User Profiling with Publicly Available Social Media Data

P.G.R.K.P. Gamlath

User Profiling with Publicly Available Social Media Data

P.G.R.K.P. Gamlath
Index No :11002281

Supervisor: Dr. D.D. Karunaratne

December 2015

Submitted in partial fulfillment of the requirements of the
B.Sc in Computer Science 4th Year Individual Project (SCS4001)

UCSC

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: Mr. P.G.R.K.P. Gamlath

______________________________ ______________________________

Signature of Candidate Date

This is to certify that this dissertation is based on the work of

Mr. P.G.R.K.P. Gamlath

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Supervisor Name: Dr. D.D. Karunaratne

______________________________ ______________________________

Signature of Supervisor Date

Abstract

The expansion of the social media usage we witnessed in the recent past has significantly contributed to the personal information exposure we see today.Due to this new found accessibility to the enormous amount of personal information combined with the advancement of INTERNET search engines, people can easily find data on others. However it can be difficult to infer meaningful information due to the vast amount of available data. Social media sites naturally contain large amounts of data and significant amount of this data tend to be less informative.

As a solution for this we propose an approach based on Text Mining and Natural Language Processing techniques to automate this information extraction process. Our approach collect information from social media feeds and build a secondary profile for a subjected user. To test our approach we used data from micro blogging social media platform Twitter. We chose Twitter because it has a big, active user base and by default Twitter data is public. When compared to other social media networks, Twitter provides easy and unrestricted access to the data.

We evaluate our approach by building profiles for set of random users. As the results suggests our proposed approach can be used for constructing a profile for any person with a digital foot print of fair size. Also as the results suggest this approach can be applied in many domains such as information aggregation, monitoring privacy leaks, monitoring suspect activities and such.

Even though we only used the data from Twitter, the proposed approach can be expanded to use multiple data sources. Which could increase the accuracy and information richness.

Acknowledgment

I am indebted to all the excellent colleagues and mentors that I've interacted during my degree program, without whose support this would not have been possible.

Since the beginning of this project I was fortunate to have the unconditional assistance of several people who have been extremely supportive in various ways. I would like to thank specially my supervisor Dr.D.D. Karunaratne, senior lecturer at University of Colombo School of Computing for the guidance he had given to me throughout the project.

Thanks also goes to my colleagues in the university for their support to my work. Finally, I would like to thank my family who had always given me the freedom and strength.

Contents

List of Figures

List of Tables

List of Acronyms

- MRF - Markov Random Field
- TCRF - Tree-structured Conditional Random Field
- CRF - Conditional Random Field
- NER - named entity recognition
- OOV - Out of Vocabulary
- POS - part of speech
- PTB - Penn Treebank
- KNN - K-Nearest Neighbors
- CMM - Conditional Markov Model
- IOB - Inside-Outside-Begin
- MUC - Message Understanding Conference
- GPS - Global Positioning Service
- PDM - Probability Distribution Model
- RBPDM - Reply Based Probability Distribution Model
- NLP - Natural Language Processing
- SMN - Social Media Network
- API - Application Program Interface
- FOSS - free and open source
- NSA - National Security Agency
- HMM - Hidden Markov Model
- KB - Knowledge Base
- URL - Uniform Resource Locater

Chapter 1

Introduction

Social media has made significant contribution to the public exposure of personal information we see today. The expansion of the social media usage provided means to users to expose their own information and easy access to the information on other users.

Due to this accessibility of the enormous amount of personal information, people can now easily find even most intimate information on almost any person.

The information which accessible via social media can be divide in to two categories.

1. Direct information from the social media profile
   This kind of information is structured and for the most of the time is in one place. It's easy to access and analyze this information.
2. Derived information from the user activities
   This kind of information can't be found in one place and can't be accessed directly. It is scattered all over the user published content such as tweets, Facebook activities, etc . . . . Manually accessing and analyzing this information is not an easy task. Mainly due to the fact that the amount is enormous. It can take a lot of time and effort. Also there's a good chance that the result is subjected to human error.

In this research we focus on this derived kind of information. Because even though manually scanning all this user generated content and extracting useful information from them could result in high accuracy of information, it is time consuming, tedious and prone to be erroneous due to human factor. As a solution for these problems we suggest an approach to automate this process, leveraging the Natural language Processing and Text mining techniques.

In this context, our proposed approach extract the relevant information about a user from his/her social media activities and organize that information in a structured manner for both human and machine consumption. We evaluate our process based on the following two factors.

1. Time consumed for generating the profile
2. Accuracy of the extracted information

1.1 Motivation

Let us assume that someone wants to get information about a particular person. Reason for this can be vary between,

- Getting a rough idea about newly made connection.
- Screening a potential candidate for a job position.
- Finding a suitable candidate for on-line dating.
- Finding potential privacy leaks on self published data.
- etc...

Given the recent INTERNET usage statics, the best place to look for this kind of data is the social media networks. Because there is a high possibility [2] that particular person has an active profile in one or more popular social media networks.

Supposing this process is done manually, when a person want to find information,(supposing he/she's using the social media networks as his/her data sources) first he/she has to find the relevant profiles of that particular person from various social media sites. Then to get the big picture he/she has to go through these multiple profiles and read and analyze their content which most probably scattered through number of pages. Only after manually scanning and analyzing all this data he/she can get a idea about the person they are interested in.

While this manual process can be tedious and time consuming there are many possible social media sources to obtain information such as Facebook account, Twitter account, Google + account, LinkedIn account, etc . . . As these sources are information rich, they contain huge amounts of data. For example in addition to user's public profiles there are freely accessible thousands of their Facebook statuses, Tweets and Google+ activities which may expose interesting insights about profile owners once carefully analyzed.

However as we stated earlier there is a problem when doing this manually. Because as we stated the amount of data is huge. But if there is a way for the computers to extract this information, analyze them and present in a standard way, we can see many potential advantages. Like batch processing all the candidates who applied for a job position or we can process number of user profiles to find a suitable partner for dating in a little time with no effort.

In this context, it motivates us to find a approach of automating the process of extracting the relevant information, analyzing them, classifying them and building up a structured profile based on that information.

This could also help social media users themselves to determine how transparent he/she is to the public in the context of social media since he/she could observe how much of their own data that public can access.

1.2 Aims and Objectives

The main aim of this research is to build and evaluate an approach to construct a digital profile using publicly available data on user activities from social media networks for any given person. For achieving this objective we have addressed following goals.

1. Extract social media activities related to a person from their social media profile.
2. Preprocess the extracted data.
3. Extract and classify Named Entities from the gathered data.
4. Classify the data sentimentally.
5. Combine the sentiments and Named Entities and build a profile for the subjected person.

1.3 Methodology

Our methodology is a combination of several different sub tasks. Firstly a data retrieval module will be developed with closely inspecting the social media APIs. This will enable us to extract as much as public data from social media networks. The collected data will be used to test the proposed approach on real world data. The extracted data is preprocessed and subsequently placed into a repository. Then the proposed algorithms for classifying and entity extraction will be applied to that data. After the the analyzing is done from the extracted information a profile will be produced. For the evaluation purpose we will run set of random user profiles through our implemented prototype and evaluate the generated profiles for each case.

