Copyright
by
Guangyu Lin
2016

The Thesis Committee for Guangyu Lin
Certifies that this is the approved version of the following thesis:

**Trust Filter for Disease Surveillance:
Identity**

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Kathleen Suzanne Barber

Razieh Nokhbeh Zaeem

**Trust Filter for Disease Surveillance:
Identity**

by

Guangyu Lin, B.E.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2016

Dedication

To my loving parents, Jun Wan and Guangwen Lin, who are always by my side and support me.

Acknowledgements

I would like to thank my supervisor, Dr. Suzanne Barber, who patiently guided my research and has always been supportive of my endeavors during my stay here at The Center for Identity.

Thanks to Dr. Razieh Nokhbeh Zaeem, for consenting to be in the supervising committee, reviewing my work and providing valuable suggestions.

I would like to thank Andy Maloney, Ethan Baer, James Zaiss, Haowei Sun and other lab members at the Center for Identity who suggested valuable tools and feedback during my research.

I thank all the professors, teaching assistants, staff, and administration at the Department of Electrical and Computer Engineering for their supportive and innovative work.

Special thanks to my dear friend Yanbin Ren, Chen Xu who have always supported me through the graduation education.

Saving the best for the last, I would like to thank my mother and father for supporting me with love, and care to help me get through the challenging process of completing a graduate degree.

Abstract

Trust Filter for Disease Surveillance: Identity

Guangyu Lin, M.S.E.

The University of Texas at Austin, 2016

Supervisor: Kathleen Suzanne Barber

A flexible and extensible mobile application was delivered for evaluation and optimal inclusion of NextGen (Next Generation) data sources into biosurveillance for early detection, situational awareness and prediction. We present trust analysis of NextGen data sources to increase data confidence. One of the trust filters is the Identity filter, which helps us determine the degree of separation between the sender and the subject of a sentence. In this thesis, the author presents the definition of Identity. To help us distinguish different degrees of separation, the author uses relation distance along with a family tree to weight different relationships. Then the author compares a discriminative algorithm and a generative algorithm to calculate a user's Identity score. In the end, the author concludes that it is a good choice to apply a binary classification algorithm combined with a Natural Language Processing algorithm because of the trade-off between accuracy and runtime complexity.

Table of Contents

List of Tables	ix
List of Figures	x
List of Illustrations	xi
Chapter 1: Introduction	1
1.1 Contributions	3
Chapter 2: Related Works	5
2.1 Current National Programs	5
2.2 Information Retrieval from Social Media Data	5
2.3 Public Health Informatics from Web	6
2.4 Natural Language Processing	6
Chapter 3: Problem Description.....	8
3.1 Surety bioevent app.....	8
3.1.1 Node Table.....	10
3.1.2 Network Graph.....	11
3.1.3 Node Histogram	12
3.1.4 Trust Score Distribution.....	13
3.2 Trust Filter	15
3.3 Identity	15
Chapter 4: Problem Formulation	19
Chapter 5: Problem Solution.....	22
5.1 Natural Language Processing Algorithm.....	22
5.2 Multi-class classification algorithm	27
5.3 Binary Classification + Natural Language Processing Algorithm.....	28
Chapter 6: Experiments & Analysis	29
6.1 Experiment Setup.....	29
6.2 Experimental Results	30

Chapter 7: Discussion	33
Chapter 8: Conclusion.....	34
Bibliography	35

List of Tables

Table 1:	Variables Used in Problem Formulation.....	19
----------	--	----

List of Figures

Figure 1:	Surety BioEvent App	9
Figure 2:	Surety BioEvent App Scenario	9
Figure 3:	Node Table	10
Figure 4:	Node Table Ascending by Identity Value	11
Figure 5:	Network Graph	12
Figure 7:	Trust Score Distribution	14
Figure 8:	Initial Social Network	16
Figure 9:	Social Network with Identity Trust Filter	17
Figure 10:	Family Tree	17
Figure 11:	Weight-Family Tree	18
Figure 12:	Three Variants of Identity Trust Filter	24
Figure 13:	Natural Language Dependency Parser	26
Figure 14:	Name Entity Recognizer	26
Figure 15:	Accuracy	31
Figure 16:	Time	32
Figure 17:	Surety Bioevent App GUI	33

List of Illustrations

Algorithm 1:	Greedy Vote for Most Possible Identity	23
--------------	--	----

Chapter 1: Introduction

The Centers for Disease Control and Prevention (CDC) surveillance system collects and reports weekly data concerning influenza activity [1]. However, this process involves a two-week delay and having a faster way of detecting an outbreak would enable us to take action more quickly and thus better contain the outbreak and reduce the loss of lives. Thanks to the development of Internet and social media, people tend to express their opinions and feelings on the platform of social media, such as Twitter, Facebook and WordPress. For example, Twitter users often post tweets to express their personal feelings or information. Messages like "I have a headache" and "My daughter got a flu" are common. Though a tweet from one user does not interest us a lot, analyzing millions of flu related tweets could reveal public health information. Therefore, we can learn a lot from the social media data, which could include public health informatics, and we can do disease surveillance while also shortening the current two-week delay.

