Python library for the Twitter API. This library is used for the authentication purpose. OauthHandler () and set_access_token () are used to authenticate the application.

The following code snippet shows the authentication process:

Code snippet:

```
# Function to authenticate the application to collect tweets
 def twitterAuthentication():
      consumer_token = 'G0MhpuLETIKgKkCduPOHtZoLn'
      consumer_secret= 'GP8thee8TXjoxj129HJ8zu7XnNsha4GW8Cx72SlCe34xMGVn'
      access_token = '2399917969-gtaFO57BLc03M6KgXbVMh82lcuAbMwIKnoc9DDx'
      access_secret = 'BXz1OaYwE5aicD9MpO5oxUQXtT4j5W0nyCKqQ7vwsT5VU'

      auth = tweepy.OauthHandler(consumer_token,consumer_secret)
      auth.set_access_token(access_token,access_secret)
      api  = tweepy.API(auth)
      return api
```

After the authentication process, tweets are collected from Twitter using Twitter REST APIs. Using the Tweepy library, Twitter data are collected from Twitter. In both the training and testing stages, only English tweets are collected. Different implementation strategies have been used in data collection for training and testing. These strategies are explained below:

Training:

For training the classifiers, data were collected from all the 8 categories. Data was collected from 15 users under each category. These 15 users were selected using the "who to follow" option. Twitter gives the tweets in the form of pages. 3 pages were collected from each user. Each page contains 200 tweets. Table 5 shows the details about the tweets collected for training.

Table 5 Details about tweets collected for training

Number of pages collected per user	3
Number of tweets collected per page	200
Number of tweets collected per user	600
Number of users collected for each category	15
Number of tweets collected per category	9000
Total number of categories	8
Total number of tweets collected	72000

Tweepy library was used to collect tweets from Twitter. Twitter API user_timeline returns tweets from a particular screen name. This API also takes a number of tweets per page as input. This API is given as an input to the tweepy.Cursor API. The cursor API in tweepy is used to get the tweets from the specified page.

The syntax of the cursor API is given below:

```
tweepy.Cursor(api.user_timeline).pages()
```

The code snippet below shows the Python code to collect tweets from a single user belonging to a particular category. This function is called multiple times to collect tweets from all the users in all the categories. This function takes the screen name of the user, number of tweets, number of pages and text file as input.

Code snippet:

Input:

api – contains wrapper for the API as provided by Twitter
screenName – screen name of the user
maxTweets – Number of tweets to be acquired from the user
numPages – Number of pages to be acquired
tweetFile - Contains the File handle

```
#Function to collect tweets and write the tweets in a text file
def tweetDataCollection (api, screenName, maxTweets, numPages, tweetFile):
        # Collect max tweets per page for a particular screen name
        for page in tweepy.Cursor(api.user_timeline, id=screenName, count =
        maxTweets).pages(numPages):
                for status in page:
                        # Collect only English language tweets
                        if(status.lang == 'en'):
                                # Write each tweet in the respective tweet category file
                                tweetFile.write(' '.join((status.text).split()) + "\n")
```

Testing:

For collecting the test data, the above approach cannot be used. URL filtering and session creation had to be done in the testing case. tweepy.cursor() API is used to collect tweets for a particular user using the screen name of the user. Only tweets with URLs and tweets belonging to English language had to be collected. So, “lang” option in Twitter API is used to check the language of the tweets and “entities” option is used to check the URL presence in the tweets. After applying language and URL filters, the tweets had to be grouped based on session duration of 1 hour. These sessions are stored in file for further processing. Table 6 shows the details about the tweets collected for testing.

Table 6 Details about tweets collected for testing

Number of pages collected per user	1
Number of tweets collected per page	200
Total Number of tweets collected per user	200
Session duration of each sessions	1 hour

The code snippet below shows the Python code for collecting tweets for testing. This function takes the screen name of the user, number of tweets and number of pages as input.

Code Snippet:

Input:

api – contains wrapper for the API as provided by Twitter

screen_name – screen name of the user

maxnumtweets – Number of tweets to be acquired from the user

numPages – Number of pages to be acquired

```
# Function to collect sessions for testing
def sessionUrlOnly (api, screen_name, maxnumtweets, numPages):
    flag = 0
    tweet_count = 0
    session = 0

    # File to store the test tweets
    testf = codecs.open(os.path.join(dirname, "tempfiles/testTweets.txt"), 'w+',
encoding='UTF-8')

    # Collect max tweets per page for a particular screen name
    for pages in tweepy.Cursor(api.user_timeline,id=screen_name, count =
maxnumtweets).pages(numPages):

            # Tweets per page
            for status in pages:

                # English language and URL filter
                if(status.lang == 'en') and (status.entities['urls']):
                    tweet_count+=1
                    # 1st tweet of the user
                     if (flag==0):
                            # Stores the time at which the 1st tweet was posted
                            max_time_date = status.created_at

                            # Calculating time with session duration = 1 hour
                            min_time_date = status.created_at - timedelta(hours = 1)

                            # storing the tweets in a string
                            temp_tweet_array = ' '.join((status.text).split())
                            flag = 1
                # Remaining tweets of the user
                 else:
                           # Check if the tweets are within the session duration
                           if(status.created_at > min_time_date):
                                     # Store the tweets in a string
                                     temp_tweet_array += ' '+' '.join((status.text).split())

                                      # Last tweet of the user
                                      if(tweet_count == maxnumtweets):
                                          session+=1
                                          # Write the session in file
                                          testf.write(temp_tweet_array + "\n")
                                          temp_tweet_array = ""
                           # New session starts
                           else:
                                      session+=1
                                      # Write the session in the file
                                      testf.write(temp_tweet_array + "\n")

                                      # Store the tweets for the new session
                                      temp_tweet_array = ' '.join((status.text).split())

                                     # Stores the new time at which this tweet was posted
                                      max_time_date = status.created_at

                                     # Calculating new time with session duration = 1 hour
                                     min_time_date = status.created_at - timedelta(hours =1)
    if(temp_tweet_array):
        testf.write(temp_tweet_array + "\n")
        session+=1
testf.seek(0)
```

In the above Python code, “timedelta” is used to calculate the session duration. "timedelta” is a function provided by “datetime” module in Python. This “datetime” module provides the classes for manipulating the data and time.

For checking the presence of URLs in the tweets “status.entities['urls']” is used. This provides the list of URLs present in the tweets. This list is empty when there are no URLs present in the tweets.

All the tweets belonging to one session are stored in a single line in the file. So, the number of lines in the file corresponds to the number of sessions. These sessions are then preprocessed and analyzed.

5.2.2 Preprocessing

After collecting the data from Twitter, these data are preprocessed. Stemming is performed on the tweets using NLTK package. After reducing the derived words to their root words, regular expression libraries are used to remove symbols from the stemmed words. These words are stored in a list. Stop words are removed from this list.

The Bigrams are collected from this list and stored in another list. Both the unigrams and bigrams are returned to store them in a file in the format specified (in subsection 4.2.2). The preprocessed results are stored in separate files for training and testing. The code snippet below shows the Python code to preprocess the tweets.

Code snippet:
Input:
 tweet - contains a single tweet of a user
 stopwords – list of stopwords
Output:
 finalwordList – contains the preprocessed word list (both unigrams and bigrams)

```
# Function to preprocess the tweets
def tweetPreprocessing(tweet, stopwords):
        stemmer = PorterStemmer()

        # Remove symbols from the tweets and store the lower case words
        unigrams = [word.lower() for word in removeSymbols(tweet.split())]

        # Perform stemming to reduce the derived words to root word
        list = [stemmer.stem(word)  for word in unigrams if word]

        # Remove empty strings
        list = [word  for word in unigrams if word]

        # Remove Stop words
        list = [word for word in list if word not in stopwords]

        # Collect bigrams from unigram list
        bigrams = [list[i] + " " + list[i + 1] for i in range(len(list) – 1)]

        # Combine  unigrams and bigrams
        finalwordList = list + bigrams

        return   finalwordList
```

```
# Function to remove unnecessary symbols from the tweets
def removeSymbols(args):
    symbolsRemovedList = []

    for arg in args:
        # Remove mentions
        arg = re.sub(r"(?:\@|')\S+", "", arg)
        arg.strip()

        # Remove URL
        arg = re.sub(r"http\S+", "", arg)
        arg.strip()

        # Remove all symbols except # and  '
        for symbols in string.punctuation:
            if not (symbols=='#' or symbols =='\"):
              arg = arg.replace(symbols, ")
              arg = arg.rstrip(symbols)
                if(len(arg) == 1):
                    continue

        # Store the words in a list
        if len(arg) > 0:
            symbolsRemovedList.append(arg.rstrip('\n'))

  return symbolsRemovedList
```

5.2.3 Representation of data

Preprocessed tweets need to be represented in the format required by the LIBLINEAR library.

The format required by the LIBLINEAR Library is give below:

[Category] [Word Index1]: [value] [word Index2]: [value]............. [Word IndexN]: [value]

This is the format for a single tweet. Category of the tweet, word index from the word list and the presence of the word in the tweet are the fields present in the above format.