1.4 Scope

The following areas have been considered (In Scope) for the application development,

- Social media is limited to Twitter because of the lack of user activities and limitations imposed from the other social networks themselves for accessing the data.

- The number of tweets used as the input for the approach has a upper limit of 3400. Which is the maximum number of most recent tweets available through twitter API.
- Data gathered/analyzed is limited to textual data.
- Data that will be searched and extracted will be limited to the publicly available data. Public in the sense that relevant user hasn't put any restrictions on accessing that data to a random user.

The following areas are not considered of the project (Out of Scope) and can be later extended as separate projects,

- Extend the data gathering to multiple social media networks.
- Aggregate information from multiple layers (consider the user X's friends' profiles when building the profile for X).
- Analyze non-textual data.

1.5 Organization of the Dissertation Chapter

Rest of the dissertation is organized as follows. Chapter two reviews the background and the existing literature on web profiling, Named entity extraction, Sentiment classification, Web article extraction and twitter mining. Chapter three discusses about the design architecture of the project and the implementation details. Chapter four illustrates the evaluation results and the chapter five concludes the dissertation with the conclusion and a discussion about the future work.

Chapter 2

Background

In this chapter, the existing literature on related topics such as the web user profiling, data extraction from web articles, text classification techniques and Named Entity Recognition algorithms were analyzed in details. In the first part of this chapter web user profiling methods are briefly described and in the second part text classification algorithms, various Named Entity Recognition approaches and twitter specific data extraction methodologies are described. In the third part methods of data extraction from web articles are discussed in details.

2.1 Web User Profiling

Web user profiling have different meaning in different contexts. In this paper we use the term "User Profiling" in the sense, finding, extracting, and fusing the semantic-based user information from the Web.

Web User profiling task can be further divided into three sub categories [3].

- Profile Extraction
- Profile Integration
- User Interest Discovery

2.1.1 Profile Extraction

Profile extraction is focused on identifying the relevant profiles for the given user. There are multiple techniques for this. Most of them are based on theory of MRF (Markov Random Field) [3].

MRF is a probability distribution of labels (hidden variables) that obeys the Markov property. It can be formally defined as follows.

Definition 2.1. *Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a Markov random field in case, when the random variable Y_v obeys the Markov property with respect to the graph:*

$$p(Y_v | Y_w, w \neq v) = p(Y_v | Y_w, w \sim v)$$

where $w \sim v$ means that w and v are neighbors in G.

Tree-structured Conditional Random Fields (TCRFs)

TCRF (Tree-structured Conditional Random Fields) is a CRF(Conditional Random Fields) method which is a special case of MRF. CRF is the conditional probability of a sequence of labels y given a sequence of observations tokens.

2.1.2 Profile Integration

Once the profiles are extracted we have to integrate them when there are multiple profiles are in present. This could be problem because of the ambiguity of the data present in profiles. For example there can be multiple profiles with same name which belongs to different people.

One possible way to solve this ambiguity problem is by using a probabilistic model [3]. This method consists three steps.

1. Data preparation.

Here all the profiles were labeled with common attributes for each profile. For example profile p_i is labeled with p_i.name, p_i.birthday, p_i.email, p_i.location, p_i.school.

2. Formulation using Hidden MRF

   Then these profiles are modeled using a HMM (Hidden Markov Model). The attributes become Hidden Markov Random Fields.

3. Parameter Estimation

   Then the Models are aligned and parameters are estimated.

At the end the profiles with similar parameters were considered similar [3].

2.1.3 User Interest Discovery

Then based on the content on those profiles the user interests were discovered. Extracting Named Entities [4], Sentiment Analysis and Opinion Mining [5] are some of ways to extract the user interests.

2.2 Extract Data from Tweets

Twitter has become one of the most popular microblogging sites for people to broadcast (or "tweet") their thoughts to the world in 140 characters or less. Since by default these messages are available for public consumption, one may expect these tweets to not contain private or incriminating information. Nevertheless it can be observed a large number of users who unwittingly or by purpose post sensitive information about themselves and other people, for whom there may be negative consequences [6].

With a wealth of information being broadcasted publicly by individuals (public can access upto 3400 most recent public tweets of a Twitter user), one may wonder how much sensitive information is contained in these messages. Some may argue that by definition information posted publicly on Twitter cannot be private and that

Twitter users ought to realize that. But in fact it shows that there is a plethora of sensitive information revealed by Twitter users not only about themselves but about other users [6].

Due to this enormous leakage of personal information to the public it has being even suggested using data leak prevent systems combined with social media [7]. In their study they had suggested an approach based on Named Entity Recognition to detect personal data leaks. Furthermore they had proved that their approach is successful on tweets and built a prototype [7] to alert users in real time when their tweet contains personal information.

2.2.1 Extracting Named Entities

Entity extraction, a.k.a. NER (Named Entity Recognition),and text classification are well-known problems that have been around for decades [8]. Because of their importance to a large variety of text-centric applications, these problems have received significant and increasing attention. As the attention grew number of solutions were developed.

But most of the solutions developed are for well-formed English texts. Because of that when most of these well performing solutions used on data from social media, the results have less accuracy. Because in social media the text is usually not well-formed. A key reason for this drop in accuracy is that Twitter contains far more OOV (Out of Vocabulary) words than well-formed, grammatical text.

One of these developed system, the state-of-the-art Stanford POS (part of speech) tagger improves on the baseline, obtaining an accuracy of 80% [9]. This performance is impressive given that its training data, the PTB (Penn Treebank) corpus, is so different in style from Twitter, however this 80% is a huge drop from the 97% accuracy reported on the PTB. [9] Due to unreliable capitalization, common nouns are often misclassified as proper nouns, and vice versa. Also, interjections and verbs are frequently misclassified as nouns.

To overcome this problem one have to identify the special characteristics of texts in social media.

When Tweets are considered the challenges of named entity recognition lie in the insufficient information in a Tweet (context, etc...) and the unavailability of training data. Tweets contain a plethora of distinctive named entity types (companies, products, bands, movies, and more). Almost all these types (except for people and locations) are relatively infrequent, so even a large sample of manually annotated tweets will contain few training examples. Secondly, due to Twitter's 140 character limit imposed on Tweets, they often lack sufficient context to determine an entity's type without the aid of background knowledge. [9]

But it has being shown that by using a combination of global and "Real-Time" Knowledge Base (KB), synergistic combination of the tasks and using the social information as context this insufficient information problem can be overcome [8]. The suggested approach to solve the lack of information is as follows

Building a Global "Real-Time" KB

Since the diversity and the real time nature of Tweets this approach uses Wikipedia as the base for the KB. First the graph structure of Wikipedia is converted into taxonomy, by finding for each concept a single "main" lineage, called the primary lineage. At the same time the other lineages are kept around, to avoid losing information.Then the data is added from other structured sources, such as Chrome (an automobile source), Adam (health), MusicBrainz (albums), City DB, and Yahoo Stocks to improve the wealth of real time data in KB. [8]

