Finding Provenance Data in Social Media

by

Geoffrey P. Barbier

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2011 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Herbert Bell
Baoxin Li
Arunabha Sen

ARIZONA STATE UNIVERSITY

December 2011

ABSTRACT

A statement appearing in social media provides a very significant challenge for determining the provenance of the statement. Provenance describes the origin, custody, and ownership of something. Most statements appearing in social media are not published with corresponding provenance data. However, the same characteristics that make the social media environment challenging, including the massive amounts of data available, large numbers of users, and a highly dynamic environment, provide unique and untapped opportunities for solving the provenance problem for social media. Current approaches for tracking provenance data do not scale for online social media and consequently there is a gap in provenance methodologies and technologies providing exciting research opportunities. The guiding vision *is the use of social media information itself to realize a useful amount of provenance data for information in social media*. This departs from traditional approaches for data provenance which rely on a central store of provenance information. The contemporary online social media environment is an enormous and constantly updated "central store" that can be mined for provenance information that is not readily made available to the average social media user. This research introduces an approach and builds a foundation aimed at realizing a provenance data capability for social media users that is not accessible today.

This effort is dedicated to my family.

Thank you for your inspiration, example, and support.

ACKNOWLEDGMENTS

The members of the Arizona State University (ASU), Data Mining and Machine Learning (DMML) laboratory have been a motivating influence and have offered thought-inspiring comments and questions with reference to this topic.

The Twitter database used for this effort was provided by Mohammed Ali Abbasi, ASU DMML laboratory colleague.

Figure 4.2 was produced with assistance from Gabriel Pui Cheong Fung.

This work was funded, in part, by OSD-T&E (Office of Secretary Defense-Test and Evaluation), DefenseWide/PE0601120D8Z National Defense Education Program (NDEP)/BA-1, Basic Research; SMART Program Office, www.asee.org/fellowships/smart, Grant Number N00244-09-1-0081.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF SYMBOLS/NOMENCLATURE

Symbol		Page
F_η	Set of friend names	45
$p(F_\eta)$	Probability of a match	45
l	Provenance data legitimacy	43
N	Number of individual statements	53

PREFACE

Finding provenance data in social media occupies an exciting and vast problem space. A challenge I faced for this effort was to formally define a specific problem to solve that is both a logical starting point for long term research and an appropriate scope for making a meaningful contribution.

Portions of this work were previously published:

- At the 2011 International Conference on Social Computing, Behavioral Modeling, and Prediction [9].
- In the book, Social Network Data Analytics [18].

The protocol used for this research effort is considered exempt by the Arizona State University, Office of Research Integrity and Assurance, Institutional Review Board (IRB). Reference Appendix A for a copy of the exemption letter dated, February 18, 2011, protocol number 1102006062.

Chapter 1

INTRODUCTION

The first microblog message, now commonly known as a *tweet*, was published in 2006 [63]. Since that time, these tweets[1] have been used by millions of people all over the world to publish statements about everything from the weather to presidential elections. Tweets can also be a great resource for emergency responders [33] and organizations providing Humanitarian Aid and Disaster Relief (HADR) [52]. For example, Figure 1.1 is a screen shot of the TweetTracker application developed by researchers at Arizona State University's Data Mining and Machine Learning Laboratory (DMML). TweetTracker is an application that can be used to assist first responders during Humanitarian Aid and Disaster Relief (HADR) operations. Research shows that tweets can have great potential to provide information faster and more accurately than some traditional sensor networks and communications paths [72]. However, with the popularity[2] and broad utility of this social media mechanism comes a challenge facing mainstream social media users today.

Amongst the factual statements published in social media, including tweets, are: opinions, rumors, hidden motivations, and deceptive content. Some noteworthy research has investigated how to distinguish between topics that are rumors and topics that are factual given a large number of number of tweets about a subject [58]. However, an individual recipient of a single statement made in social media, including a tweet, does not always have additional data *about* the particular statement that could provide important clues about where the statement came from,

[1] Messages published via the popular microblog service Twitter, http://twitter.com

[2] 140 million average tweets per day (http://blog.twitter.com/2011/03/numbers.html, accessed on October 19, 2011)

Figure 1.1: Example TweetTracker display

why it might have been published, and who (if anyone) might have modified the statement. This becomes important because collective behavior can be influenced by statements published in a social media setting such as a social networking site, a blog, microblog, or even a wiki [1, 29, 39, 74, 86].

A lack of accurate, reliable history or metadata about a social media information source can present problems as illustrated by a few case studies. In March 2010, John Roberts, a United States Supreme Court Justice, was reportedly planning to retire because of health issues. As it turned out, Justice Roberts had no plans to retire and a rumor that grew from a college professor's teaching point, meant only for a classroom example about the validity of informants, made national headlines [1, 13, 29, 69]. When Twitter was used by numerous protestors in Iran

during 2009, the source of some messages could not be verified and therefore were deemed to be of no value or even antagonistic [39]. A United States Department of Agriculture employee was forced to resign after a video clip posted on a blog was taken out of context resulting in an embarrassment for United States government administrators and a very challenging set of circumstances for the employee and her superiors [74]. Researchers at the Georgia Institute of Technology learned that trust in large groups can be complicated when they participated in a Defense Advanced Research Projects Agency (DARPA) experiment, and implemented social media as a communications mechanism, when members of competing teams "infiltrated" other teams [86]. These problems might have been avoided with provenance data related to the subject, the source, or perhaps even the ideologies in play.

Considering provenance as "the history of ownership of a valued object[3]", and the valued object as a statement in social media; provenance data in social media is the metadata associated with a statement including information about the origins, custody, and ownership of the statement published in a social media setting. Today, provenance data in social media is often only known after a group has been influenced and motivated in a particular manner. Having easier, or any, access to provenance data could prevent some undesired collective behaviors and motivate other collective behaviors based on facts instead of fiction.

Some mechanisms have been designed to record provenance data for databases, the semantic web, workflows, and distributed processing [59]. However, provenance data is not routinely tracked today for social media. Although some thought has been given about the need [42, 46] and some potential approaches [37, 46, 75], a

[3]http://www.merriam-webster.com/dictionary/provenance, accessed October 19, 2011

practical approach and responsive mechanism has not be identified or implemented for today's online social media environment. In some instances, sufficiently partial provenance data may suffice to inform groups in such a manner resulting in sound behaviors. Additionally, an approach for provenance data in social media needs to address the rapidly changing social media environment and should quickly respond to queries about the provenance of a piece of information published in social media.

The social media environment provides unique challenges for tracking and determining provenance data for statements found in social media. First, the social environment is *dynamic*. With more than half a billion[4] Facebook[5] users, new social media content is generated every day. Facebook is only one social media outlet. Another example is the popular microblogging site Twitter: there are over 140 million tweets posted every day. Today, users are leveraging social media as a routine communication mechanism and in some cases more than e-mail [30, 67]. Second, social media is *decentralized* in the sense that statements can be published by almost anyone choosing one or more social media platforms and then relayed across disparate platforms to a multitude of recipients. Third, the environment provides *multiple modes* of communication such as profile updates, blog posts, microblogs, instant messages, and videos. Given this extremely challenging environment, new approaches for managing provenance data are needed to track where a statement originated from and determine whether or not the statement can be used as a basis for a decision.

Obtaining the provenance data about statement is especially difficult because provenance data is not explicitly maintained by most social media appli-

[4]http://www.facebook.com/press/info.php?statistics, accessed on October 19, 2011
[5]www.facebook.com

cations today. However, the same characteristics that make the social media environment challenging provide unique and untapped opportunities for solving the provenance data problem for social media. Current approaches for tracking provenance information do not scale for social media. Consequently, there is a gap in provenance methodologies and technologies providing exciting research opportunities for computer scientists and sociologists. This work introduces a practical and theoretical approaches aimed guiding future efforts to realize a provenance data capability for social media that is not available today. The guiding vision is *the use of social media information itself to realize a useful amount provenance data for information in social media* [9].

This work presents novel research aimed at building a foundation from which to build upon to address the challenge of finding provenance data in social media. A brief chapter about social media is included followed by a chapter presenting research questions. A chapter introducing and discussing provenance paths is followed by a chapter focussed on provenance attributes including definitions and an approach for assessment. Following the chapter about provenance attributes, an investigation of provenance attributes through manual and automated means is presented with related discussion about results and implications. Finally, related works are highlighted in a separate chapter followed by general conclusions and recommendations for future research.

Chapter 2

SOCIAL MEDIA

Kaplan and Haenlein [50] define Social media as:

> "a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of User Generated Content."

Social Media[1] [2] also refers to a variety of information services used collaboratively by many people placed into the subcategories shown in Table 2.1.

Table 2.1: Common Social Media Subcategories

Category	**Examples**
Blogs	Blogger, LiveJournal, WordPress
Microblogs	Twitter, GoogleBuzz
Opinion mining	Epinions, Yelp
Photo and video sharing	Flickr, YouTube, Pinterest
Social bookmarking	Delicious, StumbleUpon
Social networking sites	Facebook, LinkedIn, Google+, MySpace, Orkut
Social news	Digg, Slashdot
Wikis	Scholarpedia, Wikihow, Wikipedia, Event maps

Social media is associated with social computing. Social computing is *"any type of computing application in which software serves as an intermediary or a focus for a social relation"* [73]. Social computing includes applications used for

[1] Some researchers distinguish between social media and social networks [51].

[2] Social media can also be classified based on social presence/media richness and self-presentation/self-disclosure into six categories: collaborative projects, blogs, social networking sites, content communities, virtual social worlds, and virtual game worlds [50].

interpersonal communication [73] as well as applications and research activities related to “computational social studies [89]” or “social behavior [21]”.

With traditional media such as newspaper, radio, and television, communication is almost entirely one-way, originating from the media source or advertiser to the masses of media consumers. Web 2.0 technologies and contemporary online social media changed the scene moving from one-way communication driven by media providers to where now almost anyone can publish written, audio, or video content to the masses. This many-to-many media environment is significantly changing the way business communicate with their customers [49, 87] and provides drastically unprecedented opportunities for individuals to communicate with extremely large numbers of people at an extremely low cost. The many-to-many relationships present online and manifest through social media are digitized data sets of social networks on a scale never seen before. The resulting data provides rich opportunities for sociology [19, 20, 53, 54, 82, 83, 85, 84, 92] and new insights to consumer behavior and marketing [10, 80, 89] amongst a host of related applications to similar fields.

The rise and popularity of social media is astounding. For example, consider the popular social networking site Facebook. In July 2010 Facebook users numbered over half a billion[3] and during the first eight years of operation Facebook reached over 750 million active users. Figure 2.1[4] illustrates the exponential growth of Facebook. Facebook is ranked 2nd in the world for internet sites based on the amount of daily internet traffic to the site.[5]

[3] http://www.facebook.com/press/info.php?timeline, accessed on October 19, 2011

[4] Figure produced with data found at http://www.facebook.com/press/info.php?timeline, accessed on October 19, 2011.

[5] Ranked according to http://www.alexa.com/topsites, accessed on October 19, 2011.

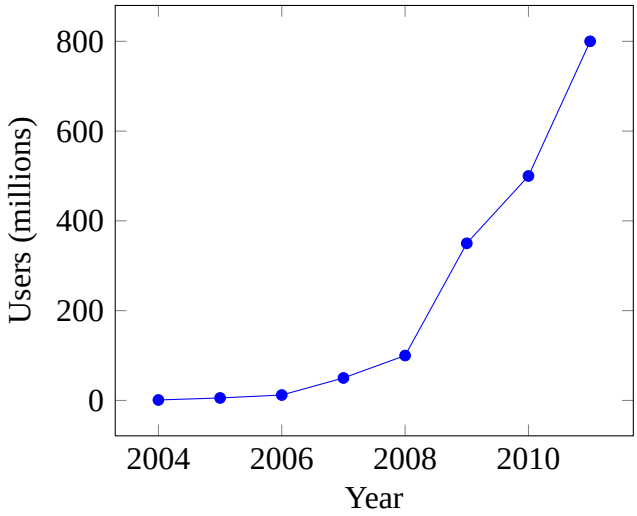

Figure 2.1: Number of Facebook Users Per Year

Social media, including social networking, blogs, and microblogs, continue to grow in popularity and are transforming the way people communicate. BlogPulse[6] automatically tracks and analyzes over 170 million blogs. The popular microblog service, Twitter, handles over 200 million 140-character messages per day [64]. Mobile devices are increasing the amount and frequency of information published in the social media environment. For example, 350 milion Facebook users are accessing the social networking service using their mobile devices[7].

The widespread use of social media is not limited to one geographic region of the world. Orkut, a popular social networking site operated by Google[8], has a majority of users from outside the United States[9]. The use of social media among internet users is now mainstream in many parts of the world including countries

[6]http://www.blogpulse.com/
[7]http://www.facebook.com/statistics#!/press/info.php?statistics, accessed on October 19, 2011
[8]http://www.google.com/
[9]http://www.orkut.com/MembersAll, accessed on October 19, 2011.

in Europe, Asia, Africa, South America, and the Middle East[10]; even well known organizations such as the United Nations are highlighting social media as a useful tool[11]. Social media is also driving significant changes in business and companies have to decide on their strategies for keeping pace with this new media [49].

2.1 Provenance Data in Social Media

With information published from so many sources, often republished and modified, it can be difficult for a recipient to know where a piece of information originated from, whether or not it should be trusted, or what latent purposes or biases might be attributed to the piece of information. Provenance metadata about pieces of information published in social media are not readily made available to users today. This can be problematic for recipient social media users who are unable to make accurate judgements about the information they receive.

Social media is rich with data linking individuals and can include a wealth of user profile data with a variety attributes[12]. Profile data can vary from very accurate and detailed information about a user to completely false information about a user, or even an altogether fabricated user. In addition to link and profile data, users make statements, join groups, share photos, post videos, and "vote" on issues.

Complete profile data and link data do not always accompany statements that are published in social media. In some cases, a more comprehensive profile could be aggregated by collecting data from the partial profiles that a single user has, in practice, spread across multiple social media service providers. However,

[10] http://www.alexa.com/topsites/countries, accessed on October 19, 2011.
[11] http://www.un.org/News/Press/docs/2011/sgsm13594.doc.htm, accessed on October 19, 2011.
[12] The variety of attributes available are dependent on individual user preferences

collecting user profile data for a single user from disparate social media sites is not effectively done today. Until a supporting infrastructure like the semantic web is widely embraced and utilized, social media users are left without a strategy and a means to meaningfully comprehend this data and realize benefits from latent provenance data present in the popular contemporary social media environment. The amount of data available in social media today is unprecedented and vastly differs from traditional media sources.

2.2 Provenance Data in Traditional Media

Statements published by traditional media methods, such as print or television, do not pose a significant challenge for determining the provenance of the information when compared to social media because of three important factors: First, *directives* (including self regulation) create a set of ethics that promote provenance data as an important aspect of the information that is provided. For example, the associated press *Statement of News Values and Principles*[13] includes the following:

> "It means we always strive to identify all the sources of our information, shielding them with anonymity only when they insist upon it and when they provide vital information - not opinion or speculation; when there is no other way to obtain that information; and when we know the source is knowledgeable and reliable."

Another example from the Canadian Broadcast Standards Council Code of Ethics[14]:

[13] http://www.ap.org/newsvalues/index.html, accessed on October 19, 2011.
[14] http://www.cbsc.ca/english/codes/cabethics.php#Clause5, accessed on October 19, 2011.

> "It shall be the responsibility of broadcasters to ensure that news shall be represented with accuracy and without bias. Broadcasters shall satisfy themselves that the arrangements made for obtaining news ensure this result. They shall also ensure that news broadcasts are not editorial."

The directives and ethical standards promote the disclosure of provenance data with the statements made in traditional media. However, social media users are not bound by regulations or formalized ethics.

Second, *production controls*. Traditional media editors and producers reinforce regulation and ethics providing a "checks and balances" service that is not present in social media. Another production control is the access to media outlets. With traditional media, access to media outlets is one-to-many. Both technology limitations and high cost of traditional media limited the number people and organizations that could publish statements. Today's social media user can publish at will, leveraging "many-to-many" communications technology that is extremely cheap in comparison to traditional media [87]. Time to publication is also a production control for traditional media methods. Television and newspaper content can be approved and delivered in minutes or hours. In the social media environment, where statements are both unregulated and easy to publish, statements can be communicated through social media almost instantaneously.

Third, *size*. The amount of social media content dwarfs the content produced by tradition media. Television networks ABC, NBC, and CBS, over the course of 60 years, produced 1.5 million hours of programming. Contrast that

amount with YouTube[15], a popular social media site. YouTube received more video in six months than all three of networks produced in total during the 60 years [92]. According to the Newspaper Association of America there were 1,387 newspapers in the United States and Canada in 2009[16]. Compare that number of newspapers with over 170 million blogs[17]. Thus, the overwhelming amount of social media complicates the process of obtaining provenance data when compared to traditional media methods.

Without binding values of integrity and formal production controls, social media users can publish freely to a massive population. Statements that would have sources identified in traditional media may not have the correct sources identified in social media.

Opinions published in social media are not limited to an editorial section. Facts may not be thoroughly checked with as much rigor as a traditional media organizations. In the end, the individual social media user is often left to judge whether a statement is fact, opinion, or rumor.

[15] http://www.youtube.com/

[16] http://www.naa.org/Trends-and-Numbers/Circulation/Newspaper-Circulation-Volume.aspx, accessed on October 19, 2011.

[17] http://blogpulse.com/

Chapter 3

FACT, OPINION, OR RUMOR?

When a user receives a statement via social media, the user must make an assessment about whether the statement is a fact, an opinion, or a rumor. Even a true statement, or mutually agreeable opinion statement, may have a hidden motivation.

Rumors, or deceptive statements, can result in a range of consequences varying from an embarrassment to causing real trouble. In May 2011, a fake quotation erroneously attributed to Martin Luther King made its way to thousands of social media users as it was a resent from user to user. The source of the erroneous quote was a Facebook post that included quotes from Martin Luther King but when the message was repeated inaccurately, it was quoted incorrectly [57].

Crosby lists several examples of how "bad information can be dispensed so easily and widely" through social media [26]. Her examples include false reports about a school shooter, rumors of anthrax in packages, and inaccurate reports about neighborhood crime.

The negative impact that rumors can have on society has been studied for years. Allport and Postman provide a "Basic Law of Rumor" in their book "The Psychology of Rumor" published in 1947 [5]. Their basic law of rumor is represented by:

$$\text{“}R \sim i \text{ x } a\text{”}$$

Where the strength of a rumor, R, depends on the importance, i, and the ambiguity, a, of the statement. In other words, whether or not a rumor will be circulated

depends on how important the subject of the rumor is to the recipient as well as how ambiguous the statement is. Allport and Postman argue when a statement is unimportant or is not ambiguous, there will not be a rumor. They also report that rumors usually are propagated among like-minded people [5].

There are important differences in 2011 compared to Allport and Postmans'1947 that enable rumors to spread more rapidly and widely than ever before throughout society. First, social media technology provides an infrastructure not only for communication but also an ideal infrastructure for rumor propagation because like-minded people are already organized in social networks. Second, people are able to communicate with thousands of other people instantaneously through social media - something that was not possible in 1947.