Word list file, tweet-category-word-list file are used as input to this module.
The code snippet below shows the Python code to represent the preprocessed tweet in LIBLINEAR format.

Code Snippet:
Input:
tweetWordIndex – Contains all the preprocessed word index in a tweet
tweetCategory – Contains the category of the tweet

Output:
StrToWriteToFile – Contains the string to write to file in the LIBLINEAR format

```
#Generating data in the format required by LIBLINEAR library
def generateDataForLiblinear(tweetWordIndex, tweetCategory, cat_cnt):
    if(cat_cnt == tweetCategory):
        CategoryId=tweetCategory
    else:
        CategoryId=-1
    StrToWriteToFile = str(CategoryId))
    for word in tweetWordIndex.split():
        StrToWriteToFile    += Str(" "+str(word)+":"+str(1)+" ")
    StrToWriteToFile    += ("\n")
    return  StrToWriteToFile
```

5.2.4 Generating classifier models

After representing the tweets in the LIBLINEAR library format. Classifier models for each category had to be generated. LIBLINEAR provides a train() API to generate the classifier models. svm_read_problem() API is also provided by LIBLINEAR library. This API collects the labels and features in all the tweets belonging to a particular category. These labels and features of a particular category are fed as input to the train() API. The classifier algorithm type and the cost have to be mentioned in the train() API. L2 Logistic regression is used as the classifier algorithm and the cost is set to 5. So the parameters for train() API is set to “-s 7 -c 5 -q”. The train() API returns the classifier model. This model is saved using save_model() API. The following code snippet shows the Python code for generating the classifier models.

Code snippet:

Input:

categories – This file contains the list of categories (8 categories)

TrainedData(*)   - Trained data file for each category

Output:

Classifier models for each category

```
def trainIndivModel():
   categoryFile = open(“categories")
   for eachLine in categoryFile:
        labelList, featureList = svm_read_problem("TrainedData"+eachLine.strip())
```

```
params = '-s 7 -c 5 -q'
#LIBLINEAR library API to train a model
model = train(labelList, featureList, str(params))

modelName = (eachLine.strip() + "TrainedModel")
#LIBLINEAR library API to save the classifier model
save_model(modelName,model)
```

5.2.5 Classification

The classifier models for all the categories were generated using the LIBLINEAR library. The test tweets are represented in the LIBLINEAR Library format. These represented test tweets are classified using predict() API of LIBLINEAR Library. The labels and features present in the represented test tweets were collected using svm_read_problem() API. The test tweets are classified using all the models. The categories generated by these models are used to generate the user interest. The following code snippet shows the Python code for classifying the test tweets.

Code snippet:

Input:

```
   modelFilePath – Contains the path of the classifier model
   testTweetDataRepFile – Contains the test tweets represented in LIBLINEAR format

Output:
   labelList – category index for each tweet (-1 or category index (1 to 8))
```

```
def predictCategory( modelFilePath, testTweetDataRepFile):

    # Load the classifier model which was generated in the training stage
    model = load_model(“modelFilePath")

    # Load the generated test tweet data
    labels,features = svm_read_problem(“testTweetDataRepFile”)

    # Call the predict method to determine the category of the tweet
    labelList = predict(labels, features, model)

    return labelList
```

5.2.6 User interest identification

The classifier predicts the categories for each tweet. Some tweets contain more than one category and some tweets contain “others” category. These tweets are ignored while identifying the user interests. The frequency of the number of categories in the final list after ignoring the above cases is calculated. From this frequency list, top 3 are identified as the user interests. The following code snippet shows the Python code for calculating the top three user interest.

Code snippet:
Input:
categoriesFromEachClassifier – Contains the categories for each tweet
CategoryFile – contains the names of the 8 categories

Output:
cat_list – contains the top 3 user interest

```
def userInterestIdentification(categoriesFromEachClassifier, categoryFile):
        temp_cat_list = []
        cat_idx = 0

        for eachline in categoriesFromEachClassifier:
            # Tweets predicted with only one category
            if (len(eachline.split()) == 1):
                    outstring =  ' '.join(eachline.split())
                   # Tweets predicted with categories other than “Other”
                   if(outstring != 'Other'):
                            temp_cat_list.insert(cat_idx,str(outstring))
                            cat_idx+=1
        cat_list = []
        cat_idx = 0
       # store the frequency of the each category with their category name
        for eachLine in categoryFile:
           cat_list.append([])

           # store the category name
           cat_list[cat_idx].append(str(eachLine.strip()))

           # store the frequency of the category
           cat_list[cat_idx].append((str(temp_cat_list).count(eachLine.strip())))
           cat_idx +=1

        # Sort the category list is descending order
        cat_list = sorted (cat_list, key=operator.itemgetter(1), reverse = True)

        # Top 3 interest of the user
        return cat_list[0:2]
```

CHAPTER 6 EXPERIMENTAL RESULTS AND ANALYSIS

This chapter presents the evaluation of the proposed approach by comparing the performance achieved by using “Sessions” and “Tweets”. Activeness of the user has also been used as a parameter to test if the activeness has any impact on the above results. This chapter also evaluates the various features of Twitter like Twitter list and Twitter description to identify the user interest. Experiments are conducted on the proposed approach by using the following methods:

1. User interest identification using “Session” and “Tweets”
2. User interest identification using URL filtered “Session” and “Tweets”
3. User interest identification using activeness of the user as a parameter in “Sessions”
4. User interest identification using “Twitter List” and “User profile description”

6.1 EXPERIMENTAL SETUP

For conducting the above mentioned experiments, tweets had to be collected from the users to identify the user interest. To verify the user interest generated by the proposed approach, the users had to be labeled with their category. Popular Twitter users whose categories are very well known to the public are chosen as the test users. Top 20 Twitter users under each category is identified from the lists provided by Twitter analytics website [28-35]. Some well-known users under each category has also been added into this list. Tweets are collected from these users to validate if the calculated interest of the user is same as the original interest of the user. The experimental setup for testing the proposed approach is shown in Table 7.

Table 7 Experimental setup details

1	Number of users	80/120/160
2	Number of tweets per user	200
3	Tweet Language	English
4	Number of categories	8
5	Number of users per category	10/15/20

6.2 Evaluation metric

Precision, recall, F-measure and accuracy are the typical measures which are used for evaluating the performance of the classifiers. For evaluating the text classification task, a contingency matrix depicting all the possible outcomes of the classification should be defined [42, 43]. The contingency matrix table is defined in Table 8.

Table 8 Contingency table for binary classification [42, 43]

		True class	
		Positive	**Negative**
Predicted class	**Positive**	True Positives (TP)	False Positives (FP)
	Negativ	True Negatives (TN)	False Negatives (FN)

The performance measures for the classifier are defined based on the contingency table. These performance measures are defined in Equation 1, 2, 3 and 4.

$$Precision\ (P) = \frac{TP}{TP + FP} \tag{1}$$

$$Recall\ (R) = \frac{TP}{TP + FN} \tag{2}$$

$$F - measure\ (F) = \frac{2 \times P \times R}{P + R} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

When several classifiers are used to classify the text corpus, then a single aggregate measure is required to evaluate the performance of the classifiers. For this purpose, Microaveraging and Macroaveraging methods are used [44]. Macroaveraging computes a simple average of the performance measure on all the classifiers. Microaveraging sums the individual True Positives, False Positives and False Negatives of all the classifiers and then calculates the performance measures. Table 9 shows an example of contingency table for two classes and a pooled contingency table for those classes.

Table 9 Example of contingency table for two classes [44]

		Class 1	
		True class	
		P	**N**
Predicted class	**P**	20	20
	N	20	900

		Class 2	
		True class	
		P	**N**
Predicted class	**P**	85	10
	N	10	800

		Pooled table	
		True class	
		P	**N**
Predicted class	**P**	105	30
	N	30	1700

P-Positive

N-Negative

Microaveraged precision and Macroaveraged precision for the classes show in Table 9 are calculated as follows.

$$\text{Macroaveraged precision} = \frac{20}{20+20} + \frac{85}{85+10} \approx 1.39$$

$$\text{Microaveraged precision} = \frac{105}{105+30} \approx 0.78$$

The performance measures which are explained above cannot be used for evaluating the proposed approach. Since the requirements for this work was driven by a client company, a different performance measure was used for evaluating the proposed approach.

P-result is used as the evaluation metric for validating the proposed approach. P-result is calculated by using Equation 5.

$$P - result\ (\%) = \frac{N1}{N1 + N2} \times 100 \qquad (5)$$

N1 - Total number of user interest identified correctly
N2 - Total number of user interest identified incorrectly

The classifier predicts top 3 categories for each user. If the true category of the user falls under any one of these three predicted categories, then the user category is predicted correctly by the classifier. If the true category of the user does not fall under any of the three predicted categories, then the user category is predicted incorrectly by the classifier. For each user this binary value is calculated. Total number of positive values returned by this method for all the users are represented as “Total number of user interests identified correctly” and the total number of negative values returned by this method is represented as “Total Number of user interests identified incorrectly”. Using these values, P-result is calculated as in Equation 5.