Generating Web and Social Contexts

Contexts for Tweets, Users, Hash-tags, and Domains

In its basic form, a Tweet is just 140 characters (and often far fewer than that). To process such short tweets, more context information is needed. Some of that context data is described bellow,

Web context for tweets : If a tweet mentions a URL(Uniform Resource

Locater), The article which contained in that URL was retrieved(if any), extract the title and a snippet (typically the first few lines) of the article, then associate this title/snippet pair with the tweet. So this title/snippet pair can be considered as the Web context of the tweet, since it captures information on the Web related to the tweet. [8]

Social context for users: For each user ID in Twitter, they define a social context that is time dependent. To compute the social context for user U at time t, retrieve the last k tweets of U up to time t (where k is prespecified), tag them, then union the tags and compute average scores. This produces a set of (tag, score) pairs that indicate what user U has often talked about in the last k tweets before time t. [8]

Social context for hashtags and Web domains: In similar way, social contexts can be defined and computed for hash tags and Web domains. To compute the social context for a hash tag h at time t, retrieve k tweets that mention h up to time t, tag them, then union the tags and compute average scores. If a tweet mentions a URL that comes from a Web domain, then we compute a social context for that domain in an analogous fashion. [8]

Contexts for the Nodes in the KB

Similarly, to define and compute Web and social contexts for each node in the KB. To compute a Web context for a node N, retrieve the articles associated with N, tag them, then union the tags and average the scores. To compute a social context for node N, retrieve the last k tweets that mention N, tag the tweets, then union the tags and average the scores. Compute the Web contexts for the nodes in the KB in an off-line process (and refresh these contexts regularly, once every few days). Compute the social contexts for the nodes using the same system that computes social contexts for users, hash tags, and domains, as described earlier. [8]

In a separate study it's also shown a KNN (K-Nearest Neighbors) classifier with a linear CRF model under a semi-supervised learning framework can successfully extract and classify named entities in social media texts. [4]

This method is a combination of three main algorithms

1. Algorithm 1

   Repeatedly adds the new confidently labeled tweets to the training set and retrains itself once the number of new accumulated training data goes above the threshold N. Algorithm 1 also demonstrates one striking characteristic of this method: A KNN classifier is applied to determine the label of the current word before the CRF model. The labels of the words that confidently assigned by the KNN classifier are treated as visible variables for the CRF model.The model is hybrid in the sense that a KNN classifier and a CRF model are sequentially applied to the target tweet, with the goal that the KNN classifier captures global coarse evidence while the CRF model captures the fine-grained information encoded in a single tweet. [4]

2. Algorithm 2

   Outlines the training process of KNN, which records the labeled word vector for every type of label. [4]

3. Algorithm 3

   Describes how the KNN classifier predicts the label of the word. [4]

Stanford POS tagger

This POS tagger is developed based on following these ideas. [1]

- Explicit use of both preceding and following tag contexts via a dependency network representation.
- Broad use of lexical features, including jointly conditioning on multiple consecutive words.
- Effective use of priors in conditional log-linear models.
- Fine-grained modeling of unknown word features

While most of the systems work in a one direction to identify parts of speech this approach works on both ways(right to left and left to right) using dependency networks.

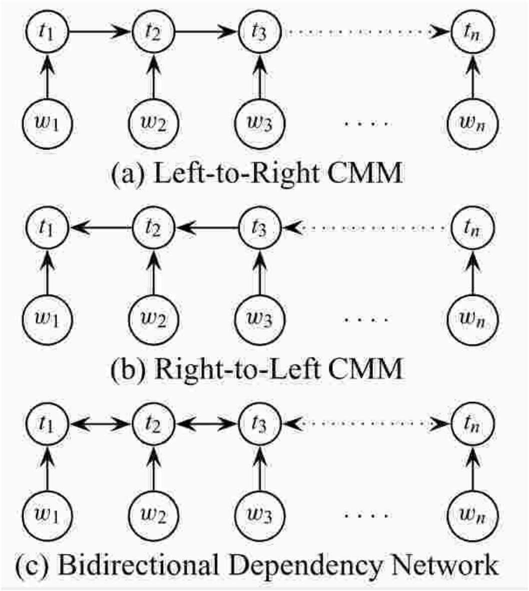

Figure 2.1: Dependency networks: (a) the (standard) left-to-right first-order CMM, (b) the (reversed) right-to-left CMM, and (c) the bidirectional dependency network. Figure taken from [1, p.2]

This setup with broad feature use, when combined with appropriate model regularization, has being reported to give part-of-speech tagger with a per-position tag accuracy of 97.24%, and a whole-sentence correct rate of 56.34% on PTB corpus.

But as mentioned in earlier and explained in next section, accuracy of Stanford POS tagger is much lower when ran on tweets.

Twitter-NLP

This is an implementation of POS tagger and named entity classifier specifically for tweets using annotated tweets and built tools, trained on unlabeled, in-domain and out-of-domain data. On tweets T-POS outperforms the Stanford POS Tagger, reducing error by 41%. [9].

T-SEG models Named Entity Segmentation as a sequence-labeling task using IOB encoding(a format used for the CoNLL-2003 shared task on language-independent named entity recognition) for representing segmentations (each word either begins, is inside, or is outside of a named entity), and uses CRFs for learning and inference. It include orthographic, contextual and dictionary features; dictionaries included a set of type lists gathered from Freebase[1]. In addition, they use the Brown clusters and outputs of T-POS, T-CHUNK and T-CAP in generating features.

The results at segmenting named entities is presented in Table 2.1. Compared with the state-of-the-art news-trained Stanford Named Entity Recognizer, T-SEG obtains a 52% increase in F1[2] score.

Table 2.1: Performance at segmenting entities with varying the features used. ”None” removes POS, Chunk, and capitalization features. Overall twitter-nlp obtain a 52% improvement in F1 score over the Stanford Named Entity Recognizer. Table taken from [9, p.5]

	P	R	F1	F1 inc
Stanford NER	0.62	0.35	0.44	-
T-SEG(None)	0.71	0.57	0.63	43%
T-SEG(T-POS)	0.70	0.60	0.65	48%
T-SEG(T-POS, T-CHUNK)	0.71	0.61	0.66	50%
T-SEG(All Features)	0.73	0.61	0.67	52%

The end to end performance on segmentation and classification (T-NER) is presented in Table 2.2. According to that data comparing against the Stanford Named Entity Recognizer on the 3 MUC types (PERSON, LOCATION, ORGANIZATION), T-NER doubles F1 score.

[1] `https://www.freebase.com/`

[2] F1-harmonic mean of precision and recall

Table 2.2: Performance at predicting both segmentation and classification. Systems labeled with PLO are evaluated on the 3 MUC types PERSON, LOCATION, ORGANIZATION. Table taken from [9, p.8]

System	P	R	F1
COTRAIN-NER (10 types)	0.55	0.33	0.41
T-NER(10 types)	0.65	0.42	**0.51**
COTRAIN-NER (PLO) (10 types)	0.57	0.42	0.49
T-NER(PLO)	0.73	0.49	**0.59**
Stanford NER (PLO)	0.30	0.27	0.29

This POS tagger, Chunker Named Entity Recognizer are open source and available for use by the research community in `http://github.com/aritter/twitter_nlp`

2.2.2 Extracting Locations

Twitter allows its users to specify their geographical location as user information (Meta Data). This location information is manually entered by the user or updated with a GPS (Global Positioning Service) enabled device. The feature to update the user location with a GPS enabled device has not been adopted by a significant number of users [10, 11]. Hence, this geographic location data for most users may be missing or incorrect.