Given today's social media infrastructure, when a piece of information is important to Allport and Postmans' "like-minded people," it can be transmitted within seconds. The ability to rapidly resend messages through the like-minded structure and network of friends often masks the ambiguity because of the trust between social media users. A logical question to ask is how to help an individual user judge whether or not a statement appearing in social media is fact or fiction? One answer is to provide the user provenance data about a statement to help the user determine what level of confidence to put in statement.

In some cases, the wisdom of the social media crowd detects false information, or rumors, and the social media crowd performs a type of auto correction. When a false statement is widely propagated, researchers have observed that it is not repeated as often as true statements and in some cases are refuted altogether by taking advantage of the social network infrastructure already in place [58].

However, false information, or rumors, are not always widely disseminated throughout social media and are not always detected until some damage has been done. When the false statement is not popular or widely disseminated, end users would benefit from provenance data *about* the source and history of the statement in order to make a sound judgment concerning the statement.

3.1 Aspects of the Provenance Data Problem

When a popular statement is made, the real provenance data of interest is metadata affiliated with the source of the statement. Since a message is repeated by so many social media users, finding the provenance data about the original source becomes the primary goal.

In cases where there are multiple sources of the message, or there are messages that are similar, the search is focussed on the message that was sent first or most likely sent first. Provenance data about the earliest message will be the most valuable to the user.

In other cases, when a social media user receives a message that is not as popular and consequently not as widespread, it is useful to consider the provenance data about the source and any other nodes that may have retransmitted the message prior to the final user's receipt of the message.

Discovering provenance data in social media helps to solve the problem of reducing uncertainty about the origins, custody, and ownership of a statement published in a social media setting. Finding metadata about the origins and custody of a statement are at the heart of the provenance data problem. Simply put, origins are characterized as the metadata about a social media user that transmits or passes

Aspect	Problem to solve
Origin	What is the original source of the statement and what is known about the source?
Custody	What was the communications path of the statement and who may have modified the statement? What is know about anyone who may have modified or retransmitted the statement?
Ownership	If the statement is about someone, how are they associated with the communications path?

Table 3.1: Three aspects of the Provenance Data problem in social media.

along a statement. Such metadata are called *provenance attributes* and will be formally defined later in this work.

A social media user might be the original source of the statement or simply one who repeats or modifies a statement made in social media. A chain of users defines the custody of a statement such as a message that has been passed along nodes in a social network. The custody information about the statement will be known as a *provenance path* and will also be formally defined later in this work.

In some cases, ownership data is also an important aspect of provenance. Ownership in the context of a social media statement refers to a subject, specifically a human subject. The owner is the individual that is the subject of the statement (when such an individual exists). This becomes important when the subject is not the original source of the message or is not included in the provenance path. Table 3.1 lists the three aspects of the problem that are the driving factors to consider in order to fully address the problem of finding provenance data in social media.

For example, consider one of the rumors that was investigated by Mendoza et al. [58]. Amongst the thousands of tweets in the 2010 earthquake in Chile, some tweets were reported on the death of a famous singer, Ricardo Arjona. However,

Ricardo Arjona did not perish in the earthquake. This is a great example to examine from the perspectives of origin, custody and ownership.

Mendoza et al. reported finding several unique tweets about Ricardo Arjona and some of those tweets were retweeted, thus propagating the rumor. In this case, finding provenance data about the original source of the message will be most helpful. How the messages were propagated and modified would also be telling to a recipient, and before the message was widely propagated, provenance data about the chain of custody, or provenance path can also be helpful. It is also useful to consider the ownership of the statement (i.e., the subject of the statement). Some statements in social media will not have an owner. However, in the case of Ricardo Arjona, because he is the subject of the statement, he is owner of the information. In other words, Ricardo Arjona himself ultimately specifies whether or not he is dead. If Ricardo Arjona is not the source of the statement (or someone who is closely associated with him), that fact is useful provenance data. The same can be said of the nodes in a provenance path, if the owner of the statement is not part of the path, and then the veracity of the statement might be questioned.

3.2 Hypothesis and Contributions

Given the widespread use of social media in its variety of forms, and the propensity of such large numbers of people to use that media to communicate a statement that is valid, mistaken, or blatantly false, the problem becomes how to find provenance data that would prove useful to recipients. The hypothesis of this work is that **it is possible to use social media itself, as it exists in its present form, to obtain useful provenance data by leveraging the massive amounts of data published**

in social media to provide meaningful context about statements published in social media.

There are three provenance questions which seem to encompass the logical starting points for building a provenance data capability for social media:

1. When a user receives conflicting statements, which one, if any, should be accepted as credible?
2. When the owner of a statement is not the source, should the statement be accepted as credible?
3. When the source of the statement is not evident, what is the source of the statement?
4. When the source of the statement is not evident, should the statement be accepted as credible?

In order to demonstrate that it is possible to use social media as a source of provenance data for statements made in social media, basic research needs to be done to:

- Define a general framework for the problem. A theoretical contribution of this research effort is a general framework, the *provenance path*, for today's most popular, contemporary, social media environment. This framework is influenced by provenance work applied to other computational and information processing domains. This framework is the first contribution of this research, and is addressed in detail by the chapter on provenance paths in social media.

- Define what meaningful provenance data is for the social media environment. A significant challenge is identifying a method that will be applicable to all social media users in today's social media environment. The second contribution of this research effort is the definition of provenance data, provenance attributes, for today's social media environment. The chapter on *provenance attributes* provides a formal definition for provenance data in the social media context. The initial approach of working with provenance attributes is addressed in a subsequent chapter.

- Develop a criterion for evaluating the effectiveness of obtaining provenance data from social media. A third contribution of this research is a set of *metrics* that can be applied for evaluating efforts to find provenance data in social media.

- Explore the framework and mechanisms for obtaining meaningful provenance data. A fourth contribution of this research effort is to obtain experimental results that demonstrate the framework's potential and explore both the value and limitations of the framework and the approach. This also resulted in a proof-of-concept application for automatically finding provenance data in social media.

- Identify long term research challenges. A fifth contribution of this research is to identify additional *research opportunities* related to finding provenance data in social media.

3.3 Beginning with Twitter

The microblog site Twitter[1] will serve as the testing ground for this research effort to explore provenance data in social media. Why Twitter? Twitter has the basic characteristics of other social media sites including user profiles, a communication mechanism, a social network framework, and large number of users. Twitter does not provide provenance data about statements that are transmitted across its social network. Twitter messages, or tweets, are effectively public broadcast giving researchers easy access to data. Twitter data provides basic elements required to investigate the utility of the provenance path framework. Twitter data provides a simple environment for exploring provenance data and developing approaches to measure provenance data in a social media setting.

Not only does Twitter provide a simple setting for researching provenance data in social media, but Twitter provides meaningful utility to millions of people around the world every day including:

- Passing information about current events [17].
- Expressing feelings [96].
- Monitoring humanitarian aid and disaster relief needs and activities [52].
- Political messaging [94].
- Political advertising [78].

[1] http://twitter.com

- Commercial advertising [61, 93].
- Stock market correlation [40].

Thus, finding provenance data for statements appearing in tweets can be meaningful for Twitter users. For example, consider a tweet sent during a political campaign. Knowing more about the message, such as the political motivations of the originator, can provide a recipient with additional insights into the impetus behind a message.

Chapter 4

A PROVENANCE PATH FRAMEWORK FOR SOCIAL MEDIA

The social media environment network can be represented by a directed graph $G = (V,E)$, $v \in V$ and $e \in E$. Where V is the set of nodes representing social media users publishing information using social media applications. E is the set of edges in G representing explicit transmission of social media communication between two nodes in V. An explicit transmission occurs when distinct information is communicated from one node to another or when one node directly accesses information available at another node. Publishing information alone is not considered an explicit transmission and does not create an edge in E.

Provenance can be characterized as a directed graph [28, 37, 59, 77]. Within the graph, a *provenance path* can be assembled for each statement produced from the social media environment. The provenance path builds a general theoretical framework for finding provenance data in social media. Given the directed graph $G = (V,E)$. The following terms are defined:

Definition: T is the set of *recipient* nodes in $G : T \subseteq V$.

Definition: A is the set of *accepted*[1] nodes in $G : A \subseteq V$ and $(T \subset A)$.

Definition: D is the set of *discarded*[2] nodes in $G : D \subset V, (D \cap A) = \emptyset$,

and $(D \cap T) = \emptyset$.

Definition: $(A \cup D)$ are *identified* nodes.

Definition: M is the set of *undecided* nodes in $G : M = V - (A \cup D)$.

[1]The criterion for accepting nodes is uniquely determined by T.

[2]The criterion for discarding nodes is uniquely determined by T.

Definition: a *provenance path, p* , is a path in $G : p = (v_1, v_2, \ldots, v_n) : v_1 \neq v_n, v_1 \in V$, and $v_n \in T$.

Definition: P is the set of *all provenance paths* in $G : \forall p_i \in P, i = 1 \ldots m : m = \mid P \mid$ and $p_1 \neq p_2 \neq p_3 \neq \ldots p_m$.

Definition: *Accepted provenance path, p* : for all nodes, v_k, in path p, $v_k \in A$.

Definition: *Heterogeneous provenance path, p*: for all nodes, v_j, in path p, $v_j \in A, v_j \in D$, or $v_j \in M$.

A provenance path is a set of nodes and edges comprising a path on which an element of social media information is communicated from a node in the graph to one or more a recipient nodes. Nodes in the set T (an individual or group) are the final recipients of information along a provenance path, hereafter referred to as the recipient. The recipient makes decisions based on the information transmitted via a provenance path. Each provenance path is unique, and there may be more than one provenance path providing information to a recipient. Figure 4.1 illustrates the most common relationship between the subsets of V. The arrows illustrate some characteristics of possible provenance paths including accepted and heterogeneous provenance paths.

The ability to assess a provenance path, or to confidently consider a set of provenance paths, is a key to providing usable provenance data to a recipient. However, the social media environment provides a very challenging problem for finding provenance data. The social media environment, like the world-wide-web, provides a theoretically bounded but practically unbounded problem space because of the large number of users in the social media environment. Consider that there are a finite number of websites as part of the world-wide-web. However, determin-

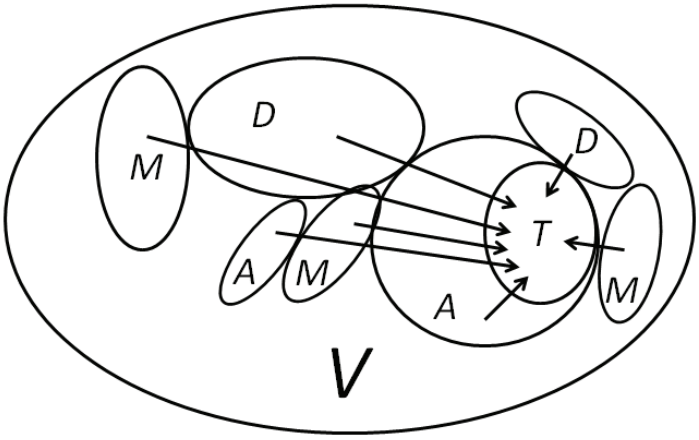

Figure 4.1: Sets and abstract paths

ing the actual number of web sites is extremely challenging [12]. Similarly, there are a finite number of social media users and a finite amount social media information. However, determining the precise number of users is practically intractable. This unbounded social media environment presents an unbounded problem space for provenance paths in practice[3].

A provenance path can begin at an identified node ($v_1 \in A$ or $v_1 \in D$) or from a node that is undecided ($v_1 \in M$). The social media environment also presents cases where a provenance path exists but all of the nodes and edges in the path are not known or only partially known to the recipient, defined as an *incomplete provenance path*. In the case of an incomplete provenance path, the complete provenance path exists in the social media environment but the complete path is not discernable to the recipient. Given an incomplete provenance path, the primary goal of solving the provenance path problem is to make all of the unknown nodes and edges known to the recipient. When all of the nodes and edges are known by the recipient, the

[3]In some cases the social media environment will be bounded such as when considering a single social networking site or small subset of social media sites.

provenance path is defined as a *complete provenance path*. When the social media environment is unbounded, it may not be possible to make a complete provenance path known to the recipient. The recipient will need to employ strategies, introduced later in this chapter, to decide whether or not the incomplete provenance data provides useful information.

Whether or not the provenance path is useful depends on how the recipient defines usable provenance information. A mechanism is needed to determine how multiple provenance paths providing the same or conflicting information should be evaluated by a recipient. The problem space can be considered and approached from different perspectives depending on whether or not the path is complete or incomplete.

4.1 Complete Provenance Paths

Assessing provenance will be easiest when the recipient can access a *complete provenance path* with node and edge relationships known to the recipient. Identified nodes are categorized, based on a criteria a recipient defines, as accepted or discarded. The criteria for accepting nodes can be based on one characteristic or a combination of characteristics attributed to nodes in the environment. Nodes in the graph that are representative of social media usually correspond to a person or a group with profile data associated with each node describing the person or group. Accepted nodes can be practically defined in many ways. Acceptance might mean trust through a friend-of-a-friend as described in [37]. Acceptance could also be defined by group affiliation, political affiliation, reliability ratings, by publicly posted comments in a social media setting, education level, etc. The provenance data

availability function formally defined in the next chapter, $r(V_\alpha)$, with a recipient-determined threshold value, could also be used to decide whether or not to accept a node with a mapping to α . Discarded nodes could be defined by the antithesis of the acceptance criteria, or more broadly as nodes which are not accepted. When all the nodes can be identified, the provenance path can be traversed, and if a discarded node is encountered, the information that was made available to a recipient individual or group can be discarded altogether or evaluated with additional scrutiny.

Given a complete heterogeneous provenance path, the first order of business is to identify any undecided nodes in question that are included in the provenance path. Perhaps one of the most exciting opportunities for coping with undecided nodes in a provenance path is leveraging social media itself to determine how to classify undecided nodes. Analyzing the content of a node can be used as a basis to identify an undecided node. For example, if the undecided node represents a group, recommendation systems might be leveraged to advise a recipient about whether or not the node is associated with a group that they might align with. Thus, the recipient would have some basis on whether or not to accept or discard the node. It is not a far stretch to see how a knowledge of social media groups [10] could be extended to nodes that represent individual people. An automated system could assess which groups the recipient aligns with based on the recipient's social media profiles and determine whether or not the undecided node representing an individual person would recommend the individual for the same groups as the recipient. Recommender systems are being implemented for various purposes uding a variety of technical approaches [2] including social media sites such as Facebook [7].

In this manner, an automated system could recommend nodes representing individuals as accepted or discarded.

Once all the nodes in the provenance path are identified, the provenance path can help a recipient assess the information that is transmitted along the path. Complete provenance paths can contain both accepted and discarded nodes. Recipients must decide whether or not a provenance path containing both accepted and discarded nodes negatively impacts the credibility of the statement communicated along the path. Similarly, when none of the nodes are identified, all nodes in the provenance path should be identified before the information is considered.

4.2 Incomplete Provenance Paths

If the actual path is not completely known to the recipient, it could be difficult to determine whether or not a discarded node contributed to or altered information presented to the recipient. In such cases, the challenge becomes to identify the complete the provenance path and it is likely that in some instances it will be impractical to identify the complete provenance path. When a significant portion of the path cannot be disclosed, an approximation or estimation of the provenance path could provide useful insights to the recipient event without the complete path identified. For example, if some nodes along the known portion of the provenance path are discarded. In particular, if the discarded nodes appear at the "beginning" of the path, the recipient might not view the statement as credible.

Social media provides opportunities to indirectly determine the actual or likely provenance path. Given a bounded social media environment (e.g., a single social networking site or small subset of social media sites), it may be possible for

a recipient to complete the provenance path by leveraging the social media data available. For example, link information from different social media sites associated with the same person might be leveraged to look for overlaps. Continuing a search on another social media site, based on the "beginning" of the path that is known may reveal other nodes and edges along the path.

A related challenge is when the incomplete provenance path is presented to a recipient in an unbounded social media environment. With hundreds of millions of social media users, it is conceivable that the complete provenance path will not be disclosed in a time frame that is usable to a recipient. It may be possible to use social media data to uncover only a portion of the provenance path. If the provenance path cannot be discovered in total, then the decision must be made about whether or not an incomplete provenance path is adequate to serve as a basis for a decision. In some cases, the content of the information may be inconsequential to the recipient and no decision will need to be made. In other cases, the recipient will need to employ probabilistic mechanisms to determine how the information should be considered. Depending on the circumstances, determinations could be made by directly finding the path in the social media environment or by obtaining information about the nodes and links in the social media network indirectly (separate from nodes and edges included in the actual provenance path).

Approaches need to be developed to create, search for, or estimate the provenance path when the provenance path is incomplete. Decision strategies need to be developed to help the recipient judge the credibility of information provided through social media or determine whether or not the information itself can be corroborated via a separate provenance path, including accepted social media nodes.

In an unbounded social media environment, it may be impossible to determine exactly who published something or who is responsible for a particular statement. However, in some cases it may be enough to know whether or not the idea being presented is adversarial, complementary, or unique, and how it might impact the recipient individual or group. This would require provenance data that is described in Chapter 5. Understanding the nuances of a publication, position, or opinion, could lend itself to a level of confidence acceptable to a recipient in order to assess information received from an incomplete provenance path characterized using only the portion of the provenance path that is available for analysis.

4.3 Multiple Provenance Paths

Multiple provenance paths present both prospects and challenges. Figure 4.2 illustrates the concept of multiple provenance paths and some of the challenges multiple paths present. When multiple provenance paths are *complementary*, the paths present consistent information to the recipient individual or group. Complementary provenance paths might help to serve as an authentication mechanism for the information presented to the recipient. However, caution is warranted because false or deceptive content can also be repeated to a recipient. The purpose of providing provenance data to a recipient is to help the recipient judge the credibility of the duplicate statements. The most challenging decisions an individual or group may need to make are when the provenance paths are incomplete and multiple provenance paths provide conflicting information.