At the Center for Identity at the University of Texas at Austin, a flexible and extensible mobile application was delivered for evaluation and optimal inclusion of NextGen data sources into biosurveillance for early detection, situational awareness and prediction. One of the challenges for utilizing NextGen data sources is that they are noisy and of poor quality. Another challenge is that data collection and decision-making by analysts, who use NextGen data for biosurveillance, take too long. In this work, we present a synthetic method for determining the levels of trustworthiness of different NextGen data sources to increase the confidence interval of our data.

In order to determine the levels of trustworthiness, we introduce the concept of a trust filter — a score between 0 and 1 assigned to a data source, which rates its trustworthiness with respect to given criteria.

Here we define Identity as one of the trust filters, based on the relationship of a sentence's grammatical subject and its author. In order to weight different relationships, we calculate Identity by the gene distance utilizing a graph of family tree. The family tree includes six categories, which are self, nuclear family, extended family, other relatives, friends and others. The friends category is used to distinguish the relationship between important non-gene-based related people, such as (childless) spouses, close friends, co-workers, neighbors and acquaintances.

We design a workflow to help us generate the Identity trust filter network. Firstly, we collect data from social media platform by cloud sourcing flu key word set. Secondly, we use a binary classifier to reduce the noise in this data. In the end, we present a Natural Language Processing algorithm, a multi-class classification algorithm and a combined algorithm of binary classification and Natural Language Processing. We use the Natural Language Dependency parser and the Natural Language Name Entity Recognizer to help us understand a user's sentence [2], [3]. Princeton University's WordNet helps us to define a greedy algorithm to search words' cognitive synonyms [4].

In order to evaluate our algorithms, we run them on a randomly generated corpus and a real world Twitter corpus. From the comparison result, we present the trade-offs among these three algorithms.

1.1 CONTRIBUTIONS

We make the following contributions.

- We introduce the concept of an Identity filter and introduce the use of a family tree to define the Identity score.
- We present an overview of Surety BioEvent App and the screenshot each part of the application with explanations.
- We show a workflow to help generate the Identity trust filter network of a social media platform. It includes collecting data from a social media platform, reducing the noisy data and calculating the Identity trust filter network.
- We formulate an Identity Trust Filter Calculation (ITFC) problem. We design and implement a Natural Language Processing algorithm, a multi-class classification algorithm and a combined algorithm of binary classification and Natural Language Processing to help solve the ITFC problem and calculate a user's Identity score.
- We use a natural language dependency parser to retrieve a sentence's grammatical subject, greedily search the relationship between the grammatical subject and the sentence author and use Name Entity Recognizer to recognize person entity.
- We simulate our three algorithms on 1,000 randomly generated sentences and 10,000 real Twitter tweets and compare the accuracy and runtime complexity of them.

The rest of the thesis is organized as follow. We discuss the related work of current national programs, information retrieval from social media data, public health informatics

from web and Natural Language Processing in Chapter 2. We introduce the concept of trust filters and Identity in Chapter 3. Chapter 4 formulates the ITFC problem and Chapter 5 provides a solution. We present the experimental results in Chapter 6. We discuss the results in Chapter 7 and conclude the thesis in Chapter 8.

Chapter 2: Related Works

2.1 CURRENT NATIONAL PROGRAMS

The Centers for Disease Control and Prevention (CDC) has developed a 5-category and 8-component surveillance system that collects and reports weekly data concerning influenza activity. The goals of the system include (1) circulating reports of the most frequently identified influenza virus types, (2) detecting geographic spread of influenza, and (3) tracking influenza-like illnesses and their associated mortality rates [1], which provide an overall figure of influenza activity. However, it takes a long time to get all data from more than 120 laboratories, more than 2000 sentinel health care providers, and vital statistics officers in 122 cities [5]; hence there is about a two-week delay.

2.2 INFORMATION RETRIEVAL FROM SOCIAL MEDIA DATA

Researchers have recently focused on social media data for information retrieval, including for the purpose of disease surveillance. Barbosa and Feng measure public opinion by detecting sentiments on Twitter messages [6]. Zhang et al. rank products based on customer reviews from Amazon.com, which is related to sales ranking of retailers [7]. Joshi et al. prove there is correlation between opening weekend revenue and movie reviews [8]. Andranik and Timm predict elections using Twitter as a forum for political deliberation [9]; Eisenstein et al. combine latent topics and geographical regions to identify an author's geographic location and raw text using social media data [10]. Lerman et al. study the spread of news and Petrovic et al. detect new events from a stream of Twitter posts and show that people are most concerned about celebrity deaths [11], [12]. Sakaki presents earthquake detection by monitoring Twitter users with its real-time characteristic [13].

Social Media Data provide us a lot of information, which can help us learn people's current condition and feelings as soon as possible.