But it's being shown that the Twitter user's city-level geographic location can be discovered from on his/her Tweet content along with the content of the related reply-tweet messages. By using a probabilistic framework that considers a distribution of terms used in the tweet messages of a certain conversation containing reply-tweet messages, initiated by the user.

There are two main probabilistic models

- PDM (Probability Distribution Model):

  This probability distribution technique is as follows. It assume that each user belong to a particular city, and thus his/her tweets also belong to that city.

That is, the terms occurring in the user's tweet can be assigned as terms related to the user's city. This forms the basic distribution of terms for the set of cities considered in the complete data set. The probability distribution of term t over the entire data set, for each city c, is given as

$$p(t|c) = |t|t \in \{terms\ t\ occurs\ in\ city\ c\}|/|t|$$

That is, the number of occurrences of term t for a city c divided by the total occurrences of the term t in the entire dataset. A probability distribution matrix of size n x m is formed, where n is the size of the term list, i.e., the size of the dictionary, and m is the total number of cities in the data set that are considered for evaluation. [10]

- RBPDM (Reply Based Probability Distribution Model):
  In the basic calculation of the PDM, the terms in a user's tweet are assigned to the city to which the user belongs. It does not consider the relation between different tweet messages. [10]

It's also shown that the RBPDM performs roughly twice better than the PDM model. [10]

These findings can be further improved by identifying the local words in tweets [11]

2.2.3 Sentiment Analysis and Opinion Mining on Tweeter data

Twitter provides users with a framework for writing brief, often-noisy postings about their lives. Usually these postings are about one or more named entities. Identifying these entities provide a better chance at creating an information rich profile about the mentioning tweeter.

The real-world nature of Tweets means they are noisy and complex, making the problem difficult. Tweets are intentionally short (limited to just 140-characters) which forces users to be creative in how they constrain the text while preserving meaning. As with text messages in general, this leads to noise. Users rely on common acronyms, disambiguation via context, combinations of the two, and other constraining mechanisms.

However, Tweets can also be information rich, because users tend to pack substantial meaning into the short space.

It's shown that a knowledge base can be used to disambiguate and categorize the entities in the Tweets. Then develop a "topic profile", which characterizes users' topics of interest, by discerning which categories appear frequently and cover the entities. [12]

Classification Techniques

1. Unigram Naive Bayes

   The ultimate task here for the sentiment analyzer is to calculate the probability that tweet d is in class c, where c = 0 or 1. It can be done via two unigram Naive Bayes models. In both of these, the Naive Bayes simplifying independence assumption is used:

   $$P(c|d) = P(c)\Pi 1 \leq k \leq n_d P(tk|c)$$

   Where tk denotes the kth token sequentially in a tweet, and nd is the size of a tweet. The Naive Bayes assumption is that these probabilities for each token are independent, and thus the joint probability is merely the product. [13]

2. Multinomial Bigram Naive Bayes

   In this multinomial bigram Naive Bayes model, which calculated the log probability in a method similar to the multinomial unigram model, but with contrast to the unigram model this model uses bigrams instead of single tokens. [13]

   $$P(c|d) = \alpha P_{unigram}(c|d) + (1 - \alpha)P_{bigram}(c|d)$$

3. Maximum Entropy Classification

   The intuition of the MaxEnt model is to use a set of user-specified features and learn appropriate weights. Combined with an appropriately smoothed Maximum Entropy Classifier that aimed to select feature parameter values to maximize the log-likelihood of the tweet test data we generated. In addition, High weights given to features mean that these are strongly indicative of a certain class. The estimate of P(c—d) for class c and tweet d is given by:

   $$P(c|d) = \frac{1}{Z(d)} exp(\sum_{i} \lambda_{i,c} F_{i,c}(d, c))$$

   Where Z(d) is a normalization function that ensures a proper probability distribution, Fi, c are binary feature functions that give a value for the presence of feature fi in class c in the tweet d, and λ are feature-weight parameters.These parameters are learned to maximize the entropy of the distribution. [13]

Analyzing methodologies

1. Knowledge base method

   Given the huge number of Tweets and Twitter users, discovering topic profiles needs to be done automatically. However, because Tweets are noisy and ambiguous, such automatic analysis is fraught with difficulties. First, their noisy nature makes finding the entities within the Tweets quite challenging, and makes their references noisy. Second, even if the entities are found in the Tweets, they are often ambiguously described, relying on the context of the Tweet and knowledge about the poster to disambiguate the entities.

   It's shown this knowledge base based methodology is successful at tackling the ambiguity problem of interests of the user. [12]

2. TreeTagger method [5]

   This method involves building sentiment classifier using the multinomial Naive Bayes classifier.

   $$P(s|M) \sim P(M|s)$$

where s is a sentiment, M is a Twitter message. Because, we have equal sets of positive, negative and neutral messages

Then train two Bayes classifiers, which use different features: presence of n-grams and part-of-speech distribution information. N-gram based classifier uses the presence of an n-gram in the post as a binary feature. The classifier based on POS distribution estimates probability of POS-tags presence within different sets of texts and uses it to calculate posterior probability. Although, POS is dependent on the n-grams, we make an assumption of conditional independence of n-gram features and POS information for the calculation simplicity:

$$P(s|M) \sim P(G|s) \cdot P(T|S)$$

Finally, calculate log-likelihood of each sentiment:

$$L(s|M) = \sum_{g \in G} log(P(g|s)) + \sum_{\in G} log(P(t|s))$$

By discarding common n-grams the accuracy of the classification can be increased.

2.3 Extracting Data from Web Articles

The objective of Web article mining is to extract explicit and implicit concepts and semantic relations between concepts using Natural Language Processing (NLP) techniques. It's aim is to get insights into large quantities of text data. While this deeply rooted in NLP, it draws on methods from statistics, machine learning, reasoning, information extraction, knowledge management, and others for its discovery process. [14]

Humans have the ability to distinguish and apply linguistic patterns to text and humans can easily overcome obstacles that computers cannot easily handle such as slang, spelling variations and contextual meaning which leads to ambiguity. However, although our language capabilities allow us to comprehend unstructured data, we lack the computer's ability to process text in large volumes or at high speeds.

Mining a web article can expressed by following steps [14]

- Information Extraction

  A starting point for computers to analyze unstructured text is to use information extraction. Information extraction identifies key phrases and relationships within text. It does this by looking for predefined sequences in text, a process called pattern matching. In this step the content of the article is extracted (HTML tags removed) and infers the relationships between all the identified people, places, and time to provide the user with meaningful information. This technology can be very useful when dealing with large volumes of text. [14]

- Topic Tracking - Keyword extraction

  Keywords are a set of significant words in an article that gives high-level description of its contents to readers. Identifying keywords from a large amount of on-line news data is very useful in that it can produce a short summary of news articles.