When multiple provenance paths are *conflicting* by presenting inconsistent or contradictory information to the recipient, the provenance paths must be recon-

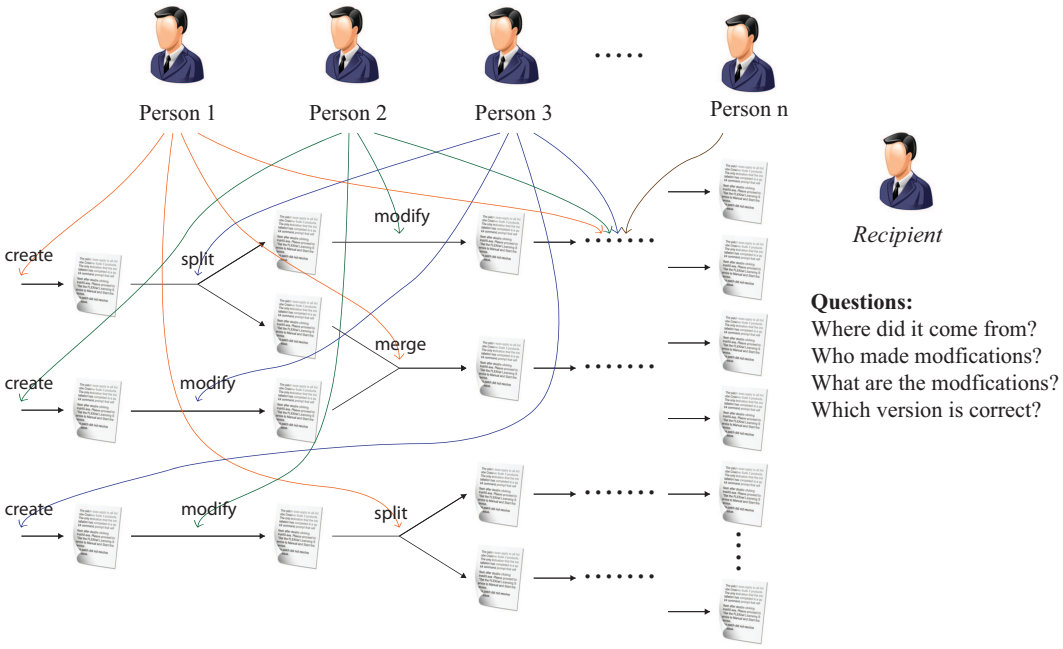

Figure 4.2: Multiple provenance paths

ciled. In cases where provenance paths provide conflicting information, a probabilistic approach might be applied to determine which provenance path should be accepted, if any. Table 4.1 summarizes the provenance path problem domains. Additional work needs to be done to research, design, develop, test, and validate solutions to the variety of problems present in the provenance path problem space.

4.4 A Case Study

Consider the case of the Justice Roberts rumor based on a simple investigation [13]. Reference Figure 4.3, a Georgetown Law School professor (node v_1) shared fictitious information in his class along edges e_1, e_2, and e_3. A student in the class, node v_3, sends a message to a blog site node v_5 along edge e_4, and the group at the blog site publishes a story based on false information. Similar provenance paths reach

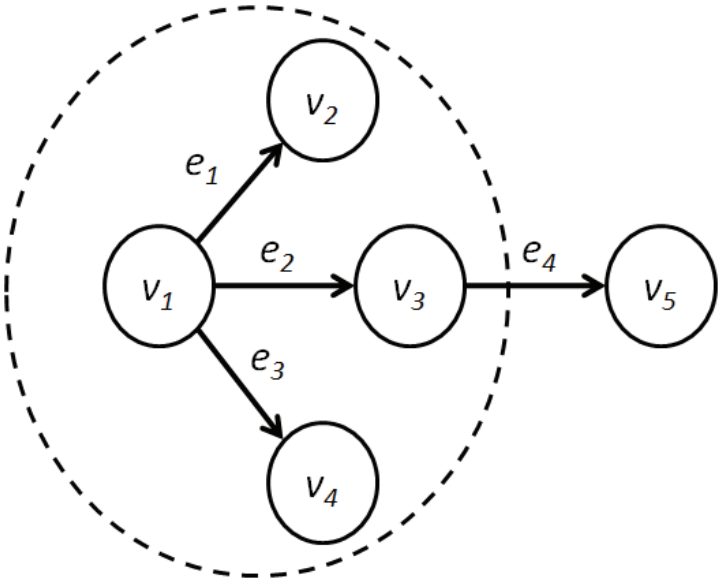

Figure 4.3: Case study diagram.

other blog sites and false information about a Justice in the United States Supreme Court becomes a well-circulated rumor.

The information communicated along e_4 may or may not be accurate. Given the provenance path shown in Figure 4.3, node v_5 should determine whether or not it should accept the information about Justice Roberts. If the recipient node v_5 analyzes the provenance path, and determines that it considers each node along the provenance path as accepted, v_5 could accept the information received via the explicit communication along e_2 and e_4. However, if v_1 or v_3 are discarded nodes, the recipient will need to consider what must be done in order to authenticate the information.

In Figure 4.4, an additional node, *JR*, is added to represent Justice Roberts. If node *JR* was included the provenance path, the information might be considered reliable. However, given that the node *JR* is not included in the path (as far as the

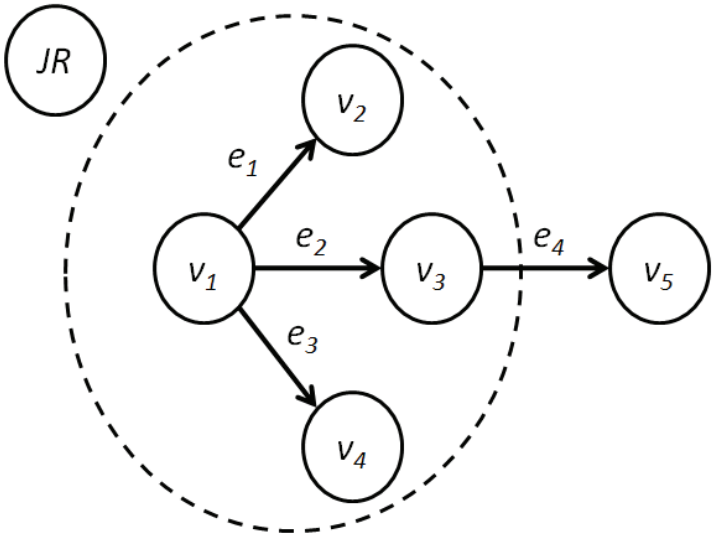

Figure 4.4: Case study diagram with additional node JR.

recipient can initially discern), questions should be raised about the validity of the information. In this case, direct or indirect connections using networking information and available social media could be examined to glean additional information. As examples, comparing the "distance" from node v_1 and node *JR* to a common reference point in social media, or analyzing the individuals' (v_1, v_3, and *JR*) group memberships and associated group traits.

The provenance path concept provides a framework for expatiating more specific techniques for finding provenance data in social media. In order to accomplish the task of assessing whether or not a node included in a provenance path should be accepted or discarded, a recipient needs a mechanism for specifying what meaningful provenance data is. Such a mechanism, provenance attributes, is presented next.

Complete provenance path	All nodes identified	Evaluating the provenance path may be as simple as traversing the path to determine whether or not a discarded node exists. When a discarded node exists in the path, a recipient may want to consider additional factors beyond the nodes and edges included in the path.
	Some nodes identified	Undecided nodes must be identified as accepted or discarded. When a discarded node exists in the path a recipient may want to consider additional factors beyond the nodes and edges included in the path.
	No nodes identified (all undecided)	All nodes must be identified as accepted or discarded. When a discarded node exists in the path a recipient may want to consider additional factors beyond the nodes and edges included in the path.
Incomplete provenance path	All nodes identified	Recipient must determine the most likely provenance path based on direct and indirect information available in the social media environment. Recipient may need to define threshold for acceptable path length (for intractable problem space).
	Some nodes identified	Undecided nodes must be identified as accepted or discarded. Recipient must determine the most likely provenance path based on direct and indirect information available in the social media environment. Recipient may need to define threshold for acceptable path length (for intractable problems space).
	No nodes identified	All nodes must be identified as accepted or discarded. Recipient must determine the most likely provenance path based on direct and indirect information available in the social media environment. Recipient may need to define threshold for acceptable path length for intractable problems space.
Multiple provenance paths	Multiple complete	Recipient can use provenance paths to authenticate or reconcile information.
	Multiple incomplete	Recipient must determine the most likely provenance path based on direct and indirect information available in the social media environment. In an intractable problem space, the recipient may need to define threshold for acceptable path length and criteria for reconciling accepted provenance paths with different lengths or heterogeneous characteristics.

Table 4.1: Provenance Path Problem Domains

Chapter 5

WORKING WITH PROVENANCE ATTRIBUTES

When a social media user receives a statement via a microblog message, a social network, or even a blog site, it is not always clear where the statement originated from, what motivated its publication, and what latent purposes may be associated with the particular message. In such circumstances, a user with additional metadata could make a better informed judgement about the information or statement received. For example, when the complete name, occupation, education level, and age can be associated with the originator of a statement, a user is better informed *about* the statement. In a particular domain, such as politics, a user may be interested in additional pieces of metadata. For example, a user with political interests may add to the list of desired metadata, political affiliation and special interests.

Provenance attributes are the metadata about the statement communicated through social media. Defining the specific pieces of metadata, or the *attributes*, a recipient is concerned about is a necessary prerequisite for finding usable provenance data in social media. The individual attributes that a recipient specifies as important are subjective based on the particular interests, values, and needs of the recipient. However, finding provenance attributes in social media can be measured objectively.

The subjective and objective aspects of provenance attributes enable the concept to be applied generally for any recipient that specifies what provenance attributes are important for their domain of interest. The recipient subjectively selects provenance attributes of interest, systematically works to find the attributes in so-

General demographic attribute set	Domain specific (political) attribute set
Formal Name (Individual or Group)	Formal Name (Individual or Group)
Location	Location
Occupation	Occupation
Education	Education
Age	Age
	Employer
	Political affiliation
	Lobby affiliation
	Special interest(s)
	Conviction(s)
	Citizenship
	Ethnicity
	Gender

Table 5.1: Provenance Attributes

cial media, and can objectively assess how accessible the attributes are to determine whether or not a provenance path is acceptable.

To further explore and illustrate the concept of provenance attributes, two sets of provenance attributes are specified for this research effort. Table 5.1 displays general and domain-specific attributes. The general set can serve as basis for other domain specific attribute sets. As an example domain-specific attribute set, the second column in Table 5.1 lists the attributes selected for a politically motivated provenance data attribute set. Both sets of attributes presented in Table 5.1 are based on standard demographic questions [14]. However, the current social media environment does not always provide this metadata with each individual message. Thus, provenance attribute data must be discovered or mined from social media.

As an example, consider a tweet from Antonio Villaraigosa, the mayor of Los Angeles, California, published in September, 2009[1]. The message is about a potential subway project creating jobs in Los Angeles. Given only the username,

[1]http://twitter.com/villaraigosa/status/4356459578, accessed on October 19, 2011.

Attributes	Source
Formal Name	Twitter profile
Location	Twitter profile
Occupation	Twitter profile
Education	Facebook profile
Age	Facebook profile
Employer	Facebook profile
Political affiliation	Facebook profile
Special interest(s)	Facebook profile
Conviction(s)	Facebook profile
Gender	Facebook profile

Table 5.2: Example Provenance Attributes Found

"villaraigosa", several provenance attributes can be obtained by openly public social media sources. Table 5.2 lists the provenance attributes that can be found for user villaraigosa through a public search of social media resources.

Specifying the particular set of provenance attributes that are of interest forms the foundation from which to begin the search for provenance data in social media. A successful search for provenance data in social media must address four challenges. First, the effort must begin from a starting point with a meaningful *signal* that can be used to direct the start of a search. Second, provenance attribute values must be *found*. Third, provenance attribute values must be *validated*. Finally, some of the duplicate attribute values might need to be *reconciled*.

5.1 Starting with Signals

Not all Twitter user pages contain data that can be mapped to attributes. Additionally, not all tweets contain a URL. It is clear that some tweets are more susceptible to mining provenance data than others. One metric for measuring the value of a microblog statement is *signal* [80]. Table 5.3 lists the characteristics defined as providing a good signal (out of the noisy statements that do not contain the charac-

Signal characteristic	Text indicator
Hyperlink	http://
Reference to another identifier	@
Hashtag	#
Retweeting	RT

Table 5.3: Twitalyzer signal characteristics

teristics) as used by Twitalyzer[2]. A statement with one or more of the signal characteristics included in the text increases the likelihood that provenance metadata can be discovered from a microblog statement. Statements containing hyperlinks can lead to web pages that provide additional information. Statements that reference another user identifier link the statement to another social media user. Statements with hashtags can be compared and contrasted to other statements containing the same hashtag. Retweeting can help link the statement to related statements or even additional identifiers.

From a database containing over 53 million randomly collected tweets, a large portion of the tweets have at least one signal metric characteristic in the message. Figure 5.1 shows the percentage of each characteristic individually and also a bar to indicate the percentage of tweets that have at least one of the characteristics. Over two thirds of the randomly selected tweets contain a signal characteristic that could be leveraged in a search for provenance data and, by extension, search for provenance paths associated with a statement published in social media (i.e., a tweet).

[2]http://twitalyzer.com

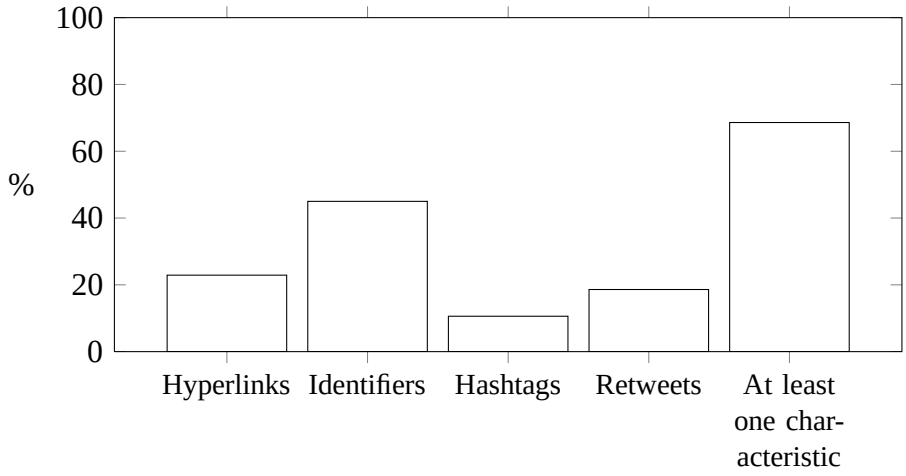

Figure 5.1: Portion of tweets with signal characteristics.

5.2 Finding Attribute Values

Finding provenance attribute values that are not readily provided, or trivially obtained, provides new information to a recipient. The following formal definitions help us to define provenance attributes and define a method for quantifying how much provenance metadata is available for a given microblog statement.

Definition: S is a microblog *statement* of interest to a recipient (i.e. social media user).

Definition: K is a set of *keywords*, $(k_1 \ldots k_m) \in K$, and $K \subseteq S$.

Definition: α is a unique microblog *identifier*, such as a username, associated with S.

Definition: A is a set of provenance *attributes*, $(a_1 \ldots a_n) \in A$, sought for any α. For example provenance attributes might include *name, occupation, education, and political affiliation.*

Definition: N is the number of provenance attributes sought after for any α. $N = |A|$.

Definition: W is the set of weights, $(w_1 \dots w_N) \in W$, associated with $(a_1 \dots a_N) \in A$.
Definition: V_α are provenance attribute values, $(v_1 \dots v_N) \in V$, the set of provenance attributes values associated with α. For example, the attribute values might be *Jeff, news anchor, republican,* and *unknown*.

In order to objectively quantify progress in obtaining provenance attribute values, an availability function is defined:
Definition: *information provenance availability* function,

$r : V_\alpha \rightarrow [0,1]$,
$r(V_\alpha) = \frac{\sum_{n=1}^{N} w_n \times x_n}{\sum_{n=1}^{N} w_n}$ where $x_n = 0$ if v_n is unknown, otherwise $x_n = 1$.

Problem Statement for Availability: Given statement S, keywords K, unique identifier α, and provenance attributes A with weights W; find attribute values V_α to maximize information provenance availability r.

The availability function quantifies how much provenance metadata is available for a particular statement. The availability function allows basic comparison of mining algorithms, search strategies, and prioritization of search results. Applications designed to automatically find provenance attributes can be compared based on the number of attribute values found. However, this is simply a beginning point for comparison because the provenance data availability function does not account for the validation aspect, i.e. were the correct attributes found.

In order to demonstrate how the provenance data availability function is applied, a simple example follows. The example tweet is sent by α "villaraigosa" and it includes the statement "MTA to pursue fed $ 4 Subway & Regional Connector! Projects that will cut pollution, create jobs and relieve traffic http://bit.ly/2vyBWK."

The tweet is amongst similar tweets containing the keywords “http://”, “cut”, “will”, “jobs.” Thus, S, K, and α are:

S = “MTA to pursue fed $ 4 Subway & Regional Connector! Projects that will cut pollution, create jobs and relieve traffic http://bit.ly/2vyBWK”

K = “http://”, “cut”, “will”, “jobs”

α = villaraigosa

To demonstrate the availability function, a subset of the domain specific attributes listed in Table 5.1 are used. Specifically, A = name, occupation, education, and political affiliation. Thus, $N = 4$. In this example weighting scheme, less emphasis is placed on the *occupation* attribute letting $W = (100, 50, 100, 100)$.

How can S be assessed from a provenance perspective (i.e., “information regarding the origins, custody, and ownership of” the tweet)? The provenance attributes desired, A (name, occupation, political affiliation, and education), are not available from the tweet alone. Thus, the provenance attributes must be discovered.

Beginning with the unique identifier α (villaraigosa), and any link information that is available, a search begins for provenance attributes. In this case, there is a link in the microblog. In other cases link information may not be available. However, searching the web and social media sites may reveal additional information, such as the Twitter user page associated with α. The Twitter user page for “villaraigosa,” `http://twitter.com/villaraigosa`, reveals name and occupation.

Note that the link contained in the tweet, `http://bit.ly/2vyBWK`, leads to a City of Los Angeles press release on Mayor Antonio Villaraigosa’s web page. By examining the press release, name is matched and occupation is found.

The provenance data available results in V = Antonio Villaraigosa, Mayor, unknown, unknown. Thus,

$$r(V) = ((100x1) + (50x1) + (100x0) + (100x0))/(100 + 50 + 100 + 100)$$
$$= 150/350 = 0.43$$

In other words, the information provenance availability of the tweet is computed to be 0.43 based completely on the provenance attribute data obtained from the Twitter profile page.

Given the name, additional provenance attributes can be found by extending the search to other social media sites. The public Facebook page, `http://www.facebook.com/antoniovillaraigosa`, reveals education and political affiliation. Continuing the example, searching for villaraigosa on the social networking site Facebook is helpful. Mayor Villaraigosa's Facebook page provides additional provenance attribute values. In particular, the attribute values for political affiliation and education are discovered, V = Antonio Villaraigosa, Mayor, Democratic Party, Juris Doctorate. With this additional information, the availability value is updated:

$$r(V) = ((100x1) + (50x1) + (100x1) + (100x1))/(100 + 50 + 100 + 100)$$
$$= 350/350 = 1.00$$

Someone new to Los Angeles, or in another geographic location, may not know "villaraigosa" is the Mayor (perhaps the message was forwarded by a friend). This fact that identifier "villaraigosa" is actually associated with the mayor of Los Angeles adds decision quality information about the tweet to better inform a recipient's understanding of the statement and reveal any latent motivation or biases.

The information provenance availability function provides a qualitative score to address the question of how much provenance metadata is available about state-

ment *S*. Additionally, the function accounts for variations in how important distinct pieces of metadata are from each other by weighting each attribute value that is identified during a search. The more provenance metadata that is available, the more a recipient can rely on the provenance information to help inform judgments about the statement. This provides a necessary foundation for provenance data in social media, but is only the first step.