2.3 PUBLIC HEALTH INFORMATICS FROM WEB

Researchers have also shown the connection between public health informatics and Web. Jeremy, William and Michael show early detection or early warning systems of influenza activity are several weeks ahead of surveillance data collected by CDC [14]. Neil et al. evaluate public strategies and show how if public strategies respond rapidly, we can reduce the impact of seasonal and pandemic influenza [15]. Google Flu Trends detect US influenza-like illness (ILI) rates from millions of user search queries 7-10 days before CDC [16]. Big data technologies and analysis inspired them to devise a new method to achieve true real-time disease outbreak surveillance [17]. Signorini et al. examine the use of Twitter message to track public opinion towards Influenza A (H1N1) and measure influenza-like illness (ILI) [18]. Paul and Dredze introduce prior knowledge to analyze Twitter for public health informatics. Culotta achieves a correlation of 0.78 with CDC statistics by using 500,000 Twitter messages [20]. Therefore, we can treat people posts on the social media platform as next generation data, which could help us predict and control disease.

2.4 NATURAL LANGUAGE PROCESSING

Stanford Natural Language Dependency Parser provides a representation of grammatical relations between words in a sentence. It effectively extracts textual relations with an 84.2% F1 score [2], which is a measure of accuracy.

Named entities are important parts of sentences, but previous work did not make much use of the semantic information about named entities [21]. For example, it can be

very helpful to identify named entities in biomedical text. This work makes use of Stanford's Named Entity Recognizer (NER) [22], [23].

Chapter 3: Problem Description

As we discussed in Chapter II, Centers for Disease Control and Prevention (CDC) systems suffer from about a two-week delay to monitor disease activities. There is great need to come up with a strategy that would enable healthcare personnel to take actions as soon as possible when a disease outbreak happens, possibly saving lives. Thanks to recent technological developments in our society and the successful stories of big data analysis, we can extract a lot of valuable public health information from social media data. However, there is a lot of noisy data to be found in social media — data that is unrelated to our topic of interest. We need to improve quality of data, and track and quantify the trustworthiness of known, preferred sources. In this thesis, we consider a specific disease — influenza.

3.1 SURETY BIOEVENT APP

Figure 1 shows a high level overview of our current Surety BioEvent App. The application collects multi-source data from social media, such as Twitter, Facebook, WordPress, and others. Figure 2 shows a Surety BioEvent App Scenario. It consists of four parts, which are Node Table, Network Graph, Node Histogram, and Trust Score Distribution. Surety BioEvent App collects social media data from social media platform by its data collector. Then noise data is reduced and imported to trust filter diagram. Six dimensional trust filters will be applied to these data to help us improve the trust confidence. Then the trust filters' calculation result is presented as a Node Table, users' network graph is presented as a Network Graph, selected users' trust scores are shown in Node Histogram, and a network curve graph shows the distribution of the number of users over score intervals.

Figure 1: Surety BioEvent App

Figure 2: Surety BioEvent App Scenario

3.1.1 Node Table

Node table consists of node id, such as @Murilo_Oliiver and @lazarojrjunior, which are Twitter's user IDs; score of Identity, Reputation, Experience, Expertise, Authority and Proximity and the combined score of them. The average score is presented on the top bar. Additionally, you can also click star beside each user in orders to add it to your favorite. You can sort it by ascending or descending. Figure 4 is a screenshot of the sort by Identity Filter. The largest Identity value is 0.899 and the average of all Identity value is 0.352.

★ Node	Identity. ⚙	Reputation. ⚙	Experience. ⚙	Expertise. ⚙	Authority. ⚙	Proximity. ⚙	Combined. ⚙
★ Average	0.352	0.010	0.990	0.008	0.010	0.080	0.242
☆ @Murilo_Oliiver	0.408	0	0.993	0.001	0	0	0.234
☆ @lazarojrjunior	0.435	0.041	0.998	0.006	0	0	0.247
☆ @LeahCypess	0.671	0.021	0.998	0.005	0.041	0.579	0.386
☆ @ohmytargaryen	0.359	0	0.995	0.003	0	0	0.226
☆ @MayFireATX	0.436	0	0.998	0.01	0	0	0.241
☆ @hukuzawayukiti	0	0	0.992	0	0	0	0.165
☆ @ElliasBugg_	0.189	0	0.994	0.002	0	0	0.197
☆ @writingchain	0.324	0	1	0.008	0	0	0.222
☆ @ScottDuthiee	0.429	0.077	0.995	0.003	0	0	0.251
☆ @lal0campo	0.256	0	0.993	0	0	0	0.208
☆ @cisaramos	0.541	0	0.998	0.005	0	0	0.257
☆ @itzzahra	0.501	0	0.997	0.004	0	0	0.25
☆ @AmandaGuimar	0.271	0.053	0.993	0.001	0	1	0.386
☆ @chafouette	0	0	0.992	0	0	0	0.165
☆ @littlgreys	0.386	0	0.993	0.001	0	0	0.23
☆ @elizabethkarr	0.569	0	0.993	0.001	0	0	0.261