6.3 EVALUATION TEST SCENARIOS

6.3.1 User interest identification using "Session" and "Tweets"

For evaluating the proposed approach, this scenario is tested to understand the effectiveness of the "Sessions". URL filtering are not used in this scenario. This scenario is designed to test the effectiveness of the "Sessions" over the "Tweets". Tweets are collected from the users to identify the user interests. Tweets containing only English words are used in this scenario. For validating this scenario, two data sets are used. Tweets which satisfy the above condition are stored in a file. This data set is represented as "Tweets".

For the second dataset, tweets are grouped based on the posted time to form sessions. This dataset is represented as "Sessions". Recently posted tweets appear first on Twitter. Twitter APIs also provides the tweets in the same order. When the first tweet is received, the posted time of that tweet is stored. Then the next tweet of the same user is acquired. If the posted time of the 2nd tweet is within 1 hour of the 1st tweet. Then the 1st and 2nd tweet are grouped as one session.

This process is continued till the tweets posted within one hour are received. If nth tweet is not within the 1 hour time duration, then the new session starts with the nth tweet as the 1st tweet in that session. This process is continued till all the specified number of tweets is received.

The user interests are identified based on these two data sets. "Session" dataset is validated across four session durations. 30 minutes, 1 hour, 1.5 hours and 2 hours are the session durations considered for the evaluation. These session durations are chosen to analyze the impact of the session duration on the P-result of interest. These data sets are also run for various numbers of users to verify if the number of users has any impact on the results. "Tweets" data set is also validated across 3 sets of users. The mean P-result from these three sets of users are calculated. The results of this evaluation are shown in the Table 10.

Table 10 P-result (%) from “Sessions” data set and “Tweets” data set

Users	**Session 30 minutes**	**Session 1 hour**	**Session 1.5 hours**	**Session 2 hours**	**Tweets**
80	95	95	93.75	92.5	95
120	95.83	95.83	94.17	95	95
160	95.63	96.25	94.37	94.37	95
Mean	95.49	95.69	94.1	93.96	95

The evaluation results show that the “Session” with 1 hour duration performs better than the other test cases. This implies that tweets when grouped as sessions perform better than the single tweets. The analysis of the results is expressed in the bar chart below (Figure 17, Figure 18 and Figure 19).

Figure 17 shows that Session with 1 hour duration performs better than the other sessions. Session with 30 minute duration performs on par with the 1 hour duration. In the case of session with 30 minutes duration, the number of tweets in the session is getting reduced. If the session duration is reduced further, then each session might contain only 1 tweet. Then this test case would become same as “Tweets” test case.

Session with 1.5 hours and 2 hours duration performs less than the other sessions. This implies that when the duration of the session increases, there is a possibility of irrelevant tweets getting grouped as one session. This increases the noisiness in the data. So the P-result goes down considerably.

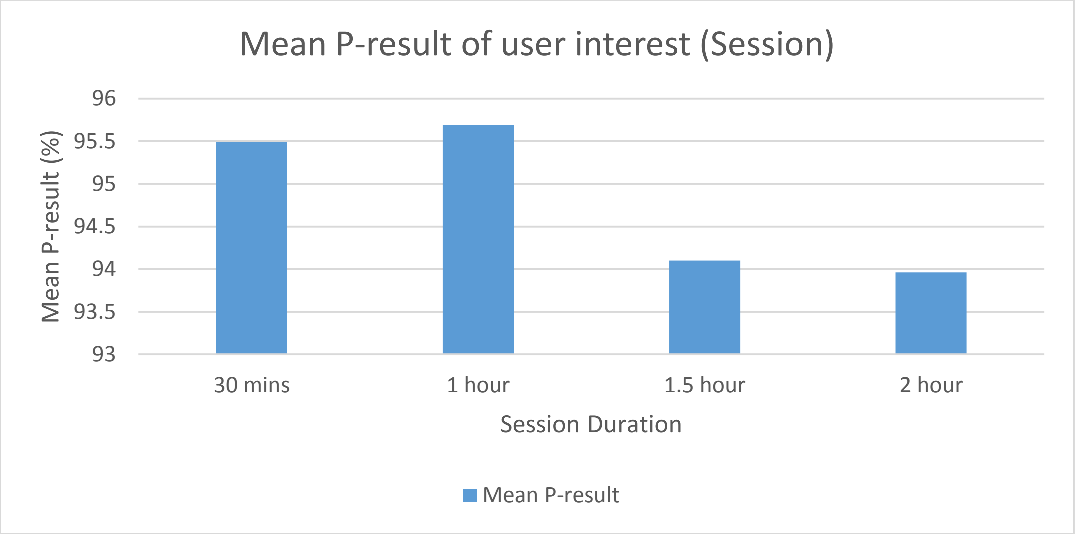


Figure 17 Mean P-result of user interest (Session)

Figure 18 shows the comparison of performance between "Session" and "Tweets". From Figure 18 it is clear that "Session" (95.69%) performs better than "Tweets" (95%).

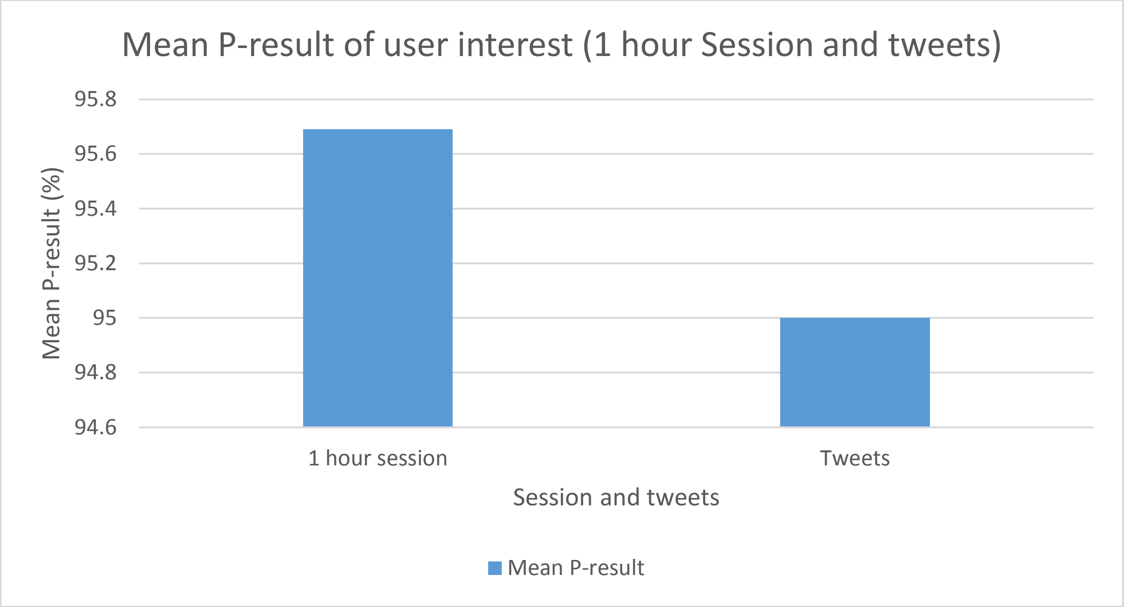


Figure 18 Mean P-result of user interest (1 hour Session and tweets)

Figure 19 shows the P-result of user interest validated across number of users. This test is run for 80, 120 and 160 users. For all the users, “Tweets” performs the same. But in the case of the “Session” with 1 hour duration, the performance increases as the number of users increases.

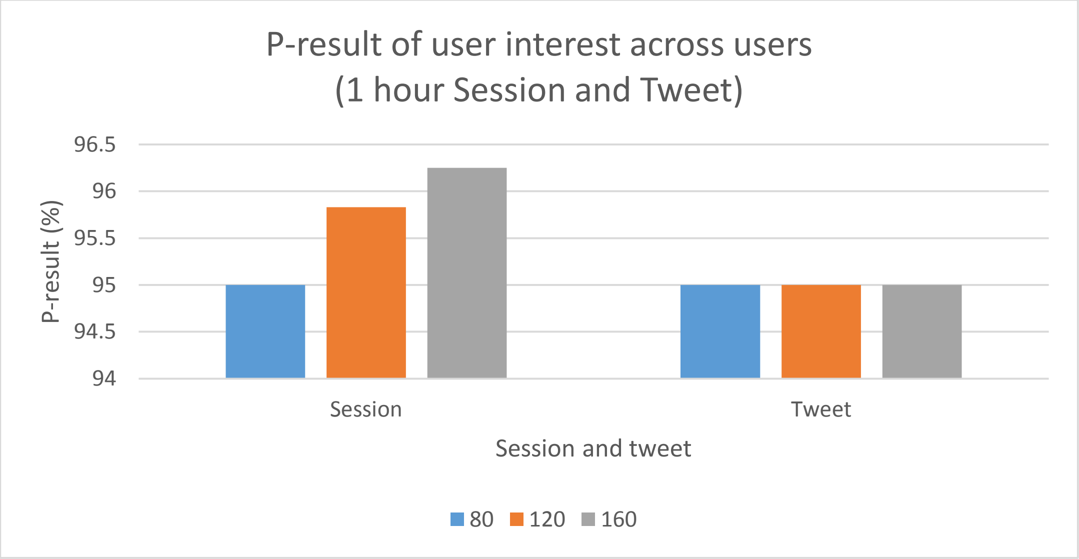


Figure 19 P-result of user interest across users (1 hour Session and Tweet)

6.3.2 User interest identification using URL filtered “Sessions” and “Tweets”

Only tweets with URLs are considered in this scenario. In the previous scenario, no filtering was used. All the tweets were considered. But in this case, URL filtering is used to collect the tweets. Two sets of data are collected. In the first dataset, tweets containing URLs are stored directly. This dataset is represented as “Tweet URL only”. In the second dataset, tweets which are posted within 1 hour time duration are grouped together. The same process which was followed in the last scenario to group tweets is used in this case also. This dataset is represented as “Session URL only”

User interests are identified based on these two data sets. “Session URL only” dataset is validated with various session durations. 30 minutes, 1 hour, 1.5 hours and 2 hours were used as the session duration for validating the “Session URL only” data set. This scenario is also run across various numbers of users. Each session duration is validated across the various numbers of users. The evaluation results for this scenario are presented in Table 11.