5.3 Validating Attribute Values

Computing the availability of provenance attributes provides a basic means to assess the provenance data of interest. However, in the case when attribute values can be discovered, it is also important to know whether the attribute values are correct (i.e., valid) for the associated statement of interest, *S*.

One approach to validating attribute values is to use multiple sources to verify that a particular attribute value associated with α is consistent across multiple sources. For example, "villaraigosa" is associated with the name "Antonio Villaraigosa" on the Twitter profile and the Facebook profile. The occupation "mayor" is associated with the name, "Antonio Villaraigosa," in the Twitter profile, Facebook profile, and the City of Los Angeles page found via the link in *S*. The political party attribute value is found on the Facebook profile and is likewise returned through a simple search using Google[3](search for "antonio villaraigosa political party"). A search using the Google web search engine returns the political party from eight sources. Counting the number of sources that provided the same attribute value associated with α can provide a validity value for the provenance attributes associated with a specific statement. Dividing the total number of sources found by

[3]http://www.google.com/

Attribute	Example Source(s)	Source counter value
Formal Name	Twitter, Facebook	2
Occupation	Twitter, Facebook, URL	3
Political affiliation	Facebook, Google	7
Education	Facebook	1

Table 5.4: Example Provenance Attribute Sources

the average total number of sources found for similar messages for a particular domain, indicates whether the provenance metadata validity is above or below average. Specifically, we define a set of counters and an expected total count value as:
Definition: I_{V_α} are attribute value *source counters*, $(i_1 \ldots i_N) \in I$, for attribute values in the corresponding V_α.
Definition: C is the *expected total source count* for a particular set of provenance attributes, A.

An example set of attribute source counters for "villaraigosa" is shown in Table 5.4. C is calculated by summing the average counter values for a particular domain. In order to illustrate how provenance data values might be assessed for accuracy as described later in this section, we will assume the average counter values for each attribute are 2, thus, $C = 8$. Obtaining actual C values for particular domains of interests will be the subject of future research efforts.

The following function is proposed to quantify whether or not the attribute values found are valid:
Definition: *provenance data legitimacy* function,

$l : I_{V_\alpha} \rightarrow \mathbb{R}$,
$l(I_{V_\alpha}) = \frac{\Sigma_{n=1}^{N} i_n}{C}$, where $i_n =$ source count for attribute n.

Problem Statement for Legitimacy: Given statement S, unique identifier α, prove-

nance attribute values V_α, expected total source count C; find attribute values V_α to maximize information provenance legitimacy l.

As an example, given the assumption that $C = 8$ (based on a hypothetical average of two sources for each attribute in V_α) yields:

$$l(I_{V_{villaraigosa}}) = \frac{\sum_{n=1}^{N} i_n}{C} = \frac{2+3+7+1}{8} = \frac{13}{8} = 1.625$$

With a valid statistical value for C identified for a particular domain, when $l(I_{V_\alpha}) \geq 1.0$, the attribute set, V_α, is defined as legitimate. Additional research is needed to obtain valid statistical values for interesting domains such as politics, news, and entertainment.

5.4 Dealing with Duplicate Attributes

There are cases where finding and validating attributes associated with a statement S are a bit more complicated. Perhaps one of the most challenging aspects of obtaining useful provenance metadata is in circumstances where duplicate attribute values are found. Suppose that the statement of interest is a tweet from one of the 20 "Tom Jones" Twitter profiles. Given the username, α, some provenance attributes might be found in the publicly available Twitter profile. However, extending the search to other social media sites will force a choice of which "Tom Jones" is the person associated with statement S. With the assumption that the full name listed on the Twitter profile is correct, the search is continued on Facebook where there are 30 "Tom Jones" profiles available to choose from.

One approach for resolving duplicate attribute values associated with α is to reveal the correct association between the attribute values and α by comparing the friend network structure between social media sites and choosing the most prob-

ably match. The friend network associated with the "Tom Jones" on Twitter can be compared with the friend networks on Facebook in order to find the most likely match between the "Tom Jones" on Twitter, and one of the "Tom Jones" profiles on Facebook. The match is chosen based on the friend networks with the greatest overlap. In particular, the followers of α define the friend network on Twitter and can be considered as a directed graph with links from followers to $alpha$. The formal name associated with α listed in the Twitter profile is used to compare friend groups from other social media sites associated with the same (duplicate) formal name. In order to determine which duplicate name on Facebook is most closely associated with α, the friend network of α on Twitter can be compared to the friend structure of each duplicate name profile on Facebook. The friend networks on Facebook are represented as undirected graphs with edges between nodes of friends. The duplicate name profile with the greatest overlap has the highest probability of being the duplicate name that should be associated with α. This approach to dealing with duplicates has its roots in entity resolution research [16], link mining [35], and identity uncertainty [65].

The following definitions could be used to assess the probability of a matching a duplicate name with a particular α:

Definition: F_α is the set of of the names of α's *followers*.

Definition: F_η is a *set of friend names* associated with one duplicate name identifier on another social media site.

Definition: $p(F_\eta)$ is the *probability of the match* of F_η to F_α,

$p : F_\eta \rightarrow [0,1]$,

$p(F_\eta) = \frac{|F_\eta|}{|F_\alpha|}$

For example, suppose α has Twitter followers with names a, b, c, d, and e. $F_\alpha = a, b, c, d, e$. When the search extends from one side (say Twitter) to another social media site like Facebook, we look for the "Tom Jones" who has the most overlap with F_α. The *first* "Tom Jones" found on Facebook has friends b, d, e, thus, $F_\eta = b, d, e$, and:

$$p(F_\eta) = \frac{|\{b,d,e\}|}{|\{a,b,c,d,e\}|} = \frac{3}{5} = 0.60$$

Since there are 29 additional profiles on Facebook with the name "Tom Jones," $p(F_\eta)$ is computed for each "Tom Jones" profile. The "Tom Jones" profile with the greatest overlap has highest probability of being the relevant profile associated with α. Additional attribute values are obtained from the profile with the highest probability.

This approach to matching is used because of the differences in the network structure amongst disparate social media sites. For example, Twitter friend networks are effectively implemented as directed graphs and Facebook friend networks are implemented as undirected graphs. When extending a search from one social network site to another site with a similar friendship network structure (i.e., from Facebook to LinkedIn[4]), more sophisticated methods might be used for disambiguation similar to those applied to web pages as in [11].

5.5 Comparing Provenance Paths

A provenance path in social media, defined previously in this text and in [9], is a set of nodes and edges comprising a path which a statement published in social media information is communicated from a node in the graph to a recipient or recipients.

[4] http://www.linkedin.com/

This is an adaptation from the way others have viewed provenance as a directed acyclic graph (DAG) [28, 37, 59, 77]. Figure 5.2 presents an abstract provenance path and illustrates how a social media statement originating at node one may be propagated through nodes two and three to a recipient. The recipient could be an individual or a group. When a recipient can discern all of the nodes and links associated with a provenance path, the path is complete. If the provenance path exists but is not readily discernable to the recipient node, the path is incomplete and must be discovered by some process or mechanism.

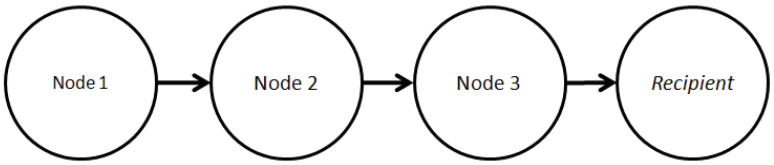

Figure 5.2: An abstract provenance path.

A provenance data search mechanism must be able to contend with incomplete paths (paths that exist but are not evident to the recipient). When portions or a path are missing, or the source of the path is not initially identified, the provenance search mechanism must jump to another segment of the social media environment. One strategy for making a jump is to choose the next social media site that boasts the largest number of users. Another strategy for making a jump is to begin the next step of the search on another social media site that represents an equally or more credible source of social media data. For example, some social media sites target working adults versus the general population.

A search mechanism could use three rules to differentiate “poor” versus “excellent” provenance paths in addition to the obvious considerations of structure

and path length. Rules used to assess provenance paths could be based on the *information provenance availability* function and *node discrimination*, supplemented with provenance attribute similarity for multiple provenance paths.

Information Provenance Availability

Information provenance availability could serve to prioritize various provenance paths. If a particular path provides information provenance availability values of less than 0.20 it might be considered poor, and greater than 0.90 might be considered excellent. The choice of threshold specific values of r to distinguish between poor and excellent is given as an example. Specific criteria should be defined based on domain expert input, recipient preferences, or detailed analysis of provenance path data and attributes for a particular domain.

Node Discrimination

Nodes included in a provenance path might be known prior to the discovery of the provenance path. Some nodes might be trusted or accepted by the recipient and others might be considered untrustworthy or rejected. Furthermore, the recipient may not know anything about other nodes along the path. A recipient could define node discrimination rules for labeling paths as poor or excellent based on the number of discarded or undecided nodes contained in a path. In general, an accepted provenance path would be labeled excellent using a node discrimination approach. If a path contains more discarded nodes than accepted nodes, it should be considered poor. Exact thresholds for the proportion of nodes used to distinguish between poor and excellent also should be defined based on domain expert input, recipient preferences, or detailed analysis of provenance path data for a particular domain.

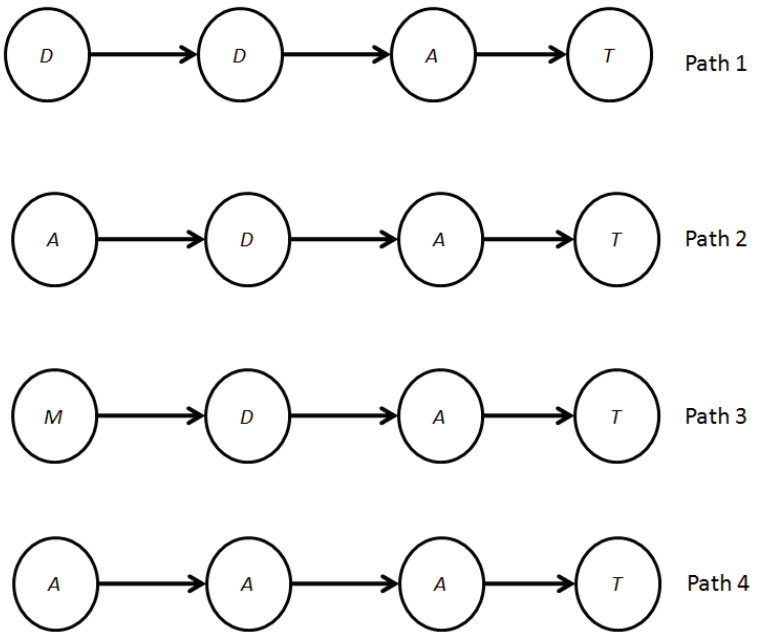

Figure 5.3: Provenance paths. Nodes labeled *A* are accepted, *D* are discarded, and *M* are unknown. *T* represents a recipient node.

Whether or not nodes are accepted, discarded, or unknown, will also inform how to assess path structure. Figure 5.3 illustrates four provenance paths with different numbers of accepted, discarded, and unknown nodes. Paths 1, 2, and 3, are heterogeneous paths. *Heterogenous paths* contain at least two different types of nodes. Both path 2 and path 3 have the same number of discarded nodes. However, the order of the two accepted nodes in the paths is different in each path. Path 3 is preferred over path 2 based on the position of the accepted nodes which are found closer along the path to recipient *T*. For example, *accepted* nodes might represent individuals that are part of a group of users working for the same firm. *Recipients* also work for the same firm as the accepted nodes. *Discarded* nodes are individuals that are working for a competing firm (i.e., viewed as potentially not credible). In

this hypothetical case, recipients should carefully consider paths that contain nodes associated with the competing firm. A node representing an individual not associated with any firm is *unknown* until the node can be assessed.

Path length can also be used as a gauge to judge the quality of a provenance path. Generally, shorter paths will be judged as better than longer paths. It is expected that shorter provenance paths will provide more accurate provenance data than longer paths, as has been shown in other areas of research like computing trust in web-based social networks [38].

A provenance search mechanism must have a strategy for dealing with incomplete paths (paths that exist but are not evident to the recipient). When portions of a path are missing, or the source of the path is not initially identified, the provenance engine will attempt to jump to another segment of the social media environment.

Decomposition, Analysis, and Recomposition

In some cases, recipients receive a message in social media with multiple statements resulting from people combining statements, repeating statements, or adding an additional statement to the message. Recall the example tweet from user villaraigosa referenced earlier in this chapter containing the statement: "'MTA to pursue fed $ 4 Subway & Regional Connector! Projects that will cut pollution, create jobs and relieve traffic http://bit.ly/2vyBWK". This statement can be divided into five shorter statements:

1. MTA to pursue federal dollars for subway.

2. MTA to pursue federal dollars for regional connector.

3. Projects will cut pollution.

4. Projects will create jobs.

5. Projects will relieve traffic.

This seems to complicate the problem of discovering a provenance path because the final message received may be the result of a combination of provenance paths. In these circumstances, the question is raised, “What is the best way to decompose, analyze, and recompose, the provenance data for the message?”

There is a distinction between determining whether or not a statement is true, and determining the information provenance of the statement. The goal of discovering and revealing the provenance data about a statement is to disclose the origins, custody, and ownership of the information. Provenance data will assist a recipient in making a decision about whether or not the information is true or false but the provenance data alone will not necessarily certify the statement. In a sense, this distinction simplifies the decomposition, analysis, and recomposition steps. The steps for analyzing provenance are simpler from more complicated domains because statements can be treated independently.

One option is to consider the provenance path separately for each piece of information. For example, given that a recipient receives statements *A*, *B*, and *C*, contained in a single communication via social media, such as a microblog or a wall posting on a social network site, provenance data might be sought independently for each statement contained in the message.

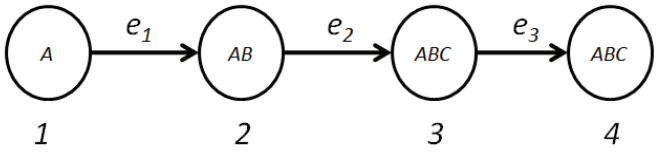

Figure 5.4: Communication with multiple statements.

Figure 5.4 illustrates the case when the recipient, node 4, receives a single communication with multiple (proposed factual) statements. In this example, the communication originated with information *A*, at node 1, and was appended with additional information, *B*, at node 2, and *C*, at node 3. Figure 5.5 illustrates how three independent sets of provenance data, one for each statement, might be represented.

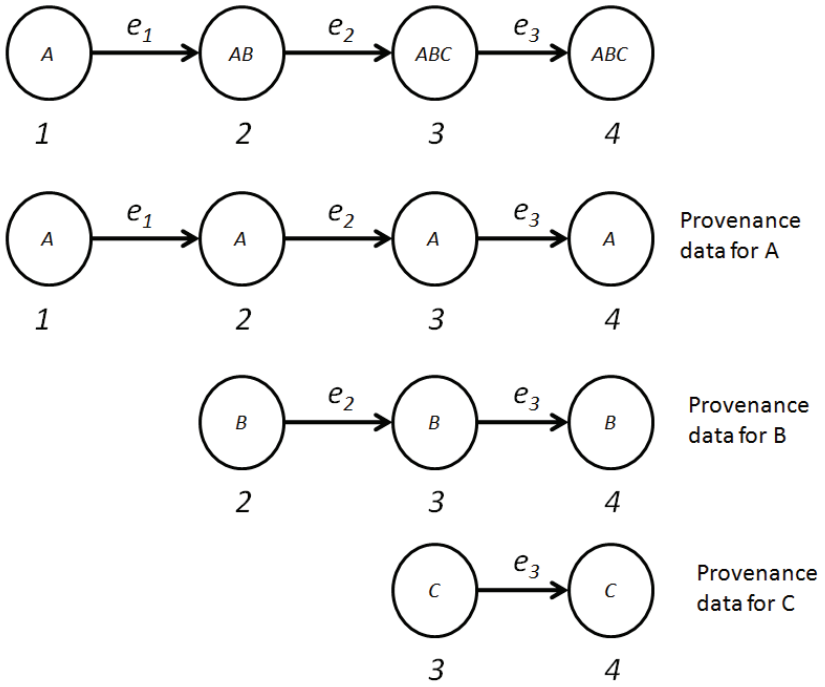

Figure 5.5: Decomposition of statements.

This poses the question, “could provenance be considered independently using criterion specified by the recipient, node 4?” For example, suppose the recipient specifies a set of provenance attributes of interest. A separate provenance availability value could be calculated for each statement. Recomposition might be accomplished using a representative provenance availability value for the communication. Specifically, a representative provenance availability value, r_{rep}, might be considered as:

$$r_{rep} = \frac{\sum_{n=1}^{N} r_n}{N}$$

Where N is the total number of individual statements, and r_n is an independent provenance availability value. For the example illustrated in Figure 5.5:

$$r_{rep} = \frac{r_A + r_B + r_C}{3}$$

Another questions is, “Would it be more helpful for the recipient to skip recomposition and consider the statements separately?” This seems logical when the cumulative availability value is low, perhaps indicating that a significant amount of provenance attributes could not be identified. Alternatively, when the provenance data for two statements are identical (availability value and path structure), the statements might be recomposed and considered together.

The ability to discern and analyze provenance paths in social media is an important part of finding provenance data in social media. Additional research is needed to help address questions related to decomposition, analysis, and recomposition. In the future, additional work should include the test and validation of metrics that will enable provenance paths to be revealed and assessed.

Chapter 6

SEEKING ATTRIBUTE VALUES

A solution for provenance attributes and a provenance path must be provided in order to provide useful provenance data to an end user receiving a statement made in social media. The end user that receives the statement from social media, and is inquisitive about the provenance data associated with the statement, is simply known as the *recipient*[1]. An approach for finding the provenance attributes for every node in a provenance path is needed. With attribute and path information accompanying a statement, a recipient can better assess whether or not ownership might be a factor to consider.

Recall the discussion and definition of a *provenance path* presented in Chapter 4 and the discussion of *provenance attributes* and the *information provenance availability* function presented in Chapter 5. The provenance path problem is essentially an extension of the provenance attribute problem. In the simplest case, a statement is made directly from a social media user to a recipient (a path with two nodes and one edge). In this case, the provenance path problem *is the same* as the provenance attribute problem, such as the case when a recipient is reading a tweet that was not retweeted. However, when the provenance path contains more than one node along the path to the recipient, the problem evolves to that of maximizing information provenance data availability r along the *entire* provenance path, excluding the recipient.

Provenance Path Problem: Given statement S, keywords K, provenance path p, and provenance attributes A with weights W; find attribute values V_α for each

[1]The recipient can be an individual or a group.

node in p to maximize provenance data availability r for each node in p. In other words, maximize $\sum_{x=1}^{n-1} r(V_\alpha)_{v_x}$ where $v_x \in (v_1 \ldots v_{n-1})$ are nodes in the provenance path (v_n is the excluded recipient node).