Figure 3: Node Table

★ Node	Identity. ⚙	Reputation. ⚙	Experience. ⚙	Expertise. ⚙	Authority. ⚙	Proximity. ⚙	Combined. ⚙
★ Average	0.352	0.010	0.990	0.008	0.010	0.080	0.242
Filter: 0.00 – 1.00 Sort: <input type="button" value="Ascending"/> <input checked="" type="button" value="Descending"/> <input type="button" value="None"/>							
☆ @NKOTB4EVER09	0.899	0	0.998	0.006	0	0	0.317
☆ @GarryLin123	0.8	0	0.998	0.006	0	0	0.301
☆ @ImCeCeJ	0.722	0	0.995	0.003	0	0	0.287
☆ @KOrionFray	0.701	0	0.999	0.006	0	0	0.284
☆ @LeahCypess	0.671	0.021	0.998	0.005	0.041	0.579	0.386
☆ @MaRcOsVibe_653	0.67	0	0.992	0	0	0	0.277
☆ @devyflot	0.66	0	0.998	0.006	0	0	0.277
☆ @SuppaSavCMoney	0.66	0	1	0.007	0	0	0.278
☆ @HeatsquadTae	0.605	0	0.995	0.003	0	0	0.267
☆ @sthncross3	0.6	0	0.987	0.02	0	0	0.268
☆ @HannahDickie	0.598	0	0.992	0.016	0	0	0.268
☆ @JasonPEdge	0.594	0.033	0.997	0.011	0.088	0.594	0.386
☆ @cameron_mac8	0.572	0	0.996	0.003	0	0	0.262
☆ @alishathkara	0.560	0	0.993	0.001	0	0	0.261

Figure 4: Node Table Ascending by Identity Value

3.1.2 Network Graph

Network Graph is a graph based on user's relationship. A line between two users means one of them follow another. For example, there is a line between @katyachamma and @hoyqueteV4s because @katyachamma follow @hoyqueteV4s. Additionally, when you select one trust filter, the size of each node is based on the value of this specific trust filter score.

Network Graph

Figure 5: Network Graph

3.1.3 Node Histogram

Figure 6 is an example of Node Histogram. It presents five users as an example, which is @NKOTB4EVER09, @GarryLin123, @ImCeCEj, @KorionFray, and @LeahCypess. We can clearly see the comparison between each trust filter.

Figure 6: Node Histogram

3.1.4 Trust Score Distribution

Figure 7 shows the trust score distribution. It represents score and number of users. For example, for score of 0.2, 13 users' Identity value are between 0.2 and 0.3, 1 users' Expertise value are between 0.2 and 0.3, 3 users' Authority value are between 0.2 and 0.3 and 64 users' Combined value are between 0.2 and 0.3.

Figure 7: Trust Score Distribution

Then, trust filters help us calculate trust scores to improve quality of data employed to make bio-surveillance decisions. The Surety BioEvent App introduces the six trust filters below. This thesis is focused on the Identity trust filter.

3.2 TRUST FILTER

In the Surety BioEvent App, we judge trustworthiness against six dimensions. We describe the six trust filters below.

1. Expertise: the probability a user will generate new content on the topic in question.
2. Experience: the degree to which a user's posts are corroborated.
3. Authority: number and quality of social media links received from Hubs as an Authority.
4. Identity: the degree of familial or social closeness between the user and the person afflicted with the disease. (The greatest value is when the user him/herself is the one afflicted.)
5. Reputation: number and quality of social media links to the posting user.
6. Proximity: the distance of a user from event (disease source), based on geographical tags.

3.3 IDENTITY

When trying to determine where infectious disease is, it is important to understand whether a user is talking about himself or someone else, such as his grandmother. If he is talking about his grandmother, and she lives with him, then any text that mentions the disease will suffice to influence him. If a user is talking about him, then it is important to recognize this when trying to find where the disease is in our surveillance algorithms. Applying the Identity algorithm - node color - we can identify those nodes that are the most important within the initial network shown in Figure 2. In Figure 3, the yellow nodes

indicate users that are talking about themselves, while the red nodes indicates users that are talking about others.

The main part of Identity algorithm uses the text that a user generates to determine if they are talking about themselves or not. We define Identity as the relationship between a sentence's author and its grammatical subject. Additionally, we find that a family tree with important non-gene-based relationships is a good way to reveal the relationship and generate a simple model of disease transmission between people according to the small-world phenomenon [24]. Figure 4 shows the relationship of each person to the original person, including cousins.

As a result, we can calculate the Identity value of each sentence by understanding each sentence and using the family tree, in order to improve confidence interval of our data.

Figure 8: Initial Social Network

Figure 9: Social Network with Identity Trust Filter

Figure 10: Family Tree

Figure 11: Weight-Family Tree

Chapter 4: Problem Formulation

For a social network, the first problem we face is how to calculate each node's Identity score. To simplify this problem, we define each user in the social network as a node and each post of a micro-blogging, such as Twitter or Facebook, as a sentence. According to the family tree in Figure 4, we can weigh each user by its genetic distance to the yellow user self in Figure 5, which ranges from 0.01 to 1. For example, we can collect all posts from a Twitter user, which is related to flu and label these posts with different relationship classes. To simplify this problem, we define it as follows.