Table 11 P-result (%) from “Session URL only” data set and “Tweet URL only” data set

Users	Session 30 minutes	Session 1 hour	Session 1.5 hours	Session 2 hours	Tweets
80	97.5	98.75	98.75	97.5	97.5
120	95.83	97.5	96.67	96.67	96.67
160	96.25	96.88	95.63	95	95.63
Mean	96.53	97.71	97.02	96.39	96.6

“Session URL only” (1 hour duration) method gives more P-result than the other test cases. The analysis of the evaluation results is presented in the bar chart (Figure 20, Figure 21 and Figure 22).

Figure 20 shows the performance of "Session URL only" across various session durations.

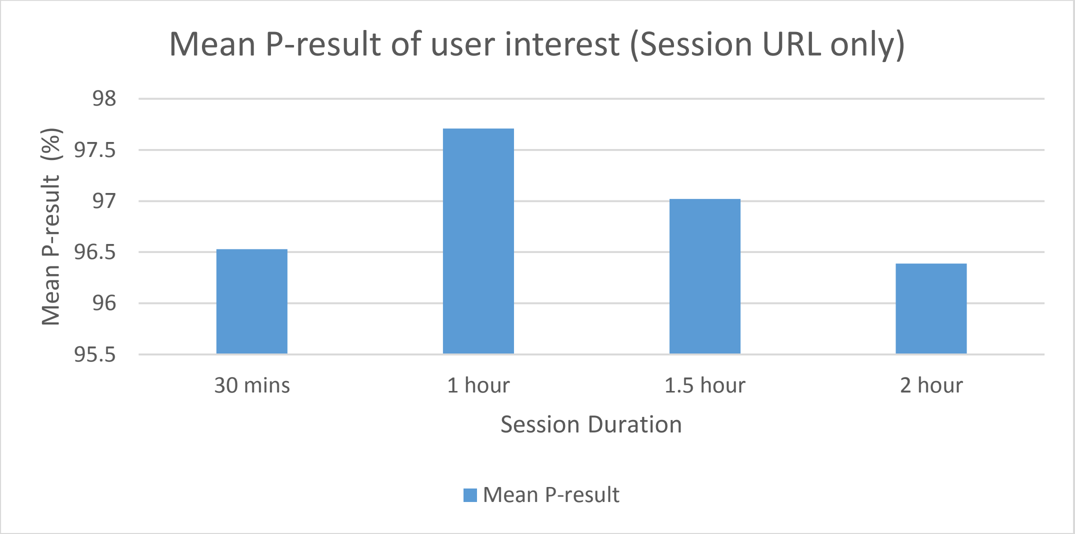


Figure 20 Mean P-result of user interest (Session URL only)

From Figure 20, it is clear that 1 hour session duration performs much better than the other session duration. In all the above four cases, 2 hour Session duration has the least performance. When the session duration increases, the chance of combining irrelevant tweets increases. Thereby increasing the noise in the session. This analysis shows that the session duration has an impact on the user interests generated. This also shows that 1 hour session duration is the appropriate session duration for identifying user interests from sessions.

Figure 21 shows the performance of "Tweet URL only" compared with the "Session URL only" (1 Hour). Since the 1 hour session duration performs better than the other session durations. This 1 hour session is chosen to compare against the "Tweet URL only".

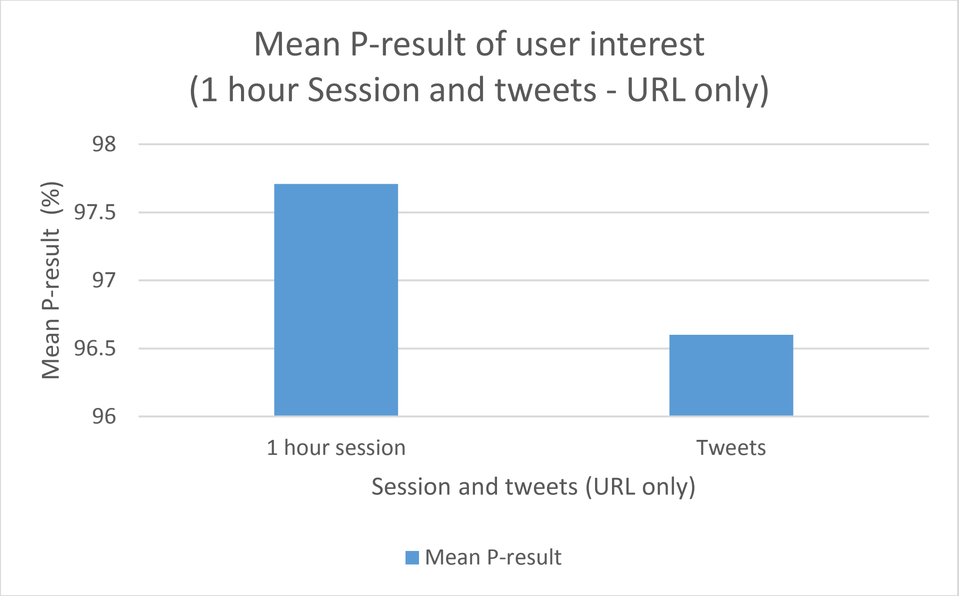


Figure 21 Mean P-result of user interest (1 hour Session and tweets – URL only)

From Figure 21, it is clear that the "Session URL only" performs approximately 1% more than the "Tweet URL only" test case. This confirms that when tweets containing URLs are grouped together to form 1 hour session, the P-result of the user interest increases compared with the user interest identified from tweets containing URLs.

From the experiments it was also found that, on an average, 40% of the acquired tweets contains URLs. All the users in the selected sample had at least one tweets which contains URLs. So, this approach could be used for predicting the interests of most of the users. Figure 22 shows the percentage of users tweeting with URLs.

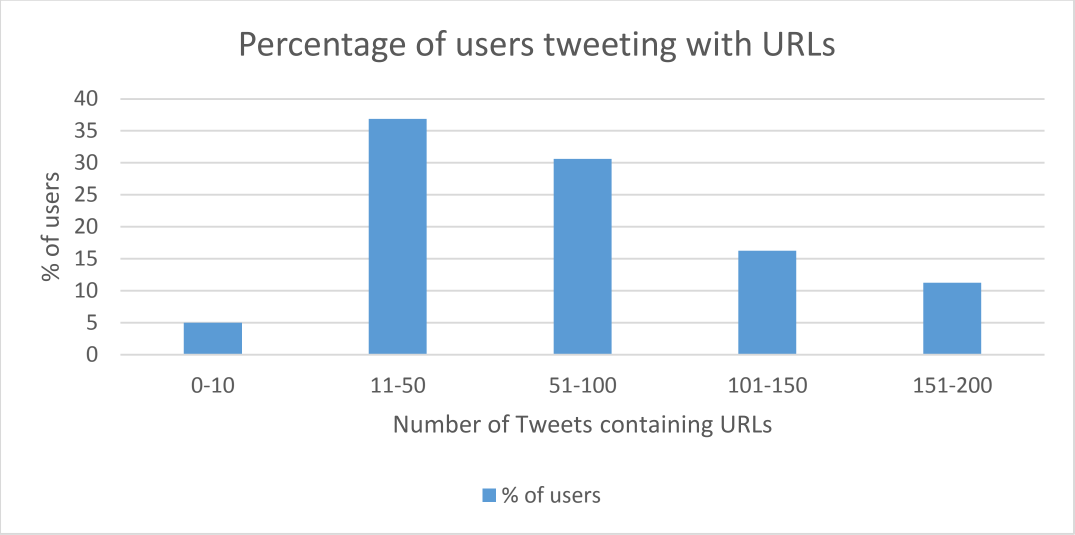


Figure 22 Percentage of users tweeting with URLs

From Figure 22, it is clear that 10% of the sample users tweet 151 to 200 tweets with URL and only 5 % of the sample users tweet in the range of 0 to 10 tweets with URL. So, most of the sample users, tweet with URLs.