Using data from the Arizona State University Data Mining and Machine Learning (DMML) laboratory[2], 300 tweets were selected for manual analysis divided into two sets. The number tweets was limited to 300 to ensure the manual analysis could be completed in a reasonable time frame and adequate time would be available to complete the other research tasks planned for this effort. One set of 150 tweets was used to manually explore searching for general attributes, and the other set of 150 tweets was used to manually explore searching for a set of domain attributes (political). The goals of the manual analysis activities were to:

- Investigate processes that would be effective for mining provenance attribute values.
- Understand the problem space pertaining to finding provenance data in social media.
- Identify issues and challenges pertaining to mining provenance attribute values.
- Collect baseline performance data for comparing the manual analysis with automated means.
- Initialize a technical foundation for future research efforts.

[2]Twitter data provided by DMML colleague Mohammad-Ali Abbasi.

The purpose of dividing the tweets into two sets was to highlight different challenges that might be uncovered when searching for provenance metadata in different domain areas. Initial results show that there are differences in the domain areas that may impact the ability to find sufficient provenance metadata about statements made in social media.

The first set of data that was collected and manually examined was used to study the availability of general attributes. *Formal name*, *location*, *occupation*, *education*, and *age* were the provenance attributes used for the general attribute set. The idea is to begin to understand how much general metadata is available and contrast it with a domain-specific attribute set. Name, location, and age are common survey questions and are included in public surveys such as the 2010 United States Census[3]. *Occupation* and *education* are amongst the additional information that is sought during the Current Population Survey (CPS) conducted by the Bureau of Labor Statistics and the United States Census Bureau[4].

The DMML Twitter database contains over 50 million microblog statements better known as "tweets." The tweets in the DMML database are obtained using a crawler application which randomly collects tweets from Twitter [5]. The tweets are stored in an SQL database along with information about the user associated with each tweet. The criteria for the general attribute set was a set of keywords: "http://", "job", and "growth". The selection of the keywords takes into account the previous definition of the provenance availability function, and is meant to be

[3]See http://2010.census.gov/2010census/text/text-form.php, accessed on October 19, 2011

[4]http://www.census.gov/apsd/techdoc/cps/CPS_Interviewing_Manual_July2008rv.pdf, accessed on October 19, 2011

[5]140 million tweets are published each day (http://blog.twitter.com/2011/03/numbers.html, accessed on October 19, 2011).

representative of a common interest area, employment. Recall that the availability function assumes a set of keywords, K.

Additionally, “http://” is included as a keyword with the motivation that some provenance attributes might be found by inspecting hyperlinked documents referenced in microblog statements as motivated by the thinking that the URL is a good signal characteristic as implemented by Twitalyzer [80]. Although the Villaraigosa case study, conducted in conjunction with the research proposal, led to a hyperlinked page which did provide additional provenance metadata about the microblog statement, the majority of hyperlinked text was not observed to provide additional information about the user-publisher of the microblog messages studied manually.

6.1 Manual Analysis

Of the 150 tweets selected for analysis in the general domain, ten tweets were removed from the manual list because the site URL was not available (eight) or the profile was suspended (two). It is important to note that the fact a profile is suspended is, in and of itself, valuable data. A boolean attribute that represents whether or not an account was suspended should be included in future attribute sets. Another 86 tweets were eliminated for further processing because they were likely originating from corporate entities or advertising organizations. One tweet was removed because the username, α, was duplicated (i.e., tweets from the same user were included in the DMML Twitter data set). Two messages were eliminated from the data set because the biography section of the profiles included languages

other than English. After manually preprocessing the messages for the reasons described, 54 tweets remained for manual analysis.

The criteria for the political attribute set were a set of keywords: ”http://”, “election”, and “12”. The motivation for selecting the keywords was based on upcoming elections in the United States for the year 2012 with the assumption that this would be a topic of interest and discussion among microbloggers. However, many of the tweets returned by the search query were statements made pertaining to elections in the nation of Iran. This was unexpected, nevertheless, the manual search still revealed interesting aspects of the problem space. The political attributes set includes *formal name*, *location*, *occupation*, *education*, *age*, *employer*, *political affiliation*, *lobby affiliation*, *special interests*, *convictions*, *citizenship*, *ethnicity*, and *gender*. This attribute set extends the general attribute set by adding additional common demographic questions including *employer*, *convictions*, *ethnicity*, and *gender* [14]. Attributes related to *Political affiliation*, *Lobby affiliation*, and *Special interests* are motivated by the types of questions[6] and results reported by political exit polls[7].

Of the 150 tweets retrieved for the political attribute manual analysis, 10 tweets were dropped from manual analysis because the profile site was unavailable (i.e., no longer exists), the profile was suspended, or the account was suspended. Eighteen tweets were dropped because the text was not in English. Surprisingly, 55 messages listed locations outside of the United States. Nine messages were linked to corporations or news agencies. Finally, one tweet was dropped because it was

[6]For example, http://election.cbsnews.com/campaign2008/pdf/NH_Dem_FINAL.pdf, accessed on October 19, 2011

[7]For example http://www.cnn.com/ELECTION/2004/pages/results/states/US/P/00/epolls.0.html, accessed on October 19, 2011

from a duplicate user. After the manual preprocessing was completed, 53 messages remained for manual analysis. Not all of the tweets in this dataset were political in nature because words like "selection[8]" also satisfied the database query used to select the subset of tweets for the study.

For each set of data, the following process is used to search for provenance attribute values. First, the Twitter username, α, was used to identify the Twitter profile page. The formal name attribute value was obtained by using the name value provided on the Twitter profile page. The location on the Twitter profile page was used as the string value for the location provenance attribute. The Twitter biography was used to obtain additional provenance attribute values such as occupation, age, and in some cases employer, political affiliation, and gender. However, not all Twitter users have a complete profile published on their profile page.

When the user does not have complete information listed on their profile page, it is necessary to search other sources for attribute values including the hyperlink associated with the microblog statement, other social networking sites such as Facebook, LinkedIn, and MySpace. Additionally, search engines such as Google and Bing can be used to search for additional provenance attributes using queries comprised from attribute values obtained earlier in the search process.

The Twitter profile page allows users to publish a public profile. Users can provide a name, location, web address, and a free text biography[9] limited to 160 characters. Information on the profile page serves as a starting point for the manual search. Surprisingly, some biography sections of the profile contained very de-

[8]Contains the substring "election".
[9]Listed as "Bio"

tailed information such as age, names of relatives, and even ages of relatives. Some biographies also listed employer information. While unexpected, this finding highlights the wide variety of data to support various provenance attribute sets relying on social media data.

Searching Facebook and LinkedIn required some duplicate names to be resolved. To resolve duplicate names, location and profile photos (if available) were used to manually match the user on other sites or web pages that corresponded with the α identified originally using the data available from α's Twitter profile.

Some Twitter profiles listed a URL that provided additional provenance attribute data. Finally, a web search for the user combined with other provenance attributes was used to search for additional provenance attribute values. In a few cases, the web search provided links to additional social networking sites such as MySpace[10], or the user's blog site. In some cases, profile attributes can be verified and in other cases, the additional attributes were found. Figure 6.1 outlines the process followed for the manual search.

This manual search for provenance attributes provided some interesting insights into the problem space:

- First, there was more data in twitter profiles than anticipated for some users. For example, some users listed age, political preferences, and at least one user included information about grandchildren. This was surprising and also somewhat alarming from a security and privacy perspective. A complete set of general attribute values was found for four of the tweets. At least one

[10]http://www.myspace.com/

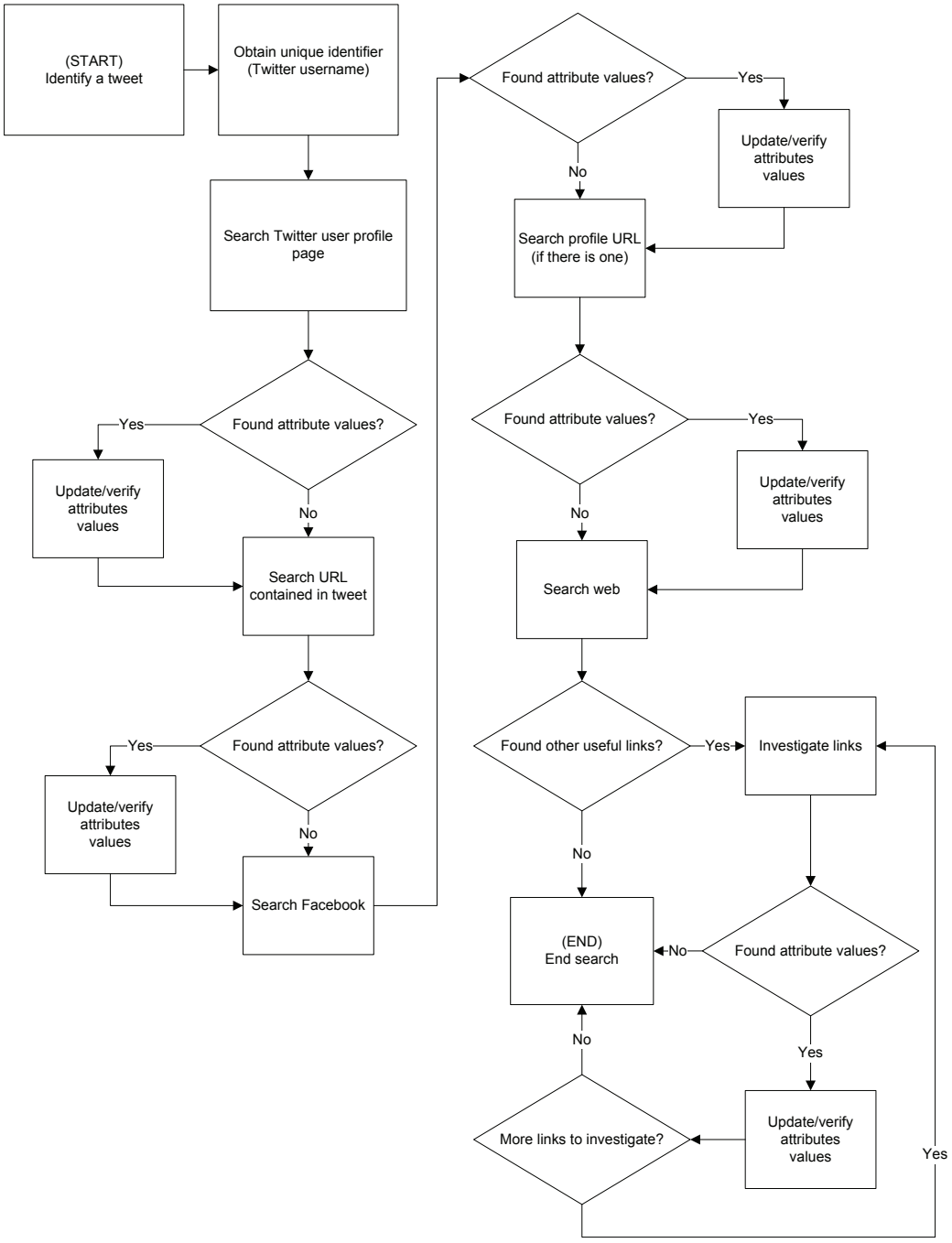

Figure 6.1: Manual search process for provenance attributes

attribute value was identified for all of the tweets investigated in the manual analysis for the general attribute set. Figure 6.2 summarizes the overall percentage of attribute values identified in each category.

- Not as many political attribute values were obtained manually as anticipated. Only 26% of all the desired attribute values were obtained. Figure 6.3 shows the percentage of each type of provenance attribute identified during the manual search.

- The URL links in the microblog message itself were not useful for obtaining provenance attribute values. *K* contained "http://" with the idea that it would provide an additional mechanism for identifying provenance attributes supported by the notion that tweets including "http://" can be preferred as a "signal versus noise" [80]. However, all of the URLs contained in the tweet text linked to news articles or web sites that did not provide additional provenance attribute data values.

- The URL listed for some users on their Twitter profile page was useful in some cases (more so than the URL in the message). Note, the URL listed on the profile page (if any) is not the same URL that is included in the statement *S*.

- Public Facebook profiles were easier to search if the author was logged in as a Facebook user (i.e., publicly available Facebook profile pages did not provide as much of the desired data thought possible.) However, a positive match for some individuals on Facebook was realized by manually matching the profile pictures in order to link some users across disparate social media sites, and to

resolve the entity resolution problem for some individuals. It is anticipated that automatically matching profile pictures would prove more challenging. However, a photo recognition capability would be a good mechanism to link people across sites because, in some cases, the profile picture is the same across social media sites.

- "Simple" web search proved very useful by providing links to sites with other profile data including social networking sites, blog posts, and personal web sites.

- Politicians appear to be more public about political attributes. As one would expect, political figures appear to be more open about political views, etc.

- In no case was the core meaning of the original message changed. This is likely due to the short length of the message. The search criteria may have also influenced the selection of a set of tweets that would not likely be modified. For example, had the search criteria included "RT" users may have been more likely to append, comment, or modify the original message. However, "RT" was not used as part of the search criteria to select the two sets of tweets manually analyzed. This finding may be unique to Twitter and might be different given different social media sites such as Facebook[11].

- As anticipated, dealing with duplicate identities is a significant challenge that must be reckoned with when searching for provenance attributes. In the manual search, this was addressed by using images, and combining provenance

[11]Facebook comments serve as a form of modification to a message. In the case of a Facebook message or post which includes comments, the immediate user would have some provenance data about the author based on the users Facebook network of friends

attributes as they are found. However, more sophisticated means for dealing with duplicate identities are needed if automatically searching for provenance attributes is to be fully realized in the future. One strategy for dealing with duplicates is to compare friend networks of social media users. In this manner, duplicate names might be resolved by finding friend networks that are most alike, that is, contain the same friends or the most friendly matches. More sophisticated means like "identity resolution," developed by IBM's Jeff Jonas [48], for dealing with duplicate identities, might be more effective.

- Both sets of tweets used for the manual search yielded similar results for the five attributes common to both attribute sets. The bar graph in Figure 6.4 gives a visual representation of the comparison. it is important to note that based on this comparison, it appears evident that although domain specific provenance attributes may differ in composition, the ability to mine *basic* provenance attribute values is likely not dependent upon the domain. Figure 6.3 illustrates that although some domain specific attributes might be highly desirable, it may be very difficult to obtain attribute values due to privacy practices, site security policies, and user personal preference. However, there can be value in seemingly unavailable attributes. It is well known that the lower the probability an event has of occurring, even greater amount of information is provided when the event occurs [34]. For example, ethnicity, citizenship, and lobby affiliation were rarely found, if at all. Thus, when a rare attribute value is located the provenance data provides an even greater amount of information to the recipient-user.

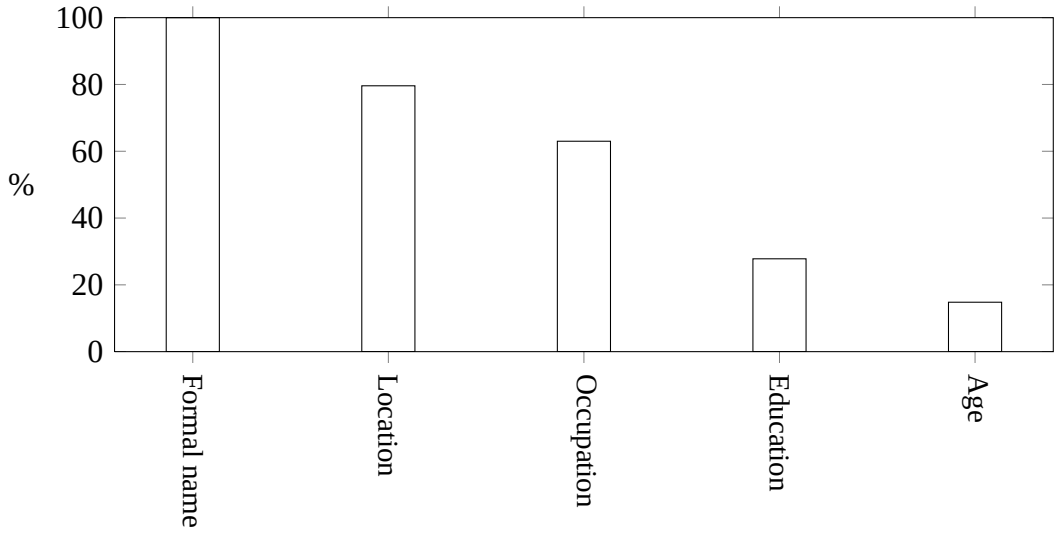

Figure 6.2: Percentage of general domain attributes found manually.

- Lastly, it is noted that the Twitter biography, if available, might be a valuable and unique provenance attribute for provenance data associated with a Twitter user. The biography provides, in some cases, a significant amount of provenance data including age, occupation, employer, political affiliation, and interests. Additionally, the Twitter biography can also provide statements indicative of opinion, attitude, and sentiment that are best interpreted by a human recipient-user. Since the Twitter biography is limited to 160 characters, including the entire biography as a provenance attribute in the future may prove valuable for some recipients. The Twitter biography could serve a dual purpose; a source of provenance attribute data and as a provenance attribute itself.

The manual analysis provided valuable insights into the challenges and opportunities of using social media itself to provide provenance data about statements

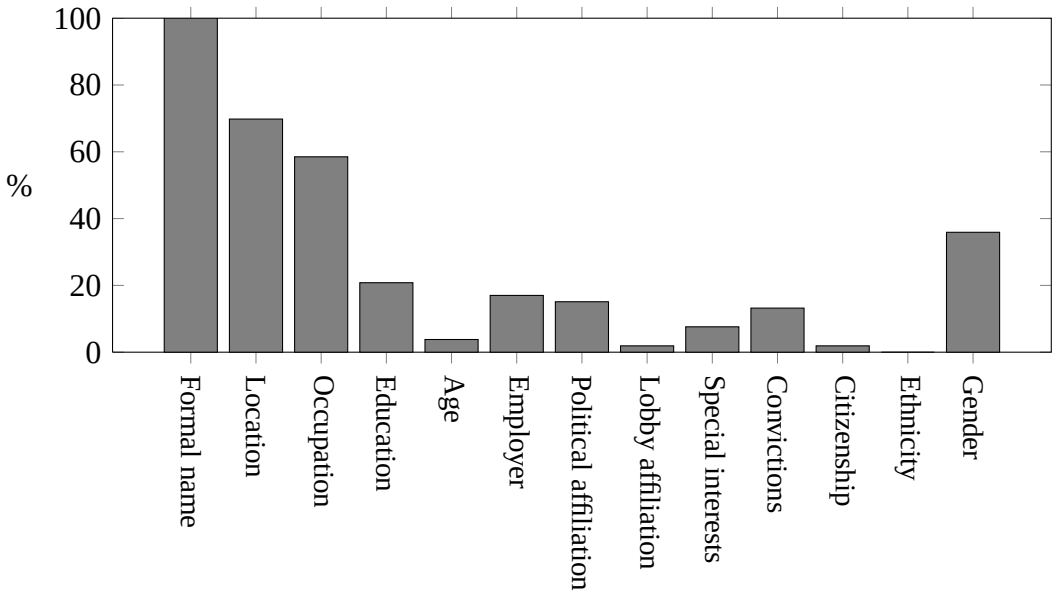

Figure 6.3: Percentage of political domain attributes found manually.