Name	Description
S	a set of sentences from a social media noisy data source
P	a set of people in a social media graph
T	the set of relationships according to a family tree and non-gene-based relationships
L	the set of Identity relationship labels
s_i	a sentence in S
p_i	a person in P
l_i	the Identity relationship label of the sentence s_i
N_i	the grammatical subject of the sentence s_i
$D_{m,n}$	the word distance from a noun m to a noun n
$F(x)$	a function that maps each Identity relationship label l_i to an Identity score
T_x	a list of noun candidates associated with a member of T
V_i	the Identity score of p_i
$PERSON$	a person Named Entity detected by NER

Table 1: Variables Used in Problem Formulation

We have a set of sentences as $S = \{s_1, s_2, \dots, s_n\}$, a set of people as $P = \{p_1, p_2, \dots, p_m\}$, $T = T_{\text{Self}}, T_{\text{NuclearFamily}}, T_{\text{ExtendedFamily}}, T_{\text{OtherRelatives}}, T_{\text{Friends}}, T_{\text{OtherRelatives}}$, We assign a list of

noun candidates to each family tree relationship map T_x and map an Identity score to each set of T. Here, we present some of noun candidates in the different relationship sets.

T_{Self} : "me", "I", "myself", "own", "we"

$T_{\text{NuclearFamily}}$: "father", "mother", "son", "daughter", "brother", "sister", "dad", "mom", "kid", "child", "baby", "honey", "wife", "husband", "darling", "boy", "girl", "kiddo"

$T_{\text{ExtendedFamily}}$: "grandfather", "niece", "nephew", "grandson", "granddaughter", "uncle", "aunt", "grandmother"

$T_{\text{OtherRelatives}}$: "great-grandfather", "grand-niece", "grand-nephew", "cousin"

T_{Friends} : "teacher", "classmate", "friend", "dude", "buddy", "partner", "mate", "colleague", "pal", "they", "you", "he", "she" (The definition of third person is the grammatical person used by the speaker of an utterance in referring to anyone or anything other than the speaker or the one being addressed. Therefore, we treat the third person as Friends category.)

T_{Others} : other nouns such as "it", "river", "flu"

We present the Identity trust filter calculation (ITFC) problem that calculates $(\bigvee p_i)$ V_{p_i} , that is

$$p_j \text{ is the author of } s_i \quad V_{p_j} = \frac{\sum_{i=1}^n F(I_{s_i})}{n} \quad (1)$$

$$\text{Subject to} \quad (\bigvee p_i) V_{p_i}$$

To illustrate the ITFC problem, we present an example. A Twitter user named Garry tweets 5 posts related to the flu, which are {"I have the flu", "My daughter has the flu", "Flu is annoying", "Fear the flu? Some answers here", "Swine Flu vs. Bird Flu vs. Zombie Flu"}. For the first sentence, its grammatical subject is 'I' and it belongs to T_{Self} , so we can

calculate its Identity value as 1.0. Similarly, we can also calculate other 4 posts as 0.5, 0.01, 0.01, and 0.01. Then the Identity trust filter is the average calculated as $(1 + 0.5 + 0.01 + 0.01 + 0.01) / 5 = 0.506$.

Therefore, finding a best way to label the sentence according to the relationship between its grammatical subject and the sentence's author is key to calculating the user's Identity value. We will discuss this in the next section.

Chapter 5: Problem Solution

In this section, we provide three methods to solve the **ITFC** problem: (1) a generative algorithm, (2) a discriminative algorithm and (3) a combined algorithm. Figure 6 shows the entire solution process. Our first step is to cloud source user posts from Twitter by a flu-related key word set using our Surety BioEvent App. However, noise cannot be avoided by this method in the collected data set; for example, we might collect some sports brand-related sentences that happen to contain the key word "flu", such as the shoes, "Air Jordan 12 Flu Game". For our second step, we use a binary classifier here to retrieve flu related data and we can see the difference of their Latent Dirichlet allocation (LDA) topic graph in Figure 6 [25], [26]. We experiment with the three **ITFC** algorithms: **Natural Language Processing**, **Multi-class classification** and **Binary classification + Natural Language Processing**.

5.1 NATURAL LANGUAGE PROCESSING ALGORITHM

We consider each Twitter post as a sentence. In the Natural Language Processing variant of the Identity filter, we find the grammatical subject and identify the relationship between it and the sentence author. By using Natural Language Processing method, Algorithm 1 shows how to solve the **ITFC** problem in a greedy way. In Algorithm 1, lines 1 to 5 are initialization steps. We initialize the noun candidate list for the different relationships as discussed in Section 4, map function $F(x)$ as in Figure 5, and clean and tokenize each sentence by their stop words and punctuation.