Figure 23 shows the P-result of user interest across a different number of users. This case is run for 80, 120 and 160 users.

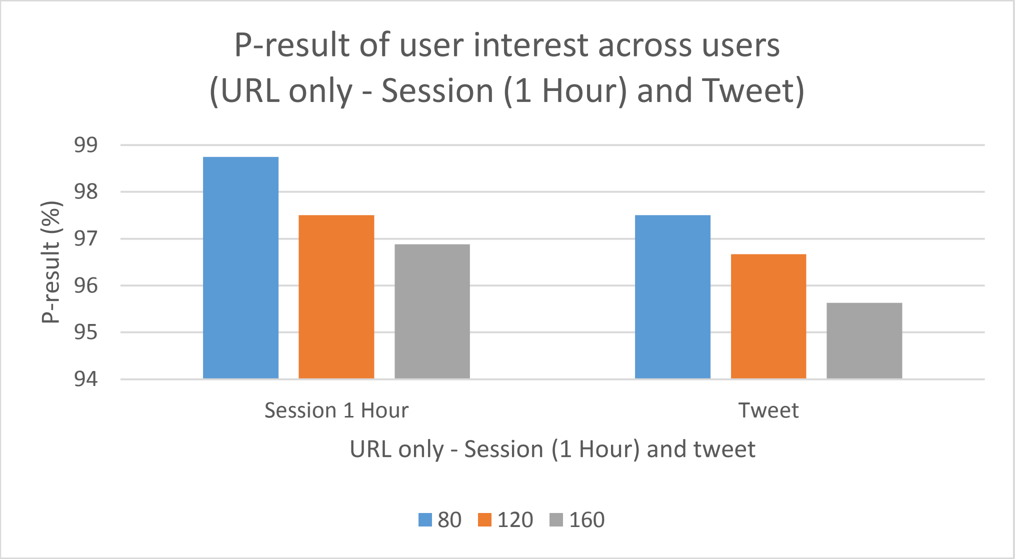


Figure 23 P-result of user interest across users (URL only - Session and Tweet)

From Figure 23, it is clear that when the number of users increases, the P-result decreases gradually in both the cases.

Figure 24 gives an overall analysis of the sessions (1 Hour) and tweets. This chart takes the evaluation results from Table 9 and Table 10. P-result of user interest calculated from “Tweets”, “Tweets URL only”, “Sessions” and “Session URL only” are analyzed in this chart.

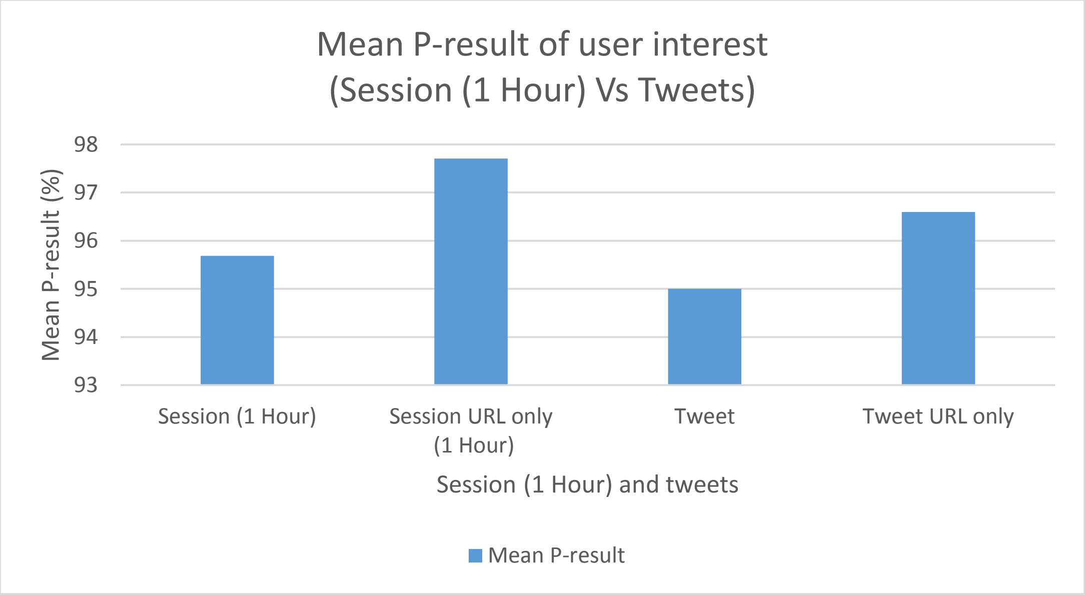


Figure 24 Mean P-result of user interest (Session (1 Hour) Vs Tweets)

From this it is clear that when tweets containing URLs are grouped as sessions with session duration as 1 hour, user interests are identified precisely than other methods. From this analysis, it is also evident that the proposed approach performs better than identifying user interests using single tweets.

6.3.3 User interest identification using activeness of the user as a parameter in "Sessions"

An active user might post more number of tweets than a passive user. So in the case of active user, number of tweets posted within a session would be more. In the previous test scenario, active and passive users were not considered while validating. But in this test scenario, if a user posts more tweets, then for that user, tweets are grouped as sessions. If a user posts less tweets, then the tweets are not grouped as sessions for this user. Single tweets are considered for this user.

If a user tweets more than 2 tweets in a session, that user is considered as an active user. Else that user is treated as a passive user. Tweets were collected from different sets of users (80,120 and 160) and validated for the various session duration (30 minutes, 1 hour, 1.5 hours and 2 hours). This dataset is termed as "Session Active". Another dataset is collected by using URL filters in the tweet collection. The same process used for "Session active" dataset is used in this dataset too. This dataset is termed as "Session URL active". Experiments are conducted on this dataset and the results are shown in the Table 12 and 13.

Table 12 P-result (%) from "Session active" data set

Users	**Session 30 minutes**	**Session 1 hour**	**Session 1.5 hours**	**Session 2 hours**
80	92.5	93.75	92.5	93.75
120	92.5	94.17	93.33	94.17
160	93.75	95	93.75	93.75
Mean	92.92	94.31	93.19	93.89

Table 13 P-result (%) from “Session URL active” data set

Users	Session 30 minutes	Session 1 hour	Session 1.5 hours	Session 2 hours
80	97.5	97.5	97.5	97.5
120	95.83	96.67	96.67	96.67
160	95	95.63	95.62	95.62
Mean	96.11	96.6	96.6	96.6

The analysis of the experimental results is shown in the bar charts below (Figure 25 and Figure 26).

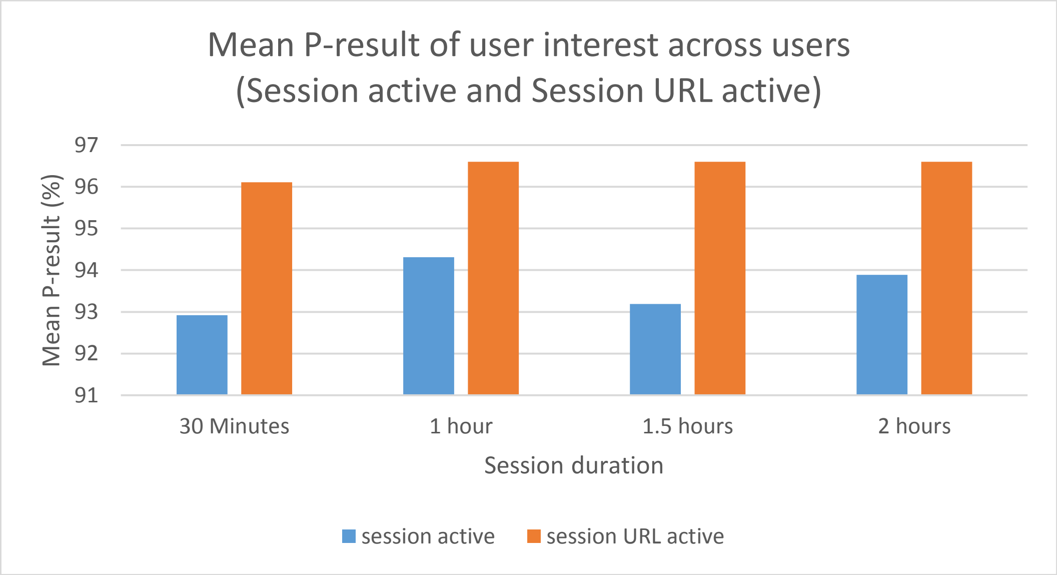


Figure 25 Mean P-result of user interest across users
(Session active and Session URL active)

From figure 25, it is clear that the “session URL active” performs better than the “Session active”. This again shows that the tweets with URLs give precise interest. Irrespective of the session duration, “Session URL active” gives the same P-result. Tweets with URLs within a session are less. Most of the users posts tweets with URLs randomly. So even if the session duration is increased, the same tweets are present in all the session durations.