  After the content of a web page is extracted the in prior step candidate keywords are extracted and thrown to keyword extraction module. And finally keywords are extracted by cross-domain comparison module. [14]

- Summarization

  Text summarization is helpful for trying to figure out whether or not a lengthy document meets the user's needs. And is worth reading for further information. The key to summarization is to reduce the length and detail of a document while retaining its main points and overall meaning.

  An automatic summarization process can be divided into three steps: [14]

  1. In the preprocessing step a structured representation of the original text is obtained

2. In the processing step an algorithm must transform the text structure into a summary structure
3. In the generation step the final summary is obtained from the summary structure.

The methods of summarization can be classified, in terms of the level in the linguistic space, in two broad groups: [14]

1. shallow approaches, which are restricted to the syntactic level of representation and try to extract salient parts of the text in a convenient way
2. deeper approaches, which assume a semantics level of representation of the original text and involve linguistic processing at some level.

- Categorization

  Categorization involves identifying the main themes of a document by placing the document into a predefined set of topics. When categorizing a document, a computer program will often treat the document as a "bag of words." It does not attempt to process the actual information as information extraction does. Rather, categorization only counts words that appear and, from the counts, identifies the main topics that the document covers.

  Using supervised learning algorithms, the objective is to learn classifiers from known examples (labeled documents) and perform the classification automatically on unknown examples (unlabeled documents) [14]

Chapter 3

Design & implementation

In this chapter we try to explain the design and the implementation details of the project.

The proposed methodology is focused on constructing an approach to extract personal attributes from a Twitter user's public Tweets and build a profile using that information. The process can be divided into five different high level components based on the data flow.

1. Extracting the Tweets from the user profile.
2. Preprocess the extracted Tweets.
3. Extract named entities from each Tweet.
4. Classify each Tweet sentimentally.
5. build a profile from the derived named entities and sentiments.

Even though there are separated five components, each (except the first) relies heavily on the output of one or more previous steps.

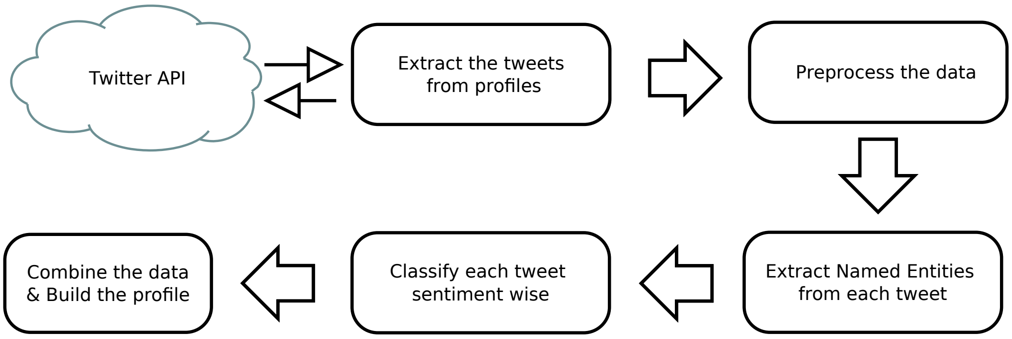
Twitter API
Extract the tweets from profiles
Preprocess the data
Extract Named Entities from each tweet
Classify each tweet sentiment wise
Combine the data & Build the profile

Figure 3.1: Data flow of the system

3.1 Design Assumptions

In this approach we assume there are sufficient content(Tweets) in user profiles to run the proposed algorithms and the contents of those profiles are trustworthy. Additionally we only use the textual data of the Tweets and we assume the text is enough to represent the idea that user is trying to convey from the given Tweet.

3.2 Architecture

Since the broken down parts of the design relies on the previous step's output we propose the following architecture. So all the components from extracting the Tweets from the user profile to finally building the profile from the derived named entities and sentiments are illustrated in the Figure 3.2.

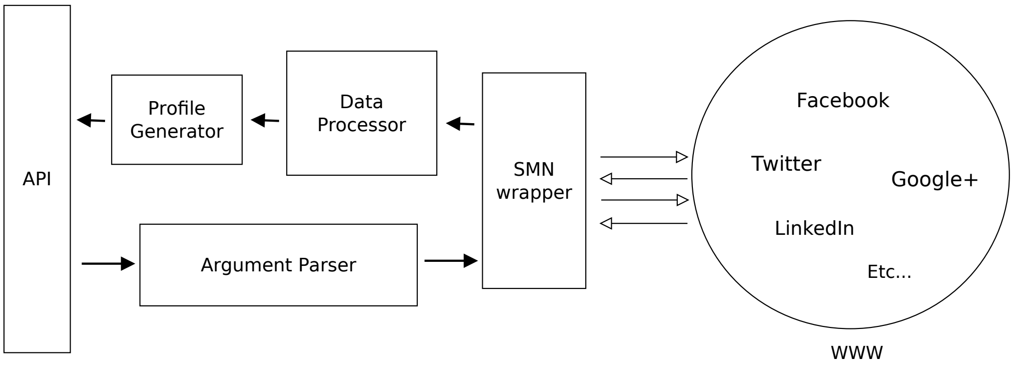

Figure 3.2: Overall design

3.2.1 API (Application Programming Interface)

This is the interface between the user and the implemented system. User is exposed to the set of commands, which once received will be passed to the argument parser by the API. All the user interactions with the system will happen through this module. This way the system can be modified without affecting the user experience.

3.2.2 Argument Parser

In this module user arguments will be parsed and passed to the SMN (Social Media Network) wrapper. Subsequently the invalid arguments will be dropped and user will be notified of the error.

3.2.3 SMN (Social Media Network) wrapper

This is responsible for communicating with the Social media networks, Since social media networks each have there own APIs there are individual sub models for each social media network. For example Twitter submodule we implemented only handles the API calls for the Twitter API. This module also responsible for cleansing and preprocessing the received data (from SMN APIs) and passing them to the respective Sentiment classifier and NER modules.

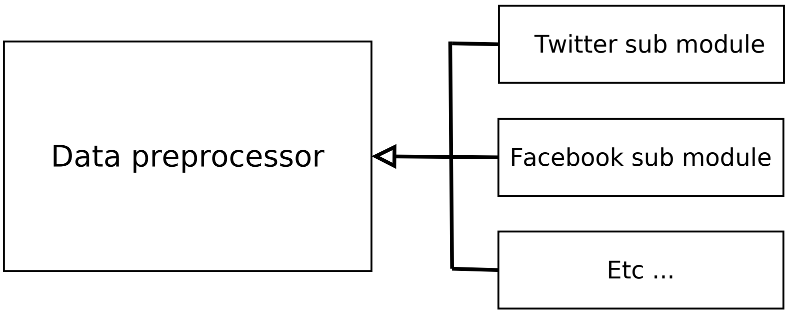

Figure 3.3: SMN wrapper design

3.2.4 Data processor

This the most important part of our system. In this module cleansed and preprocessed data will be fed in to the sentiment classifier and NER extractor. Results will be saved internally to use in the final report.