The main loop (line 6 to line 24) calculates each sentence of a specific user, and lines 10 to 23 determine the best candidate label via a greedy word meaning network comparison.

input : Flu Related Micro blogs of a Twitter user
output: Identity value of this Twitter user

- 1 Initialize noun candidate list of different relationship T_x in T and $F(L_i)$ function to map label to Identity value
- 2 $S \leftarrow$ retrieve all sentences of this specific user;
- 3 Clean tokens of each sentence ;
- 4 $sum \leftarrow 0.0$;
- 5 $Identity \leftarrow 0.0$;
- 6 **for** $i = 1$ to $length(S)$ **do**
 - 7 $distance \leftarrow \infty$;
 - 8 $candidate \leftarrow \emptyset$;
 - 9 $N_i \leftarrow$ Retrieve grammatical subject of $S[i]$;
 - 10 **if** $\exists x N_i \in T_x$ **then**
 - 11 $candidate = T_x$;
 - 12 **else if** $\exists_{NER} PERSON \in N_i$ **then**
 - 13 $candidate = PERSON$;
 - 14 **else**
 - 15 **for** T_x in T **do**
 - 16 $D_{T_x, N_i} \leftarrow \infty$;
 - 17 **for** $\forall n_j \in T_x$ **do**
 - 18 $D_{n_j, N_i} \leftarrow \text{Wordnet.synsets}(n_j, N_i)$;
 - 19 **if** $D_{n_j, N_i} < D_{T_x, N_i}$ **then**
 - 20 $D_{T_x, N_i} = D_{n_j, N_i}$;
 - 21 **if** $D_{T_x, N_i} < distance$ **then**
 - 22 $distance = D_{T_x, N_i}$;
 - 23 $candidate = T_x$;
 - 24 $sum \leftarrow sum + F(candidate)$;
- 25 $Identity \leftarrow sum / length(s)$;
- 26 **Return** $Identity$

Algorithm 1: Greedy Vote for Most Possible Identity

Figure 12: Three Variants of Identity Trust Filter

We perform a greedy search to determine whether the candidate subject nouns are in our relationship set. If not, we check whether there is a named entity in our candidate subject nouns. If not, we perform a greedy calculation of word distance according to WordNet and vote for the shortest distance among all cognitive synonyms. Line 24 calculates the average Identity score of all posts from one user and returns this value.

This problem can be summarized as below.

$$L_{s_i} \leftarrow \min_{T_x \in T} |D_{T_x, N_{s_i}}| \quad (2)$$

$$p_j \text{ is the author of } s_i \quad V_{p_j} = \frac{\sum_{i=1}^n F(l_{s_i})}{n} \quad (3)$$

$$\text{Subject to} \quad (\forall p_i) V_{p_i}$$

From now on, we reduce this problem to finding the subject of the sentence. One common way of defining the subject of a sentence s in English is as the noun phrase that is the child of Subject (S) and the sibling of Verb Phrase (VP). One acceptable way to retrieve the subject from a sentence could be via the Natural Language Dependency Parser, which has a good F1 score 84.2 [2], [3].

To illustrate how the dependency parser works, consider the sentence "Garry began to cough". The sentence has a direct relation between the controlled verb *cough* and its controller, *Garry*, which is represented as a subject edge. Then we can learn that the grammatical subject of this sentence is Garry.

Figure 13: Natural Language Dependency Parser

Named Entity Recognizer (NER)

- In some scenarios, people mention their friend's or their relative's name instead of saying "my aunt" or "my friend". Identifying the named entity is very important for determining the Identity value of the sentence. We use the Stanford Natural Language Named Entity Recognizer (NER) to help us identify the named person entity because of its 80.95% F1 score. For example, if we use NER to perform relation extraction from the sentence "Garry began to cough", it identifies Garry as a person entity.

Garry began to cough
PERSON

Figure 14: Name Entity Recognizer

Additionally, we use WordNet to find synonyms of words and to estimate the semantic distance between different words. WordNet is a large lexical database of English, which is published by Princeton University [4]. In Table I, $D_{m,n}$ denotes the meaning metric from a noun to the other noun by WordNet. We can use nouns grouped into sets of cognitive synonyms to help us find the shortest metric from one word in a greedy way. Consider the Self set as an example, we can calculate the WordNet vector as $\{Synset('i.n.03'), Synset('iodine.n.01'), Synset('maine.n.01'), ('one.n.01')\}$, which could be used to calculate the meaning distance to the candidate grammatical subject. Note that, for pronouns, we should check through brute force, because WordNet does not include determiners, prepositions, pronouns, conjunctions or particles.

5.2 MULTI-CLASS CLASSIFICATION ALGORITHM

Another way to achieve the solution to ITFC is to apply machine learning algorithms. We train a dataset manually and test the prediction for relationship classification. Here we use bag of words model of the most frequent unigrams, bigrams and trigrams appearing in the Twitter posts. Before extracting the n-grams, we first apply a word stemmer and a word lemmatizes to reduce related forms of a word to a common base form to improve our bag of words. Additionally, we filter out common stop words from the sentences. For the learning algorithm, we try Naive Bayes, Multinomial Naive Bayes and Support Vector Machine. We set different rate of training set and testing set, then compare the results.

5.3 BINARY CLASSIFICATION + NATURAL LANGUAGE PROCESSING ALGORITHM

In addition to a generative algorithm and a discriminative algorithm, we also try a combined algorithm with binary classification and Natural Language Processing. Firstly, we use a binary classifier to separate the sentences into two sets: one containing the sentences whose subject is the author (self) and the other containing the sentences whose subject is someone else. Then we use the Natural Language Processing algorithm to extract information relations from the latter set. Finally, we combine the prediction results.