Figure 26 compares all the methods discussed so far. “Session”, “Session URL only”, “Tweet”, “Tweet URL only”, “Session active” and “Session URL active”.

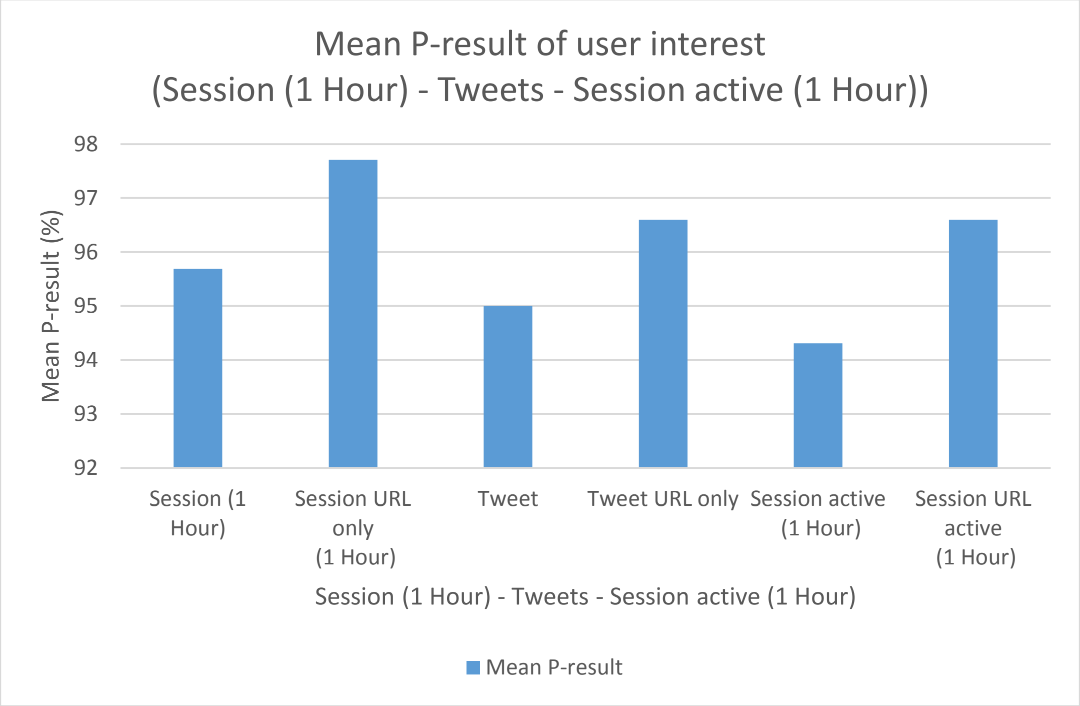


Figure 26 Mean P-result of user interest
(Session (1 Hour) - Tweets - Session active (1 Hour))

From Figure 26, it is clear that “Session URL only” performs better than all the discussed methods.

Both “Session URL active” and “Tweet URL only” methods give the same P-result (96.6%) which shows that both these methods behave in the same way. Most of the sessions in “Session URL active” contain only single tweets, which makes it to behave same as “Tweet URL only”. “Session active” gives the least P-result in the above discussed methods. So using activeness does not provide much help in identifying the interests precisely.

6.3.4 User interest identification using "Twitter List"

Social annotations were used to identify the user interests [4]. Twitter Lists feature in Twitter helps user to group the experts on a particular topic which is of interest to them. Using the title of these Twitter lists, the interests of the users are predicted. Twitter API provides the full name of the Twitter list. These list names are stored and analyzed to identify the categories of these list names. Even though this test scenario is not relevant for the proposed approach, this method has been tested to verify if the Twitter list could be used as a parameter to validate the user interest generated by the proposed approach.

Experiments were conducted to verify if the Twitter lists could identify user interest precisely. 160 users were chosen to validate this scenario. These users were distributed equally among all the categories. Twitter lists were collected from these users using Twitter API. From the experiments conducted, it was found that Twitter list gave a P-result of 74.35%. The analysis of this result is given in the chart below. Figure 27 compares the P-result of the user interest derived from the Twitter list with the "Session URL only" and "Tweet URL only" methods.

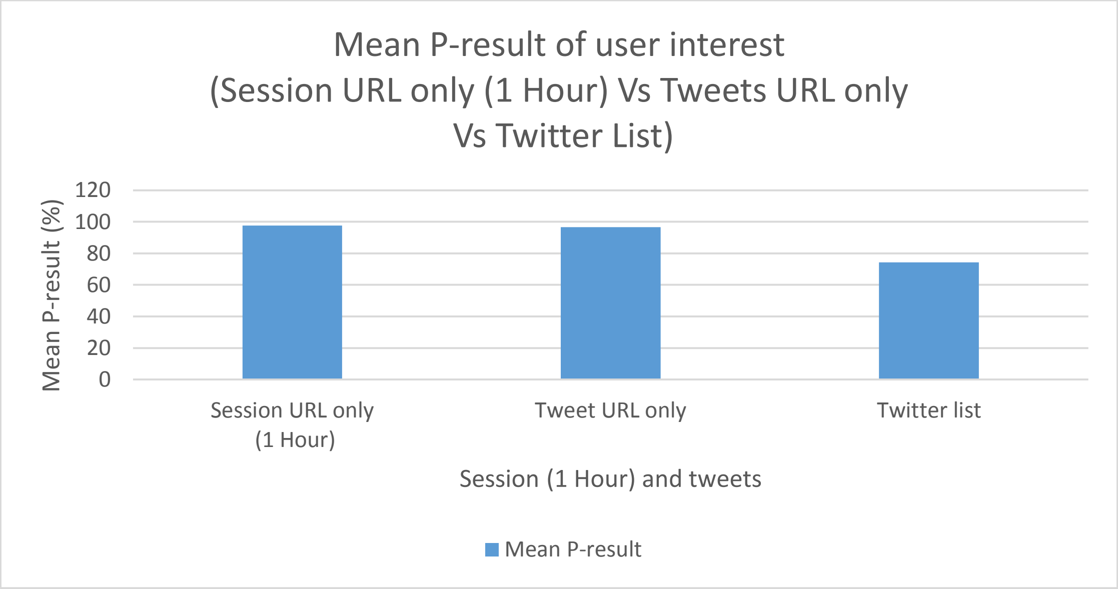


Figure 27 Mean P-result of user interest (Session URL only (1 Hour) Vs Tweets URL only Vs Twitter List)

From figure 27, it is obvious that the Twitter list performs very worse in comparison with the other methods. Moreover, from the experiments, it was noted that only 75% of the chosen users had Twitter Lists. This shows that Twitter List cannot be used as a stand-alone feature for identifying the user interests for all the users. But this could be combined with the other techniques like "Session URL only" or "Tweet URL only" to verify the identified user interest.

Figure 28 shows the percentage of the users having Twitter list distributed across the 8 categories.

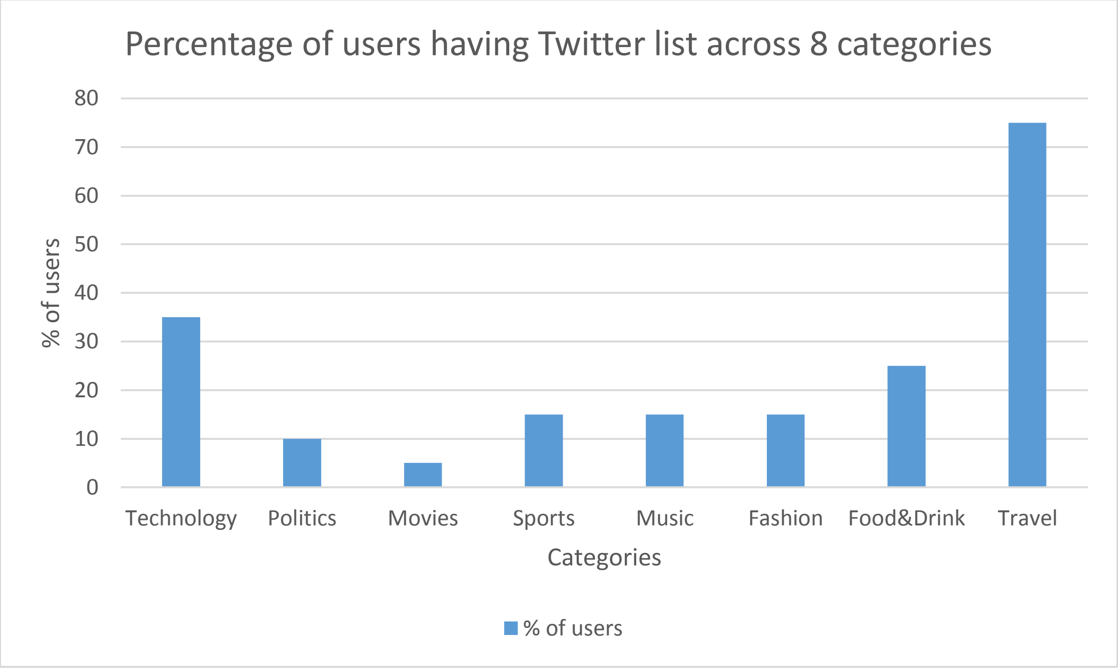


Figure 28 Percentage of users having Twitter list across 8 categories

From Figure 28, it could be inferred that Twitter list is used by more number of users belonging to Travel category than users in the other categories.