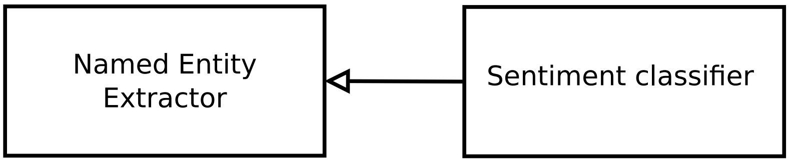

Figure 3.4: Data processor design

3.2.5 Profile generator

After analyzing the received and derived data profile generator will generate a report (user-profile) as the output of the system which will be sent as the response for the requested user.

3.3 Implementation

We used python as the main language for implementation purposes, mainly because of the availability of third party libraries that we going to need for this project and the presence of a strong community support.

3.3.1 Extracting the Tweets from the user profile

In this module we request all the publicly available Tweets of the subjected user from the Twitter public APIs(Twitter only allows the access to the most recent 3400 Tweets).

We configured an application in the Twitter platform to get the access tokens which needed for accessing the Twitter API.

We used third party, open source Twitter client Tweepy[1] to access the Twitter public API because it provides a complete documentation and all the features we require. Each Tweet returned from the Twitter API carry number of attributes including the following attributes.[2]

contributors

An collection of brief user objects (usually only one) indicating users who contributed to the authorship of the Tweet, on behalf of the official Tweet author

coordinates

Represents the geographic location of this Tweet as reported by the user or client application. The inner coordinates array is formatted as geoJSON (longitude first, then latitude).

created_at

UTC time when this Tweet was created.

current_user_reTweet

Only surfaces on methods supporting the include_my_reTweet parameter, when set to true. Details the Tweet ID of the user's own reTweet (if existent) of this Tweet.

entities

Entities which have been parsed out of the text of the Tweet.

favorite_count

Indicates approximately how many times this Tweet has been "liked" by Twitter users.

favorited

Indicates whether this Tweet has been liked by the authenticating user.

filter_level

Indicates the maximum value of the filter_level parameter which may be used

[1] http://www.tweepy.org/

[2] https://dev.twitter.com/overview/api/tweets

and still stream this Tweet. So a value of medium will be streamed on none, low, and medium streams.

id

The integer representation of the unique identifier for this Tweet. This number is greater than 53 bits and some programming languages may have difficulty/silent defects in interpreting it. Using a signed 64 bit integer for storing this identifier is safe. Use id_str for fetching the identifier to stay on the safe side.

in_reply_to_screen_name

If the represented Tweet is a reply, this field will contain the screen name of the original Tweet's author.

in_reply_to_status_id

If the represented Tweet is a reply, this field will contain the integer representation of the original Tweet's ID.

lang

When present, indicates a BCP 47 language identifier corresponding to the machine-detected language of the Tweet text, or "und" if no language could be detected.

place

When present, indicates that the Tweet is associated (but not necessarily originating from) a Place.

possibly_sensitive

This field only surfaces when a Tweet contains a link. The meaning of the field doesn't pertain to the Tweet content itself, but instead it is an indicator that the URL contained in the Tweet may contain content or media identified as sensitive content.

quoted_status_id

This field only surfaces when the Tweet is a quote Tweet. This field contains the integer value Tweet ID of the quoted Tweet.

quoted_status

This field only surfaces when the Tweet is a quote Tweet. This attribute contains the Tweet object of the original Tweet that was quoted.

scopes

A set of key-value pairs indicating the intended contextual delivery of the containing Tweet. Currently used by Twitter's Promoted Products.

reTweet_count

Number of times this Tweet has been reTweeted. This field is no longer capped at 99 and will not turn into a String for "100+"

reTweeted

Indicates whether this Tweet has been reTweeted by the authenticating user.

text

The actual UTF-8 text of the status update.

The entity attribute itself is a combination of following attributes.

media

An array of media attached to the Tweet with the Twitter Photo Upload feature.

urls

An array of URLs extracted from the Tweet text.

user_mentions

An array of Twitter screen names extracted from the Tweet text.

hashtags

An array of hashtags extracted from the Tweet text.

symbols

An array of financial symbols starting with the dollar sign extracted from the Tweet text.

extended_entities

This nested object supports various media types such as multi photos, animated gifs and video. This field also contains a lot of meta data about the type of information that is present in there; such as aspect ratio, variants, sizes, duration, bitrate etc.

Since most of these attributes has no use for our purpose we process those Tweets and keep only the text value, created_at value, entities and optional coordinates value of each Tweet. Once processed we store the Tweet in a local data repository as json objects for use in next steps.

3.3.2 Preprocessing the data

In this module we prepare the Tweets to feed in to the NER and Sentiment Analysis processes. This process is done to remove the noise from the Tweet text.

First if there is an URL present in the tweet we try to extract the the article from it. Then add the summary of the article to the tweet text. Because there's a high probability that article also relate to the main idea of the tweet. We use python-goose FOSS (free and open source) package for this article extraction purpose.

Then remove the URL(Uniform Resource Locater)s, user-names and hash-tags mentioned in the Tweet text attribute using the values present in the entities attribute. Then we create a new attribute "pure_text" and save the stripped Tweet text. After this process each Tweet object is represented with following attributes.

created_at

UTC time when this Tweet was created.

entities

Entities which have been parsed out of the text of the Tweet.

coordinates

Represents the geographic location of this Tweet as reported by the user or client application. The inner coordinates array is formatted as geoJSON (longitude first, then latitude).

text

The actual UTF-8 text of the status update.

pure_text

The actual UTF-8 text minus urls, user_mensions and hash-tags.

And according to the new findings the data repository is also updated.

3.3.3 Extract the Named Entities

In this module we extract and classify the named entities using the pure_text attribute of each Tweet. We use the T-NER [9, 15], named entity extractor and classifier system which is custom tailored for extracting and classifying the entities from Tweets. We chose T-NER over the Stanford POS tagger because even-though Stanford POS tagger yields higher accuracy for standard texts, T-NER out performs Stanford POS tagger, on Tweet processing. This is discussed in details in section 2.2.1.

Then we update the each Tweet object in data repository with a additional attribute "ne". There we store the list of extracted and classified named entities of each Tweet.

After completion of this step each Tweet in the data repository is updated with following additional attribute.

ne

List of extracted named entities classified in to categories.

3.3.4 Sentiment classification

In this module we classify all the Tweets according to the sentiment of pure_text attribute of each Tweet. For building our classifier we used an implementation of the Naive Bayes algorithm provided with python NLTK package.

For the training and evaluation we used SemEval data set [16] combined with a subset of sentiment140 data set [17].We chose 5000 positive tweets and 5000 negative Tweets. We combined two data sets because there were only 2307 negative Tweets in the semeval data set. So we added 2693 negative tweets from the sentiment140 dataset to get a even 5000 Tweets on both positive and negative data sets.

From this dataset we used 80% of Tweets to train and 20% of Tweets to evaluate the classifier model. At the evaluation phase our classifier rated a 87% of accuracy.