Chapter 6: Experiments & Analysis

6.1 EXPERIMENT SETUP

To evaluate the performance of our algorithms, we choose two kinds of text content. One is the random topic generated sentence with an online tool based on English sentence patterns and tenses. It includes five basic patterns and four tenses, which are

- **Subject + Verb**
- **Subject + Verb + Object**
- **Subject + Verb + Complement**
- **Subject + Verb + Indirect Object + Direct Object**
- **Subject + Verb + Object + Complement**
- **Present Continuous**
- **Present Simple**
- **Present Perfect**
- **Past Simple**

We got 100 randomly generated sentences of each pattern and 100 sentences of random patterns.

We compare our Natural Language Processing algorithms and machine learning classification algorithms. For the multi-class classifier, we set 80% of the sentences as the training set and 20% of the sentences as the testing set. We retrieve the features by a bag-of-words method and tokenized them to feed our Multi-class Naive Bayes algorithm. Additionally, we also set the other nine corpora as a training set and the target corpus as a testing set. For the binary classifier + Natural Language Processing algorithm, we first separated the sentences into two sets: those whose subject is the author and those whose

subject is not the author. Then we went on to use the Natural Language Processing algorithm to label the latter set. Finally, we combined the results and tested their accuracy.

The other kind of text content we use is real world Twitter data. We collect data by our flu-related key words set from Twitter. To help us understand the data we gather, we use LDA topic model to analyze the topic distribution of our data source. We use open-source labeled flu related data to train the binary classifier to reduce the amount of noisy data [27], then we reuse LDA topic model to analyze the topic distribution of our current data source in order to help us analyze the results. After that, we run and compare our three algorithms on this dataset. We collect 15,000 posts from Twitter and filter them down to 10,000 sentences that are related to flu. We manually labeled them. Then, we run and compare our three algorithms on this dataset. We implement our algorithms in Python 2.7.

6.2 EXPERIMENTAL RESULTS

Figure 7 shows the accuracy comparison among the three algorithms. The accuracy range of our Natural Language Processing algorithm is from 93% to 96%, and its 94.1% average accuracy makes it the most accurate algorithm of the three. The binary classifier plus Natural Language Processing algorithm also has a good accuracy performance, ranging from 88% to 95%. However, the accuracy of traditional Multi-class Naive Bayes algorithm ranges from 45% to 90%. The average on the large training dataset is 72.5% and the average on the small training dataset is 72.1%, which is much lower than our Natural Language Processing algorithm.

Figure 8 shows the time cost comparison between the three algorithms. Note the multi-class classifier is about 1,000 times faster than the other two algorithms. Analyzing

each sentence with our Natural Language Processing algorithm takes about 5 seconds. However, our Multi-class Naive Bayes classifier is thousand of times faster than the Natural Language Processing algorithm, which takes about 8 seconds to process 10,000 sentences. Additionally, our combined algorithm is 1 second faster than the pure Natural Language Processing algorithm to process one sentence

Figure 15: Accuracy

From these results, we find that our Natural Language Processing algorithm has a high accuracy and is very slow. The multi-class classifier is not as accurate as the Natural Language Processing algorithm, but it is very fast. This is also a well-known difference between the generative algorithm and the discriminative algorithm. It should be mentioned that the Natural Language Processing algorithm does not require us to label the data manually, because we use Stanford Natural Language Parser [2], [3], which has already

been trained on millions of sentences to adjust its accuracy. However, if we want to use the multi-class classifier, we need spend a lot of time to label our data manually; and it is not very feasible to train a large amount of data in a short time. The combined algorithms, however is a trade-off between these two algorithms. The Binary classifier + Natural Language Processing algorithm has good accuracy performance while being a little faster than the pure Natural Language Processing algorithm.

Figure 16: Time

Chapter 7: Discussion

Figure 9 is a screenshot of the Surety BioEvent App GUI. The node table shows each user's ID and his/her Identity value. The average Identity filter value of the 100 users in the table is 0.352, which shows that this network is potentially subject to a flu outbreak. The network graph shows relationships between various users, where the size of each node represents the user's Identity value. We find that user @NKOTB4EVER09 has a large Identity value, which indicates that there is a relatively high probability that he/she or a nuclear family member has the flu. By using the Identity trust filter, we can improve our confidence that our data can help us more quickly and accurately detect the presence of flu among groups of people.

Figure 17: Surety Bioevent App GUI

Chapter 8: Conclusion

In this thesis, we presented the definition of the Identity trust filter, based on the relationship between a sentence's grammatical subjects its author. We designed and implemented a workflow to retrieve, filter, and analyze data. We presented a Natural Language Processing algorithm, a multi-class classification algorithm, and a binary classification combined with Natural Language Processing algorithm. We implemented and simulated these three algorithms on both a randomly generated corpus of sentences and real Twitter data. Then we compared the accuracy and runtimes of the algorithms. We concluded that the discriminative Natural Language Processing algorithm is more accurate than the generative multi-class classification algorithm. But the latter algorithm is much faster than the former. Therefore, we used a combination of both algorithms to get an accurate Identity value in a timely manner. In our Surety BioEvent App, we present these three options, so that users can choose according to their particular requirements.