6.3.5 User interest identification using “User profile description”

Twitter users can provide a brief description about themselves. This is called as User profile Description. These descriptions could be used to identify the interest of the user. Experiments have been conducted to verify the P-result level of the user interests derived from user profile descriptions. Even though this test scenario is not relevant for the proposed approach, this method has been tested to verify if the “User profile descriptions” could be used as a parameter to validate the user interest generated by the proposed approach. From the experiments, it was found that the interest derived from the user profile descriptions has a P-result of 72.84%. Analysis of this result is given in the chart below (Figure 29 and Figure 30).

Figure 29 compares the P-result of the user interest derived from the Twitter list with the “Session URL only” and “Tweet URL only” methods. From the figure it is clear that the user profile description provides the least P-result compared with the other methods. So this cannot be used individually to identify the user interest.

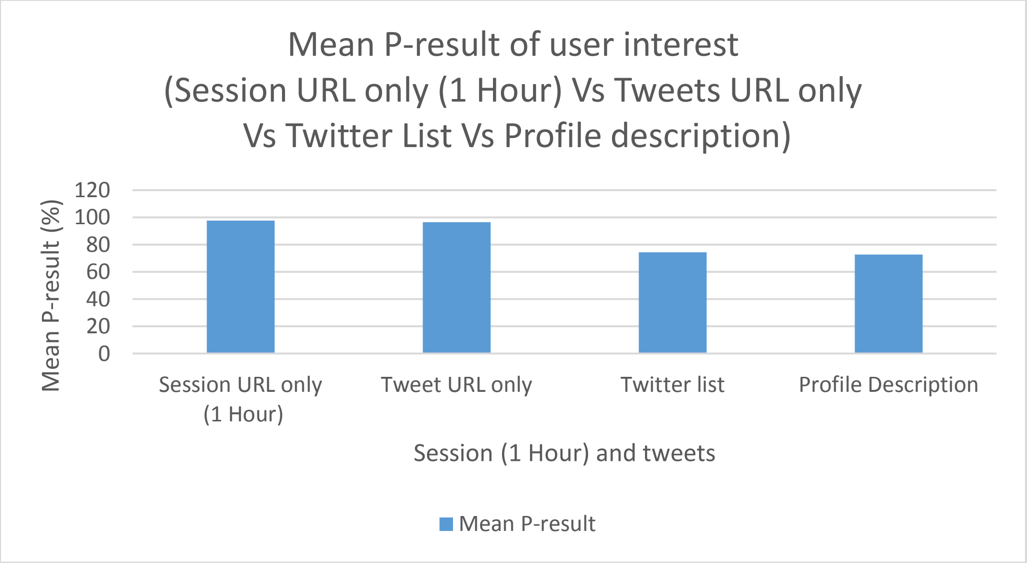


Figure 29 Mean P-result of user interest (Session URL only (1 Hour) Vs Tweets URL only Vs Twitter List Vs Profile description)

Figure 30 shows the percentage of the users having profile descriptions distributed across the 8 categories.

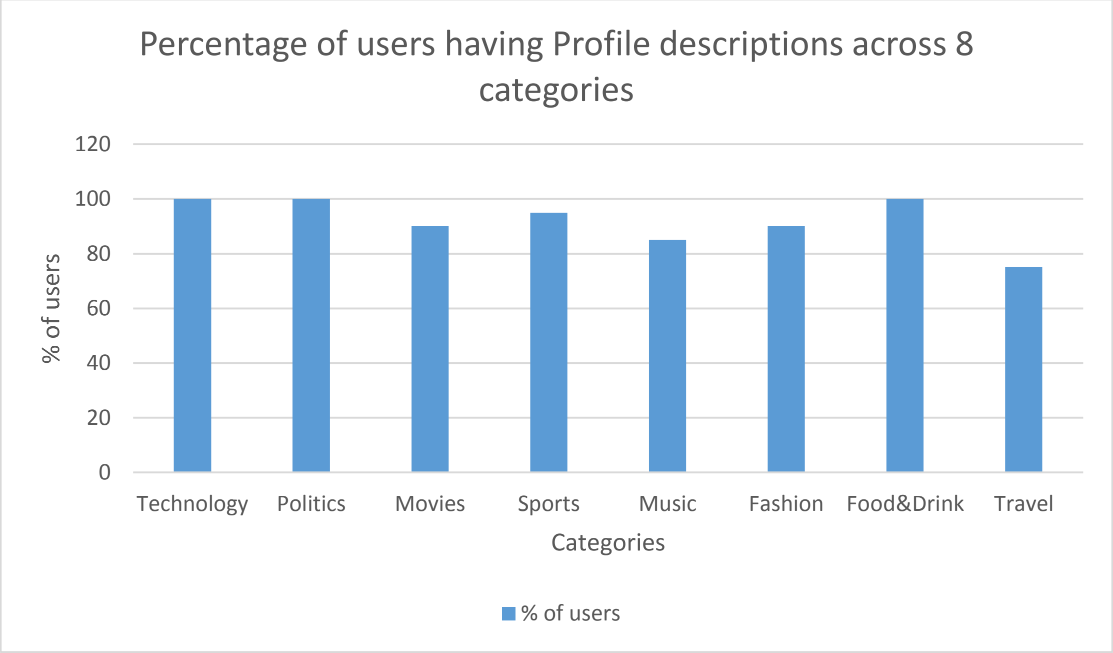


Figure 30 Percentage of users having profile descriptions across 8 categories

From Figure 30, it is clear that the most of the users (selected) have “Profile descriptions” in their Twitter profile. Since all the selected users are celebrities in the respective categories. So most of them have profile descriptions. This needs to be tested with ordinary users to validate this point.

6.4 COMPARISON WITH THE EXISTING APPROACHES

The proposed approach is able to achieve a P-result of 97.71%. Since the existing approaches [1-2, 5, 7] use different measures for evaluating the identified user interest, the proposed approach could not be compared with the existing approaches.

6.5 SUMMARY OF THE EXPERIMENTAL RESULTS

From all the above evaluation results, it is clear that the proposed approach ("Session URL only") performs better than the other methods. This also confirms that, tweets when combined together into sessions, could be used to identify user interests precisely. Performance of the "Session URL only" and "Tweets URL only" methods are better than the "Sessions" and "Tweets". This confirms that the precise user interests could be identified from tweets containing URLs. It is also to be noted that "Twitter List" and "User profile descriptions" could not be used as a stand-alone method to identify user interests. These methods could be integrated with the proposed approach to improve the P-result of the user interest.

CHAPTER 7 CONCLUSION

In this thesis, a hybrid approach has been proposed to identify user interests by grouping tweets containing URLs which are posted within 1 hour. Tweets are short in length. They can have a maximum length of 140 characters. These short texts are difficult to classify compared to a large corpus of text. Many methods have been proposed to overcome this challenge.

Twopics [7] and Hierarchical interest identification [2] used external knowledge bases like Wikipedia to overcome the data sparseness problem. They were able to achieve average precision close to 90%. But the issue in using external knowledge bases is that it takes a considerable amount of time to classify the textual data. This approach cannot be used for real time applications [10]. Most of the existing work like [1, 2, 5, 7] analyses only single tweet to predict the category of the tweet [1]. Since a single tweet contains only a limited number of useful information, classifying the tweets using those words are a challenging task for the classifier.

To overcome these challenges, a new hybrid approach has been proposed in this thesis to identify the user interests precisely. Tweets containing URLs which are posted within 1 hour time duration are grouped together to form sessions. These sessions are then classified into 8 generic categories – Technology, politics, movies, sports, music, fashion, food and travel. Each session may contain one or more tweets. Thereby overcoming the data sparseness challenge which was caused due to the short length of a single tweet. Top three categories which have a maximum frequency in all the sessions is identified as top three interests of that user.

This approach has been implemented and experiments have been conducted to verify the effectiveness of this approach. This approach is able to achieve a maximum precision of 97% (approximately) which is 1% more than the P-result achieved using single tweets.

7.1 LIMITATIONS

Even though this approach gives a good performance compared to the other methods. There are some limitations with this approach. These limitations are discussed below:

1. Noisiness in the session data: In this approach, an assumption has been taken that all the tweets posted within 1 hour are of same context. But this might not be true in all cases. So if irrelevant tweets are grouped together in one single session, then there is a high probability of having noisy data in the sessions.
2. Limited number of users in the user study: Limited number of users have been taken for evaluating this approach. Most of the users chosen for the validation are prominent personalities in their own categories. So it is difficult to generalize this result for common masses. Extensive user study is required to generalize the results.
3. Automatic re-training of the classifiers: A feedback mechanism need to be introduced to send the end results to the classifier to fine tune the classifier models. Some classifier model belonging to a particular category may perform poorly in comparison with the other classifier models. So in that case, if a classifier model predicts the categories wrongly, then a feedback mechanism should be introduced to retrain the classifier model of that particular category.
4. Comparison with other classification algorithms: Regularized logistic regression has been used in this approach to classify the sessions. There are other standard classification algorithms like L-LDA. Those algorithms had to be incorporated in this approach to perform a comparative study on the classification algorithms.