After classifying each tweet we updated each Tweet object by adding the sentiment value and the probability for that, of the pure_text attribute as the following additional attribute. Possible values for this attribute are "pos" which stands for positive and "neg" for negative.

sentiment

Sentiment value and the probability of pure_text returned by the classifier.

3.3.5 Building the Profile

When we come to this stage our data repository should contain Tweet objects with following seven attributes.

created_at

UTC time when this Tweet was created.

entities

Entities which have been parsed out of the text of the Tweet.

coordinates

Represents the geographic location of this Tweet as reported by the user or client application. The inner coordinates array is formatted as geoJSON (longitude first, then latitude).

text

The actual UTF-8 text of the status update.

pure_text

The actual UTF-8 text minus urls, user_mensions and hash-tags.

ne

List of extracted named entities classified in to categories.

sentiment

Sentiment value of pure_text returned by the classifier.

Then we analyze all the Tweet objects of the user and extract most frequent co-ordinates, user mentions, classified named entities with the sentiment value of the Tweet and categorize all this data along with time, extracted from each Tweet. Thus building a secondary profile for the user.

Following is an example of processed Tweet object.

```
{
    ”sentiment”: {
      ”prob”: 0.5410714285714283,
      ”label”: ”pos”
    },
    ”text”: ”RT @xkcdComic: The Three Laws of Robotics
       https://t.co/b3bVHENIUT https://t.co/7fuEF1X6s9
       https://t.co/wOdJtH8ErF”,
    ”created_at”: ”Mon Dec 07 09:31:54 +0000 2015”,
    ”ne”: {
      ”product”: [
        ”Robotics”
      ]
    },
    ”coordinates”: null,
    ”entities”: {
      ”symbols”: [],
      ”user_mentions”: [
        {
          ”id”: 2251623492,
          ”indices”: [
            3,
            13
          ],
        ”id_str”: ”2251623492”,
        ”screen_name”: ”xkcdComic”,
        ”name”: ”XKCD Comic”
      }
    ],
    ”hashtags”: [],
    ”urls”: [
      {
        ”url”: ”https://t.co/b3bVHENIUT”,
        ”indices”: [
          42,
          65
        ],
        ”expanded_url”: ”http://xkcd.com/1613/”,
        ”display_url”: ”xkcd.com/1613/”
      },
      {
        ”url”: ”https://t.co/7fuEF1X6s9”,
        ”indices”: [
          66,
          89
        ],
        ”expanded_url”: ”http://m.xkcd.com/1613/”,
        ”display_url”: ”m.xkcd.com/1613/”
      }
    ],
    ”media”: [
      {
        ”source_user_id”: 2251623492,
        ”source_status_id_str”: ”673773749057941505”,
        ”expanded_url”: ”http://twitter.com/xkcdComic/
           status/673773749057941505/photo/1”,
        ”display_url”: ”pic.twitter.com/wOdJtH8ErF”,
        ”url”: ”https://t.co/wOdJtH8ErF”,
        ”media_url_https”: ”https://pbs.twimg.com/media/
           CVm5p5JUEAA0C9O.png”,
        ”source_user_id_str”: ”2251623492”,
        ”source_status_id”: 673773749057941505,
        ”id_str”: ”673773748634259456”,
        ”sizes”: {
          ”small”: {
            ”h”: 311,
            ”resize”: ”fit”,
            ”w”: 340
          },
          ”large”: {
            ”h”: 570,
            ”resize”: ”fit”,
            "w": 622
          },
          "medium": {
            "h": 549,
            "resize": "fit",
            "w": 600
          },
          "thumb": {
            "h": 150,
            "resize": "crop",
            "w": 150
          }
        },
        "indices": [
          90,
          113
        ],
        "type": "photo",
        "id": 673773748634259456,
        "media_url": "http://pbs.twimg.com/media/CVm5p5
           JUEAA0C9O.png"
      }
    ]
  },
  "pure_text": "RT  The Three Laws of Robotics"
}
```

Listing 3.1: Processed tweet object example

Chapter 4

Results and Evaluation

Since the final result of our proposed methodology is a user profile and it's highly subjective for selected users, we used a user based evaluation process. For each profile we sent to the users to get the feedback we supplied a questionnaire.

4.1 User Evaluation

Given the nature of this research, the subjected user is the most suitable person to decide whether to accept the generated profile or to reject it. The questionnaire we provided contained questions to measure the accuracy, timeliness and completeness of the details in the generated profile.

First we randomly chose a set of 100 twitter accounts and 100 most followed twitter accounts with atleast 1000 public tweets. When choosing these accounts we tried to cover different demographic and ethnic settings. We tried to select user accounts with different demographic settings. Then we generated a profile per each user using our prototype application and sent the profile attached with the questionnaire as a direct message via twitter. Then for the evaluation purpose we randomly selected 20 feedbacks, 10 most followed accounts and 10 normal tweeter users.

4.2 Results

The full content of Results of the questionnaire could be found in Appendix B.

4.2.1 Time

The most time required process is the named entity extraction process. It took about average of 70% of total time. Tweet extraction phase took the second most portion of the time. Mainly because the time restrictions imposed by the twitter API.

4.2.2 Accuracy of Sentiment Analysis

The Naive Bayes sentiment classifier we used reported an accuracy of 87% at the classifier evaluation phase. We used SemEval data set [16] diluted with the sentiment140 data set [17] for training and evaluation of the classification. Total of 10,000 tweets were used in the training process (5,000 positive and 5,000 negative). 2,000 tweets (1,000 positive and 1,000 negative) tweets used for the evaluation.

For the user evaluation purpose we asked users to rate the sentimentally classified data in the generated profile. Following are the user feedbacks.

A - Highly satisfactory, B - Satisfactory, C - Neutral, D - Unsatisfactory, E - Highly Unsatisfactory

Table 4.1: Accuracy of Sentiment Classification

	A	B	C	D	E	Total
Number of responses	2	14	4	0	0	20
As a precentage	10%	70%	20%	0%	0%	100%

80% users replied that the data is sentimentally correct. Other 20% were neutral about this. No one has given negative feedback on this.

4.2.3 Accuracy of Named Entity Recognition

For the entity extraction we used T-NER [9,15], named entity extractor and classifier system which is custom tailored for extracting and classifying the entities from Tweets.

Following are the user feedbacks on accuracy of the Named Entity Recognition.

A - Highly satisfactory, B - Satisfactory, C - Neutral, D - Unsatisfactory, E - Highly Unsatisfactory

Table 4.2: Accuracy of Named Entity Recognition

	A	B	C	D	E	Total
Number of responses	2	15	3	0	0	20
As a precentage	10%	75%	15%	0%	0%	100%

Here 85% users replied that the Named Entities are correctly identified. Other 15% were neutral about this. No one has given negative feedback on this.

Following are the user feedbacks on accuracy of the Named Entity Classification.

A - Highly satisfactory, B - Satisfactory, C - Neutral, D - Unsatisfactory, E - Highly Unsatisfactory

Table 4.3: Accuracy of Named Entity classification

	A	B	C	D	E	Total
Number of responses	1	8	7	4	0	20
As a precentage	5%	40%	35%	20%	0%	100%

Though 85% said that named entities were identified correctly only 45% is positive that they were classified correct. 35% were neutral and other 20% is not satisfied with named entity classification.