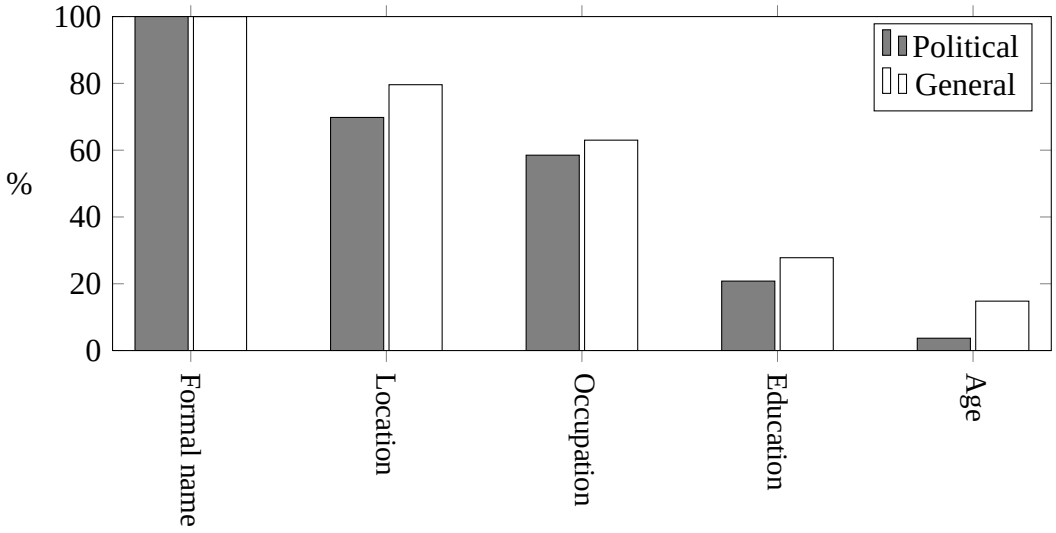

Figure 6.4: Comparison of percentage of common attributes found manually between the sets of “general” and “political” tweets used for manual analysis.

made in social media. Although not as many attribute values were found as anticipated, a significant amount of attribute data was discovered for the tweets included in the two sets of research data. In some cases, it was extremely exciting to see that individuals can be identified across social media sites and that additional provenance attribute data can be obtained as individuals are examined across disparate sites. With the manual analysis completed, efforts turned to automating the search for provenance attributes.

6.2 Automated Analysis

With insights learned from the manual analysis, an application was designed to automatically search for provenance attributes associated with a Twitter username. The application was built around the vision of a Provenance Engine. The concept of a Provenance Engine is depicted in Figure 6.5. The Provenance Engine takes as input: a statement S, α associated with S, a set of provenance attribute A, and the associated set of provenance attribute weights W. The Provenance Engine application searches social media sites for attribute values. The Provenance Engine application outputs the associated provenance attribute values V_α, the provenance availability value $r(V_\alpha)$, the provenance legitimacy value $l(I_{V_\alpha})$, and the set of provenance paths P (or likely provenance paths).

Two different approaches were envisioned for the automated process. First, "scraping" provenance attribute values from web pages directly. Second, using social media service Application Programmer's Interfaces (APIs) to request data directly from service providers. The assumption was that the APIs would provide easier access to user profile data from each social media service. However, in the

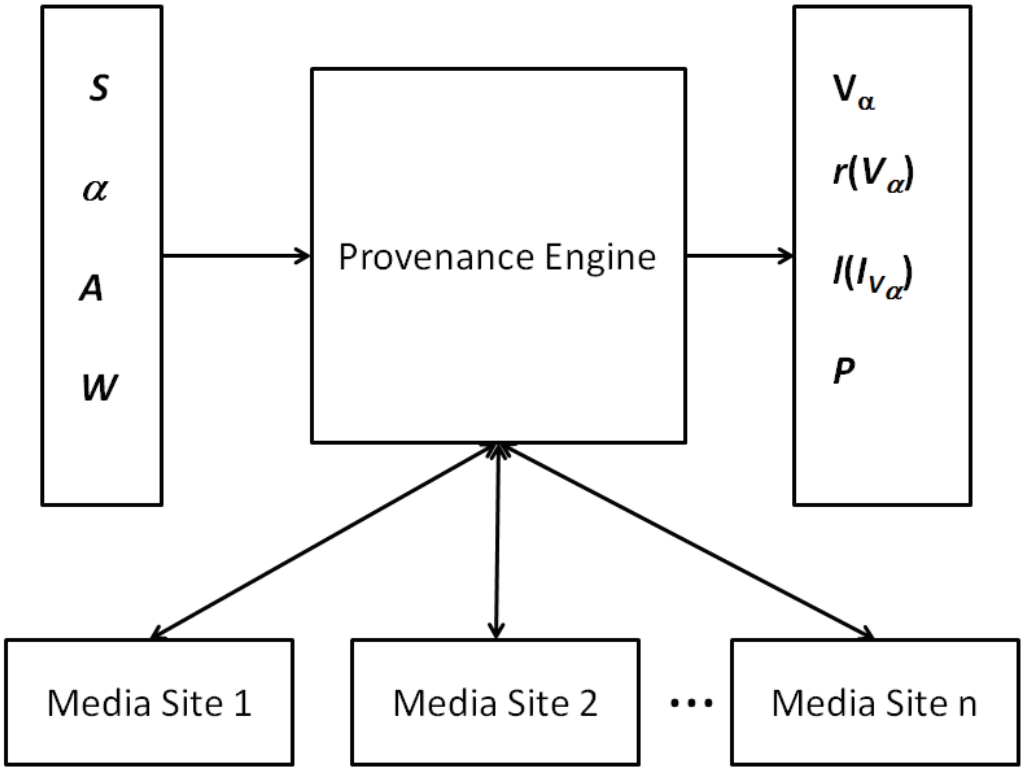

Figure 6.5: Provenance engine concept

end, a hybrid approach works best taking advantage of open APIs and publicly available profile data.

The automated process follows similar, but simpler steps of the manual process. Figure 6.6 illustrates the process flow. After a tweet of interest is identified, α (Twitter username) is used to search Twitter data for the profile associated with α. Ideally, at least a formal name and location are returned from the profile. If no profile data is available for the associated Twitter username the search is ended. After available provenance attribute data is captured from the twitter database, the search for profile attributes continues with data available from LinkedIn.

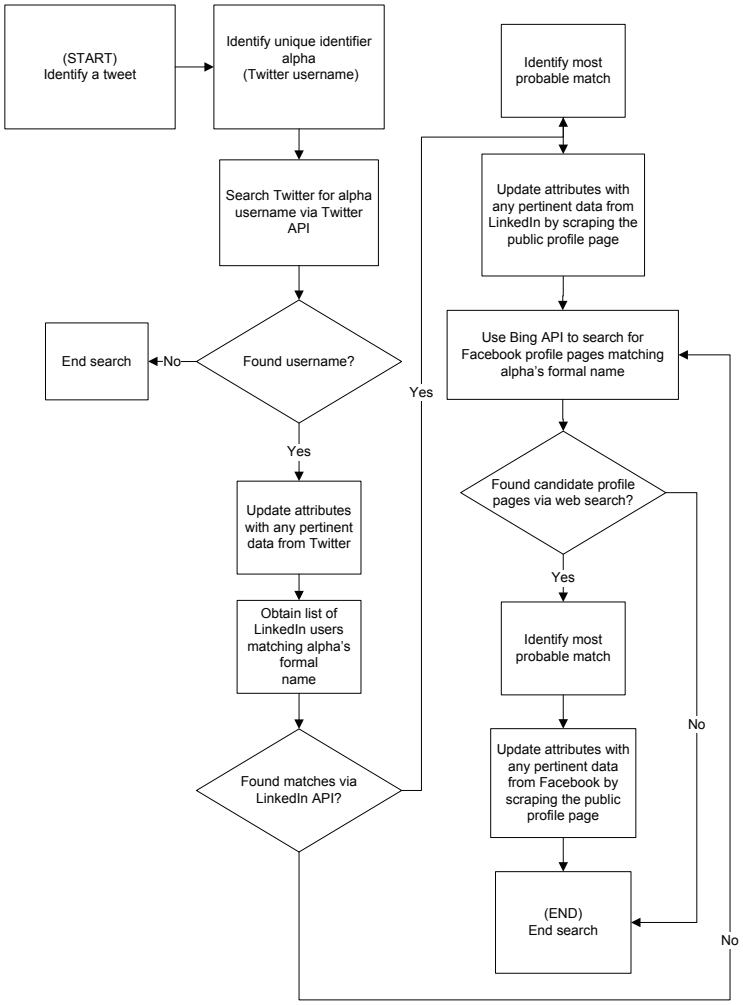

Figure 6.6: Automated Search process for provenance attributes

LinkedIn is an online social networking service use primarily for professional contacts. LinkedIn was chosen as the next search site based on the assumption that LinkedIn users are motivated by professional and business aspirations, and it would logically follow that data in a LinkedIn profile is more likely to be accurate and less likely to be falsified or purposely incorrect. Thus, LinkedIn data is assumed to be more accurate than other social networking sites such as Facebook and MySpace. Beginning with data sources that are likely to be more accurate improves the probability that provenance attribute data can be used to accurately portray α as the search for provenance attribute values continues. Second, the LinkedIn API is easy to access and consequently public profile pages are easy to mine for provenance attribute values.

The location string obtained from the Twitter profile is compared with the location of each LinkedIn profile that matches the α formal name. Duplicate names are resolved using the string values for the location attribute. The location strings are compared using edit distance. The lowest edit distance indicates the most probable match between the α's Twitter profile and a LinkedIn profile. Although not perfect, this approach provides a simple means for resolving duplicate identities and demonstrates how a more sophisticated assessment criteria might be integrated into future versions of the application. If there are no LinkedIn profiles that match, the search continues on the next planned social media site.

Once the most probable LinkedIn profile is identified, the public profile URL returned by the LinkedIn API is used to access and download the public profile page for the LinkedIn user. The application scrapes the profile page for any additional provenance attribute values. After updating α's provenance attribute val-

ues with any data from LinkedIn, the application moves on to search for potential attribute values in a Facebook profile.

Instead of utilizing the Facebook APIs, the Bing search API was employed to search for public profile pages matching α's formal name. The same process was used to deal with duplicates (using attribute values previously obtained). If the search results do not provide options for the formal name, the search is ended. Of course, if a formal name is matched with α's formal name and location, α's attribute values are updated and the search is complete.

A simple Provenance Engine was implemented in the Java programming language with the Netbeans[12] Integrated Development Environment (IDE). Additionally, a MySQL[13] database server was used to store the provenance attributes that were found for each α by the Provenance Engine for detailed off-line analysis. Figure 6.7 presents an example of the Find Provenance Attribute window that was implemented in the Provenance Engine application developed as a part of this research effort.

The application implements and automates the provenance attribute search process detailed in Figure 6.6. The Provenance Engine attribute search function implemented and depicted in Figure 6.7 searches for provenance attributes for one α (Twitter username) at a time. Text boxes display attributes associated with three potential sources of provenance attribute data (Twitter, LinkedIn, and Facebook) identified by the Provenance Engine associated with α. Select examples of the Provenance Engine Application source code are included in Appendix C.

[12]http://www.netbeans.com/
[13]http://www.mysql.com/

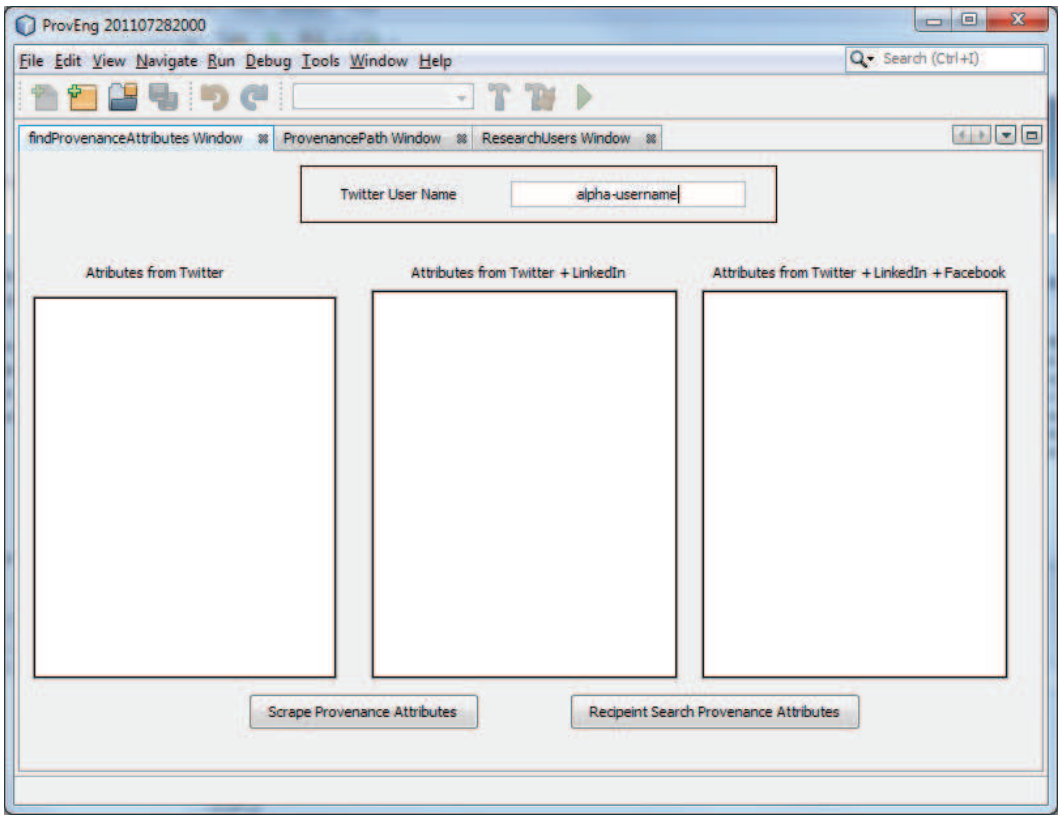

Figure 6.7: The Find Provenance Attributes window allows a recipient to enter an α user name associated with Twitter and to determine what provenance attribute values can be found.

This simple process implemented in the Provenance Engine application provided some interesting results comparable to the manual search.

6.3 Automated Search Results

The same 54 Twitter users from the "general" data set and 53 Twitter users from the "political" data set were used as a test set during the automated search experiment. The same methods used to implement the functionality, shown in the Find Provenance Attributes window of the Provenance Engine, shown in Figure 6.7, were used to collect data for all of the users for the "general" and "political" data sets.

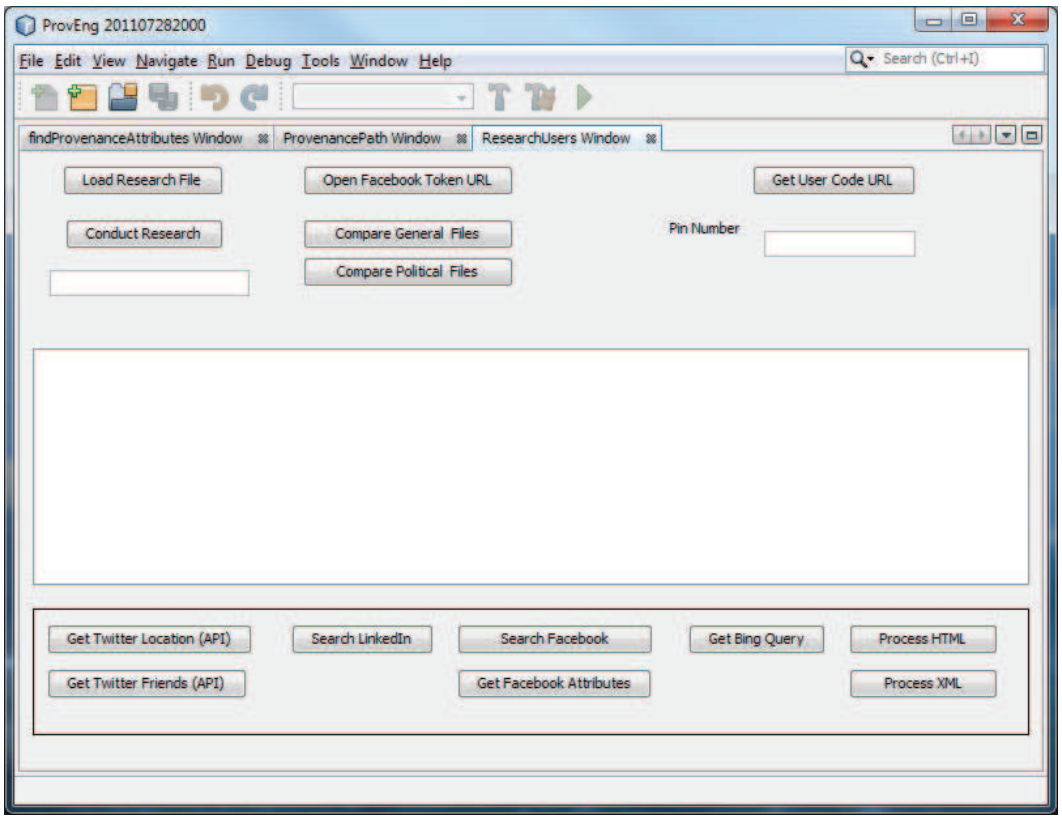

Figure 6.8: Research Users Window

The search results for each user were saved automatically to a MySQL database. Figure 6.8 shows the interface that was developed for the research experiment and application testing.

Figure 6.9 presents the percentage of general provenance attributes that the Provenance Engine application found for the general data set contrasted with the percentage of general provenance attributes that were found during the manual analysis. The manual search for provenance attributes yielded the same or more attribute values for four of the five general provenance attributes. The Provenance Engine returned more values for the *Location* attribute than the manual search.

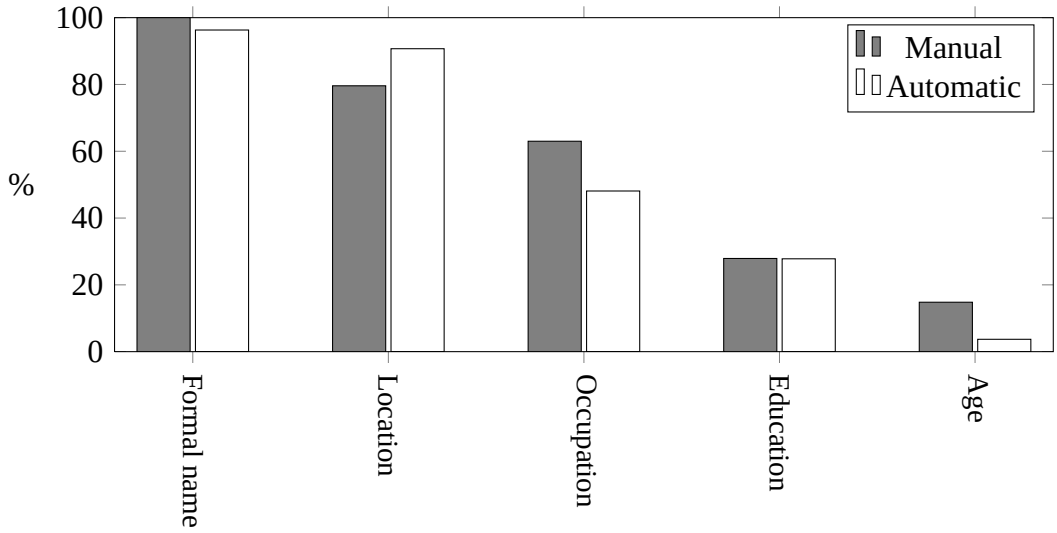

Figure 6.9: Comparison of percentage between manual and automatic search of general attributes.