7.2 FUTURE WORK

There is a lot of scope to explore and improve on this approach. One important feature to be improved with this approach is the reduction of the noise level in the sessions. The noisiness in the session could be reduced by combining only relevant tweets in the session. The relevancy can be achieved by combining tweets having similar hashtags. And tweets containing same mentions could also be combined together to improve the tweet relevancy in the session. Another feature to be improved is the introduction of the feedback mechanism for fine-tuning the classifier models. An exhaustive user study has to be conducted to ensure the generalizability of the result.

BIBLIOGRAPHY

[1] S. H. Yang, A. Kolcz, A. Schlaikjer and P. Gupta, "Large-scale high-precision topic modeling on Twitter," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[2] P. Kapanipathi, P. Jain, C. Venkataramani and A. Sheth, "User Interests Identification on Twitter Using a Hierarchical Knowledge Base," in *In The Semantic Web: Trends and Challenges. Springer International Publishing.*, 2014.

[3] R. Ottoni, D. Las Casas, J. Pesce, W. Meira Jr, C. Wilson, A. Mislove and V. Almeida, "Of Pins and Tweets: Investigating how users behave across image-and text-based social networks," in *AAAI ICWSM*, 2014.

[4] P. Bhattacharya, M. Zafar, N. Ganguly, S. Ghosh and K. Gummadi, "Inferring user interests in the Twitter social network," in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014.

[5] C. Budak, A. Kannan, R. Agrawal and J. Pedersen, "Inferring User Interests From Microblogs," Technical report, Microsoft research, 2014.

[6] D. Ramasamy, S. Venkateswaran and U. Madhow, "Inferring user interests from tweet times," in *Proceedings of the first ACM conference on Online social networks*, 2013.

[7] M. Michelson and S. Macskassy, "Discovering users' topics of interest on Twitter: a first look," in *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, 2010.

[8] C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining text data*, Chicago, Springer US, 2012, pp. 163-222.

[9] "Topic model," Wikipedia, The Free Encyclopedia, [Online]. Available: http://en.wikipedia.org/wiki/Topic_model. [Accessed 6 March 2015].

[10] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu and M. Demirbas, "Short text classification in Twitter to improve information filtering," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010.

[11] "The top 500 sites on the web," Alexa, [Online]. Available: http://www.alexa.com/topsites. [Accessed 6 March 2015].

[12] X. Hu and H. Liu, "Text analytics in social media," in *Mining text data*, Springer US, 2012, pp. 385-414.

[13] "Social media," Wikipedia, The Free Encyclopedia, [Online]. Available: http://en.wikipedia.org/wiki/Social_media. [Accessed 06 March 2015].

[14] "Twitter Usage Statistics," Internet Live Stats, [Online]. Available: http://www.internetlivestats.com/twitter-statistics/. [Accessed 6 March 2015].

[15] "Text mining," Wikipedia, The Free Encyclopedia, [Online]. Available: http://en.wikipedia.org/wiki/Text_mining#cite_note-1. [Accessed 6 March 2015].

[16] "An Exhaustive Study of Twitter Users Across the World," Beevolve, [Online]. Available: http://www.beevolve.com/twitter-statistics/. [Accessed 6 March 2015].

[17] "Microblogging," Wikipedia, The Free Encyclopedia, [Online]. Available: http://en.wikipedia.org/wiki/Microblogging. [Accessed 6 March 2015].

[18] "Twitter," Wikipedia, The Free Encyclopedia, [Online]. Available: http://en.wikipedia.org/wiki/Twitter. [Accessed 6 March 2015].

[19] "The story of a Tweet," Twitter, [Online]. Available: https://about.twitter.com/what-is-twitter/story-of-a-tweet. [Accessed 6 March 2015].

[20] "REST APIs," Twitter developers, [Online]. Available: https://dev.twitter.com/rest/public. [Accessed 6 March 2015].

[21] "The Streaming APIs," Twitter developers, [Online]. Available: https://dev.twitter.com/streaming/overview. [Accessed 6 March 2015].

[22] S. Sonali, I. Priya and V. Vaidyanath, "Twist:User Timeline Tweets Classifier," Project report, UC Berkely, 2012.

[23] S. Sonali, I. Priya and V. Vaidyanath, "User Timeline Auto Classifier," GitHub, 2012. [Online]. Available: https://github.com/priya-I/Twist. [Accessed 6 March 2015].

[24] "GET statuses/user_timeline," Twitter developers, [Online]. Available: https://dev.twitter.com/rest/reference/get/statuses/user_timeline. [Accessed 6 March 2015].

[25] "Twitter sentiment analysis using machine learning techniques," Google code, [Online]. Available: https://code.google.com/p/twitter-sentiment-analysis/source/browse/trunk/files/stopwords.txt?r=51. [Accessed 6 March 2015].

[26] "Full-Text Stopwords," MySQL developers, [Online]. Available: http://dev.mysql.com/doc/refman/5.1/en/fulltext-stopwords.html. [Accessed 6 March 2015].

[27] F. Rong-En, C. Kai-Wei, H. Cho-Jui, W. Xiang-Rui and L. Chih-Jen, "LIBLINEAR: A Library for Large Linear Classification," *The Journal of Machine Learning Research,* vol. 9, pp. 1871-1874, 2008.

[28] "Top 20 Twitter fashion accounts to follow," Social media delivered, [Online]. Available: http://www.socialmediadelivered.com/2011/09/12/top-20-twitter-fashion-accounts-to-follow/. [Accessed 6 March 2015].

[29] "Top 20 Twitter Travel Experts," Midlife road trip, [Online]. Available: http://midliferoadtrip.tv/top-20-twitter-travel-experts/. [Accessed 6 March 2015].

[30] "Top Travel Twitter Accounts to Follow," Tripit, [Online]. Available: http://www.tripit.com/blog/2014/09/travel-twitter-accounts-to-follow.html. [Accessed 6 March 2015].

[31] "10 Must-Follow Twitter Accounts for Sports Fans," Mashable, [Online]. Available: http://mashable.com/2012/01/16/twitter-sports/. [Accessed 6 March 2015].

[32] "Top sports Twitter accounts to follow in 2015," Toronto Star Newspapers, [Online]. Available: http://www.thestar.com/sports/2014/12/21/top_sports_twitter_accounts_to_follow_in_2015.html. [Accessed 6 March 2015].

[33] "The Top 100 Must-Follow Sports Business Twitter Accounts Of 2014," Forbes, [Online]. Available: http://www.forbes.com/sites/maurybrown/2014/12/08/the-top-100-must-follow-sports-business-twitter-accounts-of-2014/. [Accessed 6 Mach 2015].

[34] "Top Politicians on Twitter," Fan page list, [Online]. Available: http://fanpagelist.com/category/politicians/view/list/sort/followers_today/. [Accessed 6 March 2015].

[35] "Top 50 Twitter Accounts For The Latest On Technology, Business, Software, and More!," RAD development, [Online]. Available: http://raddevelopment.io/blog/top-50-twitter-accounts-to-follow-for-the-latest-on-technology-business-software-and-more/. [Accessed 6 March 2015].

[36] S. Bihao, "Weak signal detection on twitter datasets: a non-accumulated approach for non-famous events," Doctoral dissertation, TU Delft, Delft University of Technology, 2012.

[37] O. Fabrizio, B. John and P. Alexandre, "Aggregated, interoperable and multi-domain user profiles for the social web," in *Proceedings of the 8th International Conference on Semantic Systems*, 2012.

[38] A. Fabian, G. Qi, H. Geert-Jan and T. Ke, "Semantic enrichment of Twitter posts for user profile construction on the social web," in *The Semanic Web: Research and Applications*, Springer Berlin Heidelberg, 2011, pp. 375-389.

[39] "About Twitter," Twitter, [Online]. Available: https://about.twitter.com/company. [Accessed 6 March 2015].

[40] T. Eran, W. Yang and C. Lorrie Faith, "Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems," *User Modeling and User-Adapted Interaction,* vol. 22, no. 1-2, pp. 203-220, 2012.

[41] G. Baris, "Information filtering on micro-blogging services," Doctoral dissertation, Swiss Federal Institute of Technology Zurich, Institute of Information Systems, 2010.

[42] S. Catarina and R. Bernardete , "Background on Text Classification," in *Inductive Inference for Large Scale Text Classification*, Springer Berlin Heidelberg, 2010, pp. 3-29.

[43] D. L. Olson and D. Delen, "Performance Evaluation for Predictive Modeling," in *Advanced data mining techniques*, Springer Science & Business Media, 2008, pp. 137-147.

[44] C. D. Manning,, P. Raghavan and H. Schutze, "Evaluation of text classification," in *Introduction to information retrieval*, Cambridge, Cambridge university press, 2008, pp. 279-287.