4.2.4 Timeliness of generated profile

Following are the user feedbacks on accuracy of the timeliness of the generated profile.

A - Highly satisfactory, B - Satisfactory, C - Neutral, D - Unsatisfactory, E - Highly Unsatisfactory

Table 4.4: Accuracy of timeliness

	A	B	C	D	E	Total
Number of responses	3	17	0	0	0	20
As a precentage	15%	85%	0%	0%	0%	100%

Here 100% users replied that the timeliness of generated data is correct.

Chapter 5

Conclusions and Future Work

5.1 Discussion

In the recent years, information retrieval had become a popular topic in academic field. Mainly due to the fact that the availability of huge amounts of raw data. Several interesting works were done in Information retrieval from the web and social media. The applications are widely used in a widespread range such as target marketing, building crowd sourced data repositories, identifying natural disasters and even unauthorized citizen surveillance projects run by government bodies like NSA.

The main contribution of this research is a scalable approach to extract meaningful information from social media platforms by combining the natural language processing and text mining techniques. According to the chapter four our suggested methodology yields positive results. We were able to generate a curated profile for twitter users by only analyzing there public Twitter feed.

Most significant short coming we found was classifying the extracted named entities. The main reason for this is the vast amount of entities represented in tweets in many different ways and most of the time they belong to more than one category.

We tried this system on single core processor 4 threads running which had 2GB RAM. We were able to scan the whole(at most 3,400 tweets) twitter feed and generate a profile for user under 3 minutes. Named entity recognition took about 70% of time of this. Given the manually processing this amount of data could take hours to complete, our scored time is better in magnitudes.

Even though we only used twitter as data source our approach can easily take inputs from other social media platforms. Which could lead to a more accurate and information rich user profile.

5.2 Future Work

Our proposed methodology shows that it's possible to use text mining techniques and natural language processing techniques on social media streams for deriving information from social media feeds.

We can expect that it would be possible to generate a more complete user profiles in the the future by combining multiple data sources like other social network feeds, personal blog pages, etc.... And it's also possible to combine information from multiple layers like subjected user's followers'/friends' social media feeds. Also it seems it's possible to add images and videos to the textual content for the mining purposes.

Bibliography

[1] K. Toutanova, D. Klein, and C. D. Manning, “Feature-rich part-of-speech tagging with a cyclic dependency network,” *In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 (NAACL ’03),*, pp. 252–259, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=1073478

[2] N. B. E. Maeve Duggan, A. L. Cliff Lampe, and M. Madden. (2015) Social media update 2014. [Online]. Available: http://www.pewinternet.org/2015/01/09/social-media-update-2014/

[3] J. Tang, L. Yao, D. Zhang, and J. Zhang, “A combination approach to web user profiling,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 1, p. 2, 2010.

[4] X. Liu, S. Zhang, F. Wei, and M. Zhou, “Recognizing named entities in tweets,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1.* Association for Computational Linguistics, 2011, pp. 359–367.

[5] A. Pak and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining.” *LREC*, vol. 10, pp. 1320–1326, 2010.

[6] H. Mao, X. Shuai, and A. Kapadia, “Loose tweets: an analysis of privacy leaks on twitter,” in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society.* ACM, 2011, pp. 1–12.

[7] J. M. Gómez-Hidalgo, J. M. Martín-Abreu, J. Nieves, I. Santos, F. Brezo, and P. G. Bringas, “Data leak prevention through named entity recognition,” in *Proceedings - SocialCom 2010: 2nd IEEE International Conference on Social Computing, PASSAT 2010: 2nd IEEE International Conference on Privacy, Security, Risk and Trust*, 2010, pp. 1129–1134.

[8] A. Gattani, A. Doan, D. S. Lamba, N. Garera, M. Tiwari, X. Chai, S. Das, S. Subramaniam, A. Rajaraman, and V. Harinarayan, “Entity extraction, linking, classification, and tagging for social media,” *Proceedings of the VLDB Endowment*, vol. 6, no. 11, pp. 1126–1137, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2536222.2536237

[9] A. Ritter, S. Clark, Mausam, and O. Etzioni, “Named entity recognition in tweets: An experimental study,” in *EMNLP*, 2011.

[10] S. Chandra, L. Khan, and F. B. Muhaya, “Estimating twitter user location using social interactions–a content based approach,” in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on.* IEEE, 2011, pp. 838–843.

[11] Z. Cheng, J. Caverlee, and K. Lee, “You are where you tweet: a content-based approach to geo-locating twitter users,” in *Proceedings of the 19th ACM international conference on Information and knowledge management.* ACM, 2010, pp. 759–768.

[12] M. Michelson and S. A. Macskassy, “Discovering users’ topics of interest on twitter: a first look,” in *Proceedings of the fourth workshop on Analytics for noisy unstructured text data.* ACM, 2010, pp. 73–80.

[13] R. Parikh and M. Movassate, “Sentiment analysis of user-generated twitter updates using various classification techniques,” *CS224N Final Report*, pp. 1–18, 2009.

[14] V. Gupta and G. S. Lehal, “A survey of text mining techniques and applications,” *Journal of emerging technologies in web intelligence*, vol. 1, no. 1, pp. 60–76, 2009.

[15] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from twitter," in *KDD*, 2012.

[16] S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov, "Semeval-2014 task 9: Sentiment analysis in twitter," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, August 2014, pp. 73–80. [Online]. Available: http://www.aclweb.org/anthology/S14-2009

[17] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *Processing*, vol. 150, no. 12, pp. 1–6, 2009. [Online]. Available: http://www.stanford.edu/{~}alecmgo/papers/TwitterDistantSupervision09.pdf

Appendices

A: Questionnaire

1. Correlation between entities, that I'm interested and listed in the profile are
    - Highly satisfactory
    - Satisfactory
    - Neutral
    - Unsatisfactory
    - Highly unsatisfactory
2. Classification correctness of entities listed in the profile are
    - Highly satisfactory
    - Satisfactory
    - Neutral
    - Unsatisfactory
    - Highly unsatisfactory
3. Correctness of Sentimental classification of entities listed in the profile are
    - Highly satisfactory
    - Satisfactory
    - Neutral
    - Unsatisfactory
    - Highly unsatisfactory
4. Timeliness of information of the profile is
    - Highly satisfactory
    - Satisfactory
    - Neutral
    - Unsatisfactory
    - Highly unsatisfactory

B: Results

1. Correlation between entities, that I'm interested and listed in the profile are
2. Classification correctness of entities listed in the profile are
3. Correctness of Sentimental classification of entities listed in the profile are
4. Timeliness of information of the profile is

- A - Highly satisfactory,
- B - Satisfactory,
- C - Neutral,
- D - Unsatisfactory,
- E - Highly Unsatisfactory

	A	B	C	D	E	Total
Question 1	2	15	3	0	0	20
Question 2	1	8	7	4	0	20
Question 3	3	17	0	0	0	20
Question 4	3	17	0	0	0	20

Total results