However, closer examination reveals that the additional location values that were returned by the automated process do not always provide accurate data for the *Location* attribute. For example, "USA" and "Everywhere" were two of the location values returned by the Provenance Engine. While "USA" and "Everywhere" do not provide the same type of specific location data returned for other users[14], these types of general location values can still provide meaningful context to a social media statement. A user who has a *Location* attribute value that is more abstract may also convey a sentiment to a recipient that has some utility for the recipient to make a judgement when combined with additional provenance attribute data.

Although the automated search for provenance attributes yielded very similar numbers of attributes, it is not sufficient to judge the success and potential of the automated search process based purely on the number of attributes that were

[14]Many users had specific City-State pairs for a location attribute value including "Portland, OR" and "Salt Lake City, UT.

Provenance Attribute	Number of Mismatches
Formal name	0
Location	2
Occupation	12
Education	5
Age	0

Table 6.1: Mismatched attribute values for the general data set.

returned by the Provenance Engine application. Whether or not the Provenance Engine returned the same attributes that were found during the manual process is also important to consider[15]. To compare the performance of the automated search to the manual search, a short program was written to compare the attribute values identified by each method. For instances in which both the manual analysis and the automated analysis yielded an attribute value for a user, a string comparison was performed. The results of the comparison for the general demographic attribute data set are presented in Table 6.1. Eleven users, approximately 20% from the general data set, had at least one mismatched attribute value between the manually and automatically obtained values.

As in the case of the manual analysis, dealing with duplicate identities provides a challenge for automated analysis. For example, one user with a common first name Scott, and a common last name[16], was matched on Facebook during the manual search, but the automated search yielded a different attribute value attributed to a different Scott with the same last name. Since the manual analysis and the automated analysis were not conducted concurrently[17], some users updated

[15]The assumption is made that the manual process returned the correct attribute values associated with the αs used in the study.

[16]Last name withheld to protect privacy in accordance with Arizona State University Institutional Review Board (IRB) directions.

[17]The automated analysis was conducted a few weeks after the manual analysis.

their profiles with new data, or removed data, after the manual analysis was completed. For example, one user's location changed from particular city in California, to "Sunny California," and yet another user listed a particular city in Texas and changed to "Central Texas." It is important to note that some string mismatches were not semantically different. For example, the strings "TX" and "Texas," as part of the Location attribute, are semantically the same.

Some of the attribute values retrieved make clear that a future version of a Provenance Engine application could leverage more sophisticated text processing techniques. For example, one user lists several occupations such as "author" and "professor." This situation presents an interesting question from the provenance perspective: Which occupation best describes the user to the recipient? It is intuitive that providing data about both occupations can be valuable to the recipient but should one be emphasized over another and if so, which one?

The results of the comparison for the political attribute data set are presented in Table 6.2. Similar to the general demographic data set, most of differences in attribute values associated with the same α in the political attribute data set appear to be caused by updates to user profiles. However, there are some instances where it appears that when the "hop" was made from Twitter to LinkedIn, the entity resolution was incorrect. For example, two of the five discrepancies with the formal name attribute value have completely different formal names for the manual versus versus automated approach of obtaining attribute values.

It is also interesting to compare the results of automatically finding common attributes between the data sets. Figure 6.10 presents the comparison of the percentage of common attributes found automatically between the sets of "general"

Provenance Attribute	Number of Mismatches
Formal name	5
Location	4
Occupation	5
Education	1
Age	0
Employer	1
Political affiliation	0
Lobby affiliation	0
Special interests	0
Convictions	0
Citizenship	0
Ethnicity	0
Gender	0

Table 6.2: Mismatched attribute values for the political data set.

and "political" tweets used for manual analysis. Note that Figure 6.10 is similar to Figure 6.4 which presents the same comparison for the manual analysis results. The automated approach provides a similar amount of provenance data as was obtained during the manual analysis for common provenance attributes (i.e., the general attribute set) for both the general data set and the political data set.

Figure 6.11 compares the results of the manual analysis of the political data set with the automated analysis of the political data set. Although roughly the same number of common attributes, and some of the unique political attributes were found in the same amounts, there are some important differences. The *Gender* attribute was difficult to obtain automatically. Gender was only identified automatically for 1.9% (1 of 53) of the users in the political data set. During manual analysis, gender was identified for 35.9% of the users (19 of 53). During manual analysis, the author was able to distinguish gender based on profile photographs or through human natural language processing skills. The discrepancy between the number of instances of gender attributes identified between manual and automated

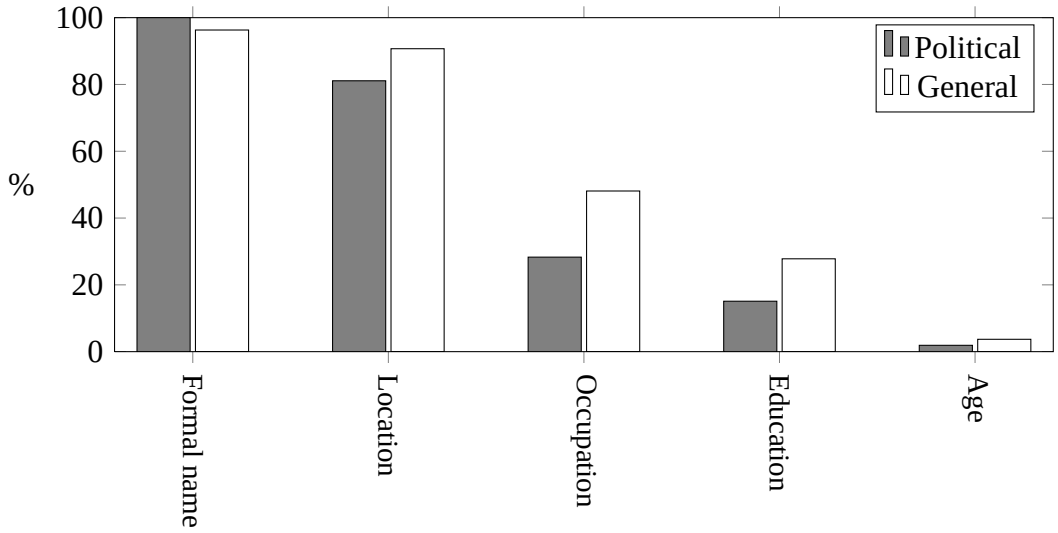

Figure 6.10: Comparison of percentage of common attributes found automatically between the sets of “general” and “political” tweets used for manual analysis.

analysis highlights important issues that need to be addressed for future provenance engine applications:

- Gender could be assigned based on a user’s formal name using the likelihood that a gender is associated with a specific name. However, this will not be completely reliable for all formal names. For example, the name Pat is used by both males and females in the United States.

- More complex text analysis techniques could be employed to automatically obtain occupation. The author implemented methods based on regular expressions to analyze text to obtain occupation. This was a simple approach that might be supplemented nicely with other approaches used to analyze text.

- Some attributes such as *Special Interests* and *Convictions* will also require more sophisticated approaches for automated analysis in order to obtain attribute values. Although it was relatively straight forward to map user profile fields to the *Special Interests* attribute, many user's do not publish data for all of the profile fields that are available. Additionally, political special interest are often different than the interests that were included in profile data and mapped to the *Special Interest* provenance attribute such as "travel, history, art, and fashion." However, in other cases the interests are clearly politically related such as "conservative politics." Thus, some of the some values returned automatically may not provide the exact insight a recipient is expecting when value is obtained for a particular provenance attribute. Nevertheless, it appears a reasonable mapping or closely related mapping of profile data to a provenance attribute would be better than not having a value for a particular provenance attribute as long as there is a reasonable degree of confidence that the attribute value is associated with the correct α.

- It is likely that some attributes a recipient may be interested in will be difficult to obtain because the data is simply not published or not accessible. It was observed that the *Convictions* provenance attribute can be difficult to ascertain in some circumstances, and based on the manual analysis of the data sets, are often not included in user profile data.

The graph shown in Figure 6.12 emphasizes the consistencies and inconsistencies between the manual and automated approach in the context of the political attribute data set. Consistency was measured simply as the difference between the percentage of attributes found during manual analysis versus automated analysis.

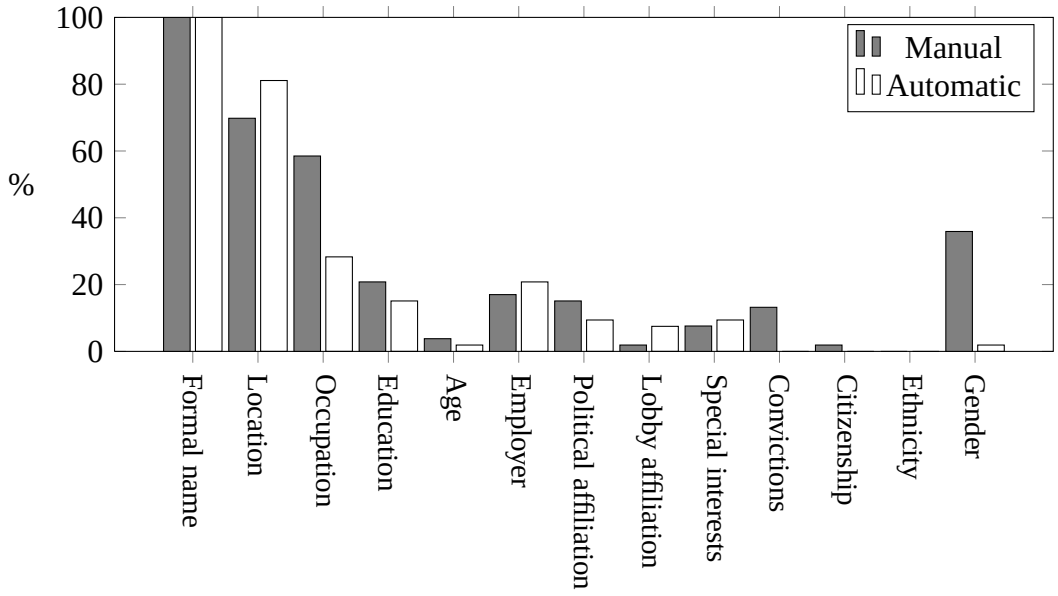

Figure 6.11: Comparison of percentage between manual and automatic search of political attributes.

Attributes with zero difference are consistent between manual and automated analysis. Inconsistent attribute values have a greater difference in the percentage of attribute values found during manual and automated analysis. For example, both the manual and the automated approach identified formal name attribute values for every α included in the political attribute data set. However, the manual approach yields very different amounts of attribute values for attributes including location, convictions, occupation, and gender.

In addition to the automated analysis of the political attribute data set, an automated analysis was performed with over 5,000 user names. Figure 6.13 exhibits how a larger sample compares with the political attribute data set that was used for manual analysis and automated analysis. The results suggests the process developed may be applied successfully more generally, and supports the need for

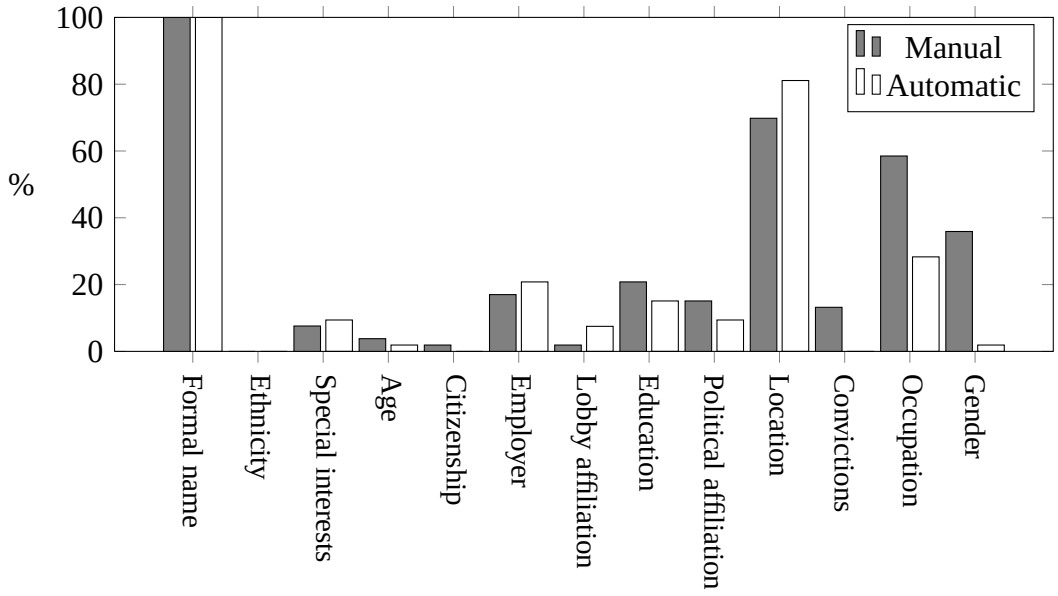

Figure 6.12: Consistency comparison between manual and automatic search of political attributes. Inconsistency (measured as the difference between the percentage of attribute values found) increases from left to right.

more sophisticated text analysis techniques.

Two approaches for obtaining provenance attributes were implemented in the Provenance Engine. First, the approach of "scraping" provenance attributes from social media sites without the benefit of APIs was used. Second, the approach of only using APIs was tried. A hybrid approach of scraping and employing APIs is most effective to obtain publicly available profile data. However, an individual Provenance Engine user of the future that is also a social media user will likely benefit from using their own social media sites credentials to access social media data that is not easily accessible publicly and not trivially extracted by scraping publicly available web pages. This research effort relied on publicly available data in accordance with Arizona State University Institutional Review Board guidance.

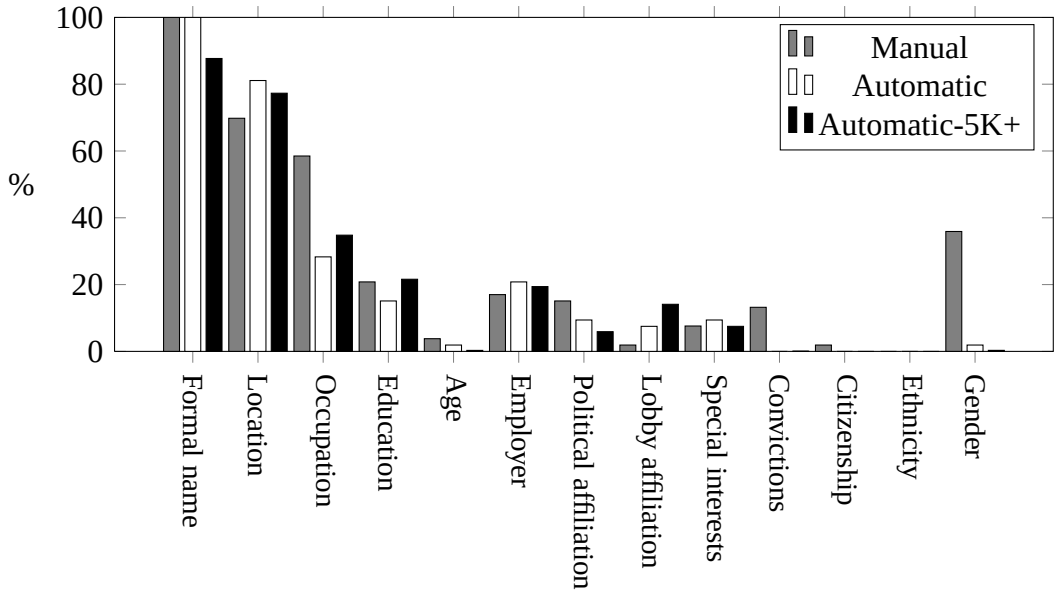

Figure 6.13: Comparison of percentage between manual and automatic search of political attributes to include over 5,000 α identifiers.

The manual and automated analysis provided some additional insights into approaches for finding and dealing with provenance attributes. In addition to the provenance attributes specified by the recipient, there are latent provenance attributes that might be useful during a provenance data search. *Latent provenance attributes* are attributes that are not explicitly specified by the recipient, but can be leveraged to identify explicit provenance attributes. For example a profile identification number that is unique to sites such as Twitter and Facebook might be useful for API calls. A friend set is another example of a latent provenance attribute that might be used to assist with entity resolution during an automated for provenance data. For example, the set of friends associated with α's Twitter profile could be saved as a provenance attribute and later compared with the friends associated with α's Facebook profile as a mechanism for dealing with duplicate formal names.

There is another latent attribute that would be meaningful to include in future work - time. The time a message was sent or posted could be compared to the time a profile was updated, and the time that the Provenance Engine collected provenance data associated with α. Without considering and presenting the provenance data with a frame of reference associated with time, the recipient is left to assume the provenance data is current. Time was not considered as a critical aspect during this phase of research because the focus of this effort was examining more fundamental questions about finding provenance data in social media including defining a general framework for the problem, defining and exploring what meaningful provenance data is for social media, and developing a criterion for evaluating the effectiveness of obtaining provenance data from social media.

Although it can be easily argued that the manual analysis produced better results, the Provenance Engine still produced usable provenance data and much faster[18] than is possible with manual analysis. This becomes particularly important when several disparate αs need be assessed to judge a provenance path.

6.4 Simple Provenance Paths

With a very basic automated means of searching for provenance attributes, the concept of a provenance path can be explored further. Twitter users have the option of tweeting a message that originated from another user. This is commonly known as a *retweet* and is abbreviated as "RT." It is also not uncommon for one user to retweet a message from another user that included a retweet from yet another user and so on. Retweets provide real-world examples of a provenance path.

[18]The Provenance Engine can return results for α in only seconds instead of the several minutes needed for manual analysis.

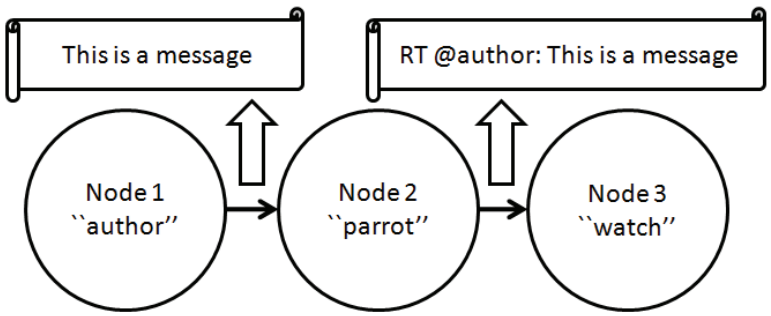

Figure 6.14: Example provenance path for hypothetical retweet.