Bibliography

- [1] Centers for Disease Control and Prevention. Overview of influenza surveillance in the United States. Available at: <http://www.cdc.gov/flu/weekly/>. Accessed 8 October 2016.
- [2] Schuster Sebastian, and Christopher D. Manning. “Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks.” Proceedings of the 10th International Conference on Language Resources and Evaluation. 2016.
- [3] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. “Generating typed dependency parses from phrase structure parses.” Proceedings of LREC. Vol. 6. No. 2006. 2006.
- [4] George A. Miller (1995). “WordNet: A Lexical Database for English.” Communications of the ACM Vol. 38, No. 11: 39-41.
- [5] William W. Thompson, Lorraine Comanor, and David K. Shay. “Epidemiology of seasonal influenza: use of surveillance data and statistical models to estimate the burden of disease.” Journal of Infectious Diseases 194.Supplement 2 (2006): S82-S91.
- [6] Luciano Barbosa, and Junlan Feng. “Robust sentiment detection on twitter from biased and noisy data.” In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 36-44. Association for Computational Linguistics, 2010.
- [7] Kunpeng Zhang, Yu Cheng, Wei-keng Liao, and Alok Choudhary. “Mining millions of reviews: a technique to rank products based on importance of reviews.” In Proceedings of the 13th International Conference on Electronic Commerce, p. 12. ACM, 2011.
- [8] Mahesh Joshi, Dipanjan Das, Kevin Gimpel, and Noah A. Smith. “Movie reviews and revenues: An experiment in text regression.” In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 293-296. Association for Computational Linguistics, 2010.
- [9] Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, and Isabell M. Welpe. “Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment.” ICWSM 10 (2010): 178-185.

- [10] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. "A latent variable model for geographic lexical variation." In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 1277-1287. Association for Computational Linguistics, 2010.
- [11] Kristina Lerman, and Rumi Ghosh "Information contagion: An empirical study of the spread of news on Digg and Twitter social networks." ICWSM 10 (2010): 90-97.
- [12] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. "Streaming first story detection with application to twitter." In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 181-189. Association for Computational Linguistics, 2010.
- [13] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. "Earthquake shakes Twitter users: real-time event detection by social sensors." In Proceedings of the 19th international conference on World wide web, pp. 851-860. ACM, 2010.
- [14] Jeremy U. Espino, William R. Hogan, and Michael M. Wagner. "Telephone triage: a timely data source for surveillance of influenza-like diseases." AMIA. 2003.
- [15] Neil M. Ferguson, Derek A.T. Cummings, Simon Cauchemez, Christophe Fraser, Steven Riley, Aronrag Meeyai, Sapon Iamsirithaworn, and Donald S. Burke "Strategies for containing an emerging influenza pandemic in Southeast Asia." Nature 437, no. 7056 (2005): 209-214.
- [16] Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant. "Detecting influenza epidemics using search engine query data." Nature 457, no. 7232 (2009): 1012-1014.
- [17] Herman Anthony Carneiro, and Eleftherios Mylonakis. "Google trends: a web-based tool for real-time surveillance of disease outbreaks." Clinical infectious diseases 49.10 (2009): 1557-1564.
- [18] Alessio Signorini, Alberto Maria Segre, and Philip M. Polgreen. "The use of Twitter to track levels of disease activity and public concern in the US during the influenza A H1N1 pandemic." PloS one 6.5 (2011): e19467.
- [19] Michael J. Paul, and Mark Dredze. "You are what you Tweet: Analyzing Twitter for public health." ICWSM 20 (2011): 265-272.

- [20] Aron Culotta. “Towards detecting influenza epidemics by analyzing Twitter messages.” In Proceedings of the first workshop on social media analytics, pp. 115-122. ACM, 2010.
- [21] Alan Ritter, Sam Clark, and Oren Etzioni. “Named entity recognition in tweets: an experimental study.” In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1524-1534. Association for Computational Linguistics, 2011.
- [22] David Nadeau, and Satoshi Sekine. “A survey of named entity recognition and classification.” *Linguisticae Investigationes* 30.1 (2007): 3-26.
- [23] Shipra Dingare, Malvina Nissim, Jenny Finkel, Christopher Manning, and Claire Grover. “A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations.” *Comparative and Functional Genomics* 6, no. 1-2 (2005): 77-85.
- [24] Duncan J. Watts “Networks, Dynamics, and the Small-World Phenomenon 1.” *American Journal of sociology* 105.2 (1999): 493-527.
- [25] Carson Sievert, and Kenneth E. Shirley. “LDAvis: A method for visualizing and interpreting topics.” In Proceedings of the workshop on interactive language learning, visualization, and interfaces, pp. 63-70. 2014.
- [26] Ian. Fellows. “wordcloud: Word clouds.” R package version 2 (2012): 109.
- [27] Alex Lamb, Michael J. Paul, and Mark Dredze. (2013, June). Separating Fact from Fear: Tracking Flu Infe