A message that is retweeted only once provides a provenance path with three nodes including the originating user-author, the user that retweeted the message, and the recipient reader. It is possible that a tweet can contain more than two retweets but the maximum message size of 140 characters places practical limits on the maximum number of retweets that are discernable based solely on the text of the tweet. The abbreviation "RT" and the word "via" are common indicators that a Twitter user is retweeting a message (or a portion of a message) [63].[19]

It is useful to consider an example. A hypothetical tweet, "RT @author: This is a message", is sent by a user with the user name "parrot." A recipient user, with user name "watch," receives the tweet because "watch" follows "parrot." However, "watch" does not follow "author." In this case, "watch" may be familiar with "parrot" but not with "author." Provenance data associated with "author" and "parrot" might provide "watch" additional insight into the message. Figure 6.14 illustrates the provenance path associated with the example retweet messages and hypothetical users.

[19]Note that Modified Retweet (MRT) can also indicate a retweet, for a nice summary of retweet syntax see http://blog.tweetsmarter.com/retweeting/retweet-glossary-syntax-and-punctuation/ (accessed on October 19, 2011)

Variation	Example
RT used at the beginning of the message	RT @author This is a message
RT used at the end of a message	This is a message:RT @author
RT used with added text	This is exciting: RT @author This is a message
“via” inserted in message	This is a message via @author
“via” also used for a string of retweets	This is a message via @author @user1 @user2

Table 6.3: Example options to indicate a message has been retweeted.

Users may also modify the original message or add content to the message prior to retweeting. Another consideration when trying to construct a provenance path for a message that is retweeted is the various methods Twitter users employ to indicate a message was retweeted.

Twitter users have a variety of options to indicate that the message is retweeted. Table 6.3 lists some of the options commonly used to indicate that a tweet was retransmitted.

Figure 6.15 presents another screen shot from the Provenance Engine application. The Provenance Path window employs methods to evaluate a provenance path given a tweet that was retweeted by one or more users. The provenance path analysis is based on a few simple assumptions that must be made in order to address the free form text options that are used in practice to indicate that a message is retweeted (reference Table 6.3 for examples). The following assumptions are used as a basis for analyzing provenance paths in the context of Twitter:

1. All of the retweet annotations are included together in a single message.

2. “RT” precedes the user that is being referenced[20].

[20]Future application could also utilize “via” as an indicator for the ordered portion of the path. In cases which the “via” portion only contains the first and last users in a chain of retweets, it may be possible to look for overlaps in friend networks to estimate the provenance path.

3. The first retweet in the sequence of retweets is the original source. This also implies an assumption that the tweet contains all of the information about the provenance path[21].

4. The retweet text contains equivalent meaning to the original text.

5. The tweet contains all of the original text included in the message.

Additional quantitative analysis on a set of retweet messages is left for future work. However, the Provenance Engine application successfully demonstrated the concept and utility of finding provenance attributes for each node in a provenance path as well as structuring a provenance path given real-world social media data.

[21]The assumption that the tweet contains all of the information about the provenance path allows reasonable exploration of the provenance path concept bounded by the data available from Twitter. However, this ignores the possibility that a tweet might communicate or repeat information originating from another social media source.

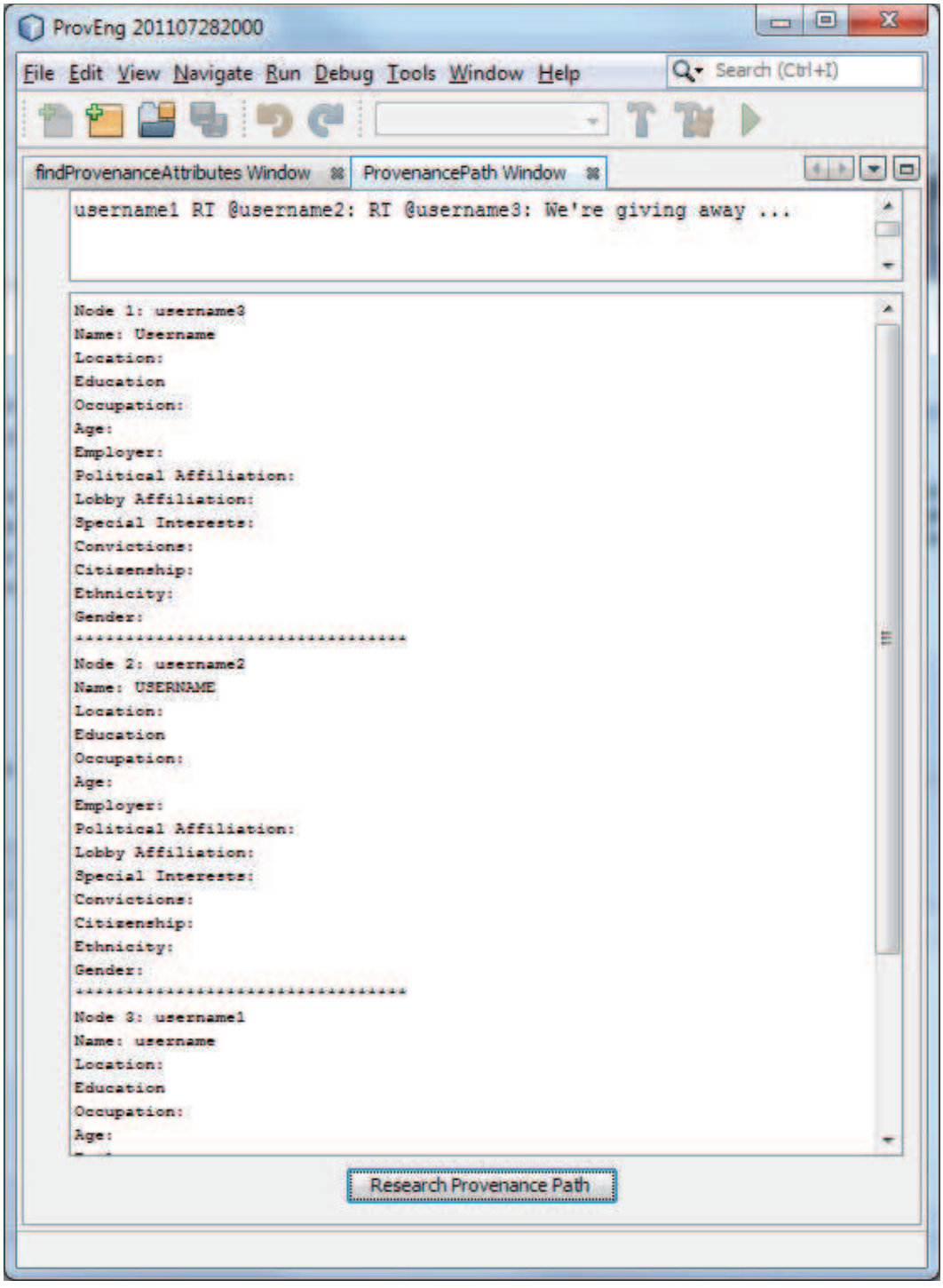

Figure 6.15: Provenance Path Window

Chapter 7

RELATED WORK

Provenance data is valuable in a variety of circumstances including database validation, tracing electronic workflows such as science simulations, and information products produced by a combination of distributed services. Agarwal and Liu included provenance as one of many research topics for the blogosphere in 2008 [3]. Simmhan and Gomadam briefly highlighted overarching provenance issues for the social web in 2010 [75]. However, provenance related research aimed at social media has received very little attention.

Moreau [59] identified six clusters of provenance literature including: "database, workflows, eScience, 'Provenance Challenge,' Open Provenance Model, Semantic Web, and electronic notebooks." Moreau's survey thoroughly covers the scope of efforts considering provenance from a web-based perspective. Although the survey provides over 450 references with an emphasis on data provenance, the survey does not identify a significant body of literature relating to provenance and: social media, social computing, or online social networks. In his words, "the bulk of the work on provenance has been undertaken by the database and workflow communities, specifically in the context of scientific applications."

Considering provenance from a data perspective aims to cover the provenance of a particular element of data such as a single value in a relational database. In the context of social media, the provenance of some specific piece of information could be broken down into pieces of data. False information about Chief Justice Roberts contained data about the person involved, his health status, future plans,

and even when the information allegedly would be made public. In Moreau's terms, this type of provenance would be described as provenance of a "data product" [59].

Considering the difference between provenance on the web and provenance in social media, it is also reasonable to reference Moreau's survey. Moreau defines provenance on the web as provenance relating to "data produced by computer systems, published and discovered on the web" [59]. From his perspective provenance in social media could almost be considered a subset of provenance on the web. However, there is an important distinction to make. In the social media environment, information is published by *people* using computer systems and is not "produced" by a computer system. Second, the distinction between data and provenance, as described previously, better represents the provenance problem space as it relates to social media.

Similar to the "complex workflows [4]" found in e-science (such as bioinformatics) and distributed service oriented applications, the flow of communication through the social media environment can also be complex. A message can be modified as it is passed from one user to another and can be distributed across disparate social media platforms. For applications areas with complex workflows such as bioinformatics, provenance research is characterized as data provenance. This is consistent with Moreau's terms where provenance would be described as provenance of a "data product" [59] and the discussion of "Mass Communication," referring to information published via the web, in [4].

Most approaches to collecting and managing provenance data for computational processes rely on some form of a provenance store. The provenance store

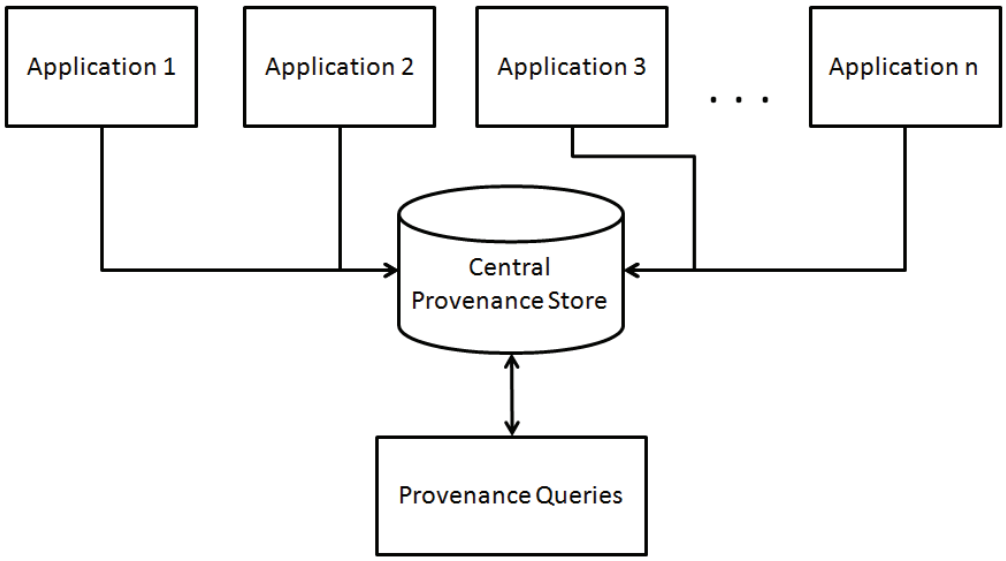

Figure 7.1: Central Provenance Store

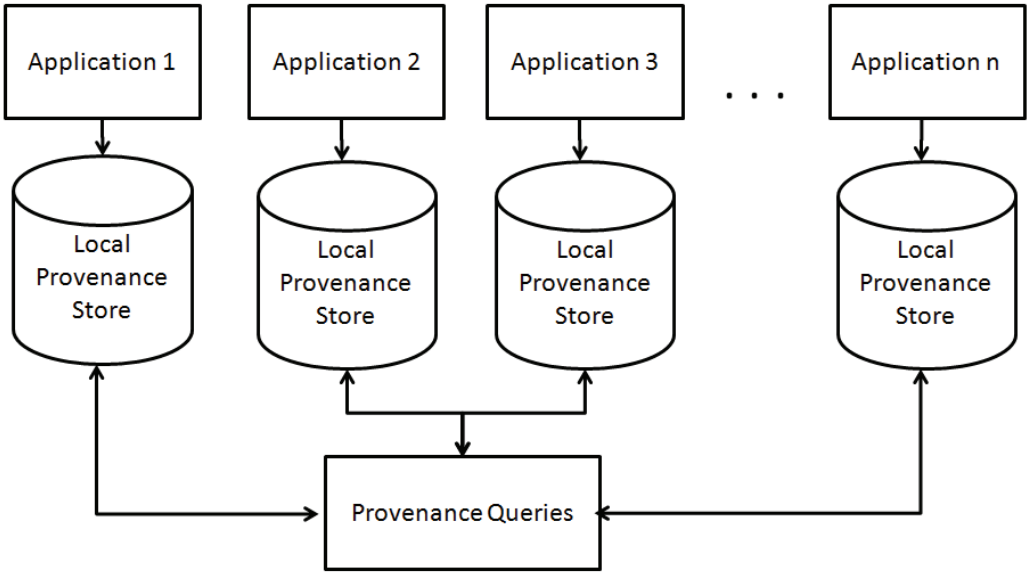

Figure 7.2: Distributed Provenance Store

can be centralized or distributed [36, 77]. With such approaches, provenance data is collected during processing. For example provenance data for a biology experiment based on a simulation may include variables such as databases used, parameters selected, and simulation components included in the in the experiment. In a central store implementation, each simulation component logs the provenance data to the central store so that the data can be queried later as shown in Figure 7.1. In a distributed store implementation, each simulation component logs and stores data locally that can be queried using an interface allowing a user to query all of the components, shown in Figure 7.2. Queries return provenance data that could be used to analyze problems in the simulation run, document progress, or even reused as initial settings to duplicate an experiment at a later time. With the popularity of cloud computing also comes new approaches for implementing a
provenance store [71].

7.1 Provenance Methods

An open provenance model (OPM) was developed to facilitate a "data exchange format" for provenance information [59, 60]. The OPM defines a provenance graph. The OPM graph is a directed graph representing "past computations" and the "Open Provenance Vision" requires individual system to collect their provenance data [59]. One tool, called *ourSpaces*, implements the OPM as part of a social connected Virtual Research Environment (VRE) to facilitate collaboration among scientist based on social links with a provenance logging capability for shared resources [68].

Provenance information for the web as a whole has been given more attention than provenance for social media. The World Wide Web Consortium (W3C)

Provenance Incubator Group recently published their final report [24]. The incubator group identified three flagship scenarios to highlight provenance issues. The use cases are News Aggregator, Disease Outbreak, and Business Contract. The News Aggregator scenario is the closest scenario related to the provenance data challenge in social media. The final report highlights 11 provenance issues. Six of these issues are pertinent to consider for finding and managing provenance data in social media:

- "Checking authority."
- "Recency of information."
- "Verification of original sources."
- "Conveying to an end user the derivation of a source of information."
- "Tracking user/reuse of content."
- "Scalable provenance management."

The W3C incubator group provides a list of provenance dimensions that could be applied to provenance data in social media. In particular, this work is related to the attribution dimension identified by the group. Attribution is characterized by the source and information about the source. The report [24] also documents an analysis of the state of the art for each flagship scenario including a gap analysis. The gap analysis for the News Aggregator scenario lists challenges also found in social media which motivate the approach of mining social media for provenance data. Specific items follow including a few notes about the implications for social media:

- "*No common format and application programmer's interface (API) to access and understand provenance information whether is explicitly indicated or implicitly determined.*" Social media sites do not provide provenance data today.

- "*Developers rarely include provenance management or publish provenance records.*"

- "*No widely accepted architecture solution to managing the scale of provenance records.*" Searching for provenance data "on-demand" and in near real-time helps to reduce the need to maintain large provenance stores.

- "*No existing mechanisms for tying identity to objects or provenance traces.*" The same challenge exists in social media which is the motivation for developing approaches to discover provenance paths [9].

- "*Incompleteness of provenance records and the potential for errors and inconsistencies in a widely distributed and open setting such as the web.*" This is also a challenge in the dynamic social media environment where information is published rapidly, by many people simultaneously, and with different view points.

As a predecessor to this work, the author defined information provenance for social media and formally defined the concept of a provenance path for social media in the prerequisite research proposal presented in November 2010, and at the 4th International Conference on Social Computing, Behavioral Modeling, and Prediction (SBP) [9].

Wang et al. defines information provenance and provenance paths for multi-hop networks in [91]. This definition of provenance metadata is restricted to creation time, owner, location history, and information items [90]. Their Provenance Based Trust Model requires each node in the multi-hop network to transmit provenance data and also implements a "Centralized Reputation Manager." Thus, provenance data is maintained by members of the network and transmitted with information (also referred to as statements in [90, 91]). Additionally, the trust model assumes the source is knowingly transmitted with the information across the network with provenance data.

Golbeck [37] connects provenance with individual trust leveraging Resource Description Framework (RDF) supported social networks. Simmhan etal.'s focused survey puts forward a taxonomy for provenance techniques [77]. Groth etal. [66] present a case for an interaction model as an overall representation for defining provenance for computational settings. The value and motivation for obtaining or providing provenance data for contemporary social media has been given only a small amount of attention. Golbeck's work relating provenance and trust to social networks relies on explicit information declared via the semantic web [37].

However, explicit information in the form of the semantic web is not widely available or implemented in contemporary social media services. Additionally, social media services do not currently collect provenance information or provide a subscription model. Unless an alternative solution is implemented, users are left to manually research provenance data.

Simmhan and Gomadam [75] state that "Provenance information for resources on the social web can be characterized" using three terms: resource prove-

nance, social provenance, and system provenance. Resource provenance pertains to tracking the creation of "social data artifacts" such as an image, documents, or other data element. Social provenance relates to the "social operators applied to the resource" such as a comments and relationships between individuals. Finally, system provenance addresses "passive tracking of the resources" such as download statistics.

Fox and Huang define what they call Knowledge Provenance (KP) [31, 32, 45, 46]. Their KP construct accounts for varying levels of certainty about information found in an enterprise (i.e. the web). Fox and Huang list a set of KP axioms dependant on documents annotated with KP meta-data that can be evaluated by a KP software agent capable of making a recommendation about trust. [45] addresses uncertainty in KP. [46] discusses trust in social networks and argues that "trust assessment" is an important component needed to make a "trust judgment." [31] states that among other things, social network users "need to define their trust relationships" to utilize a KP reasoned in an environment that has KP annotated documents.

Hasan et al. [43] defines a "provenance chain" and emphasizes the importance of "integrity and confidentiality" from a security vantage point. Deolalikar and Laffitte [28] investigate data mining (text mining specifically) as a basis for determining provenance given a set of documents.

Provenance can be characterized as a directed graph [28, 37, 59, 77]. Broadening the problem perspective beyond provenance attributes, considered in isolation, leads to applying the directed graph model in a new way to consider

information provenance in social media. Specifically, a provenance path can be assembled for each statement produced from the social media environment.

Determining the appropriate granularity of the provenance data to be collected is documented as an important consideration for designing and implementing provenance tracking systems [15, 22, 24, 77]. This is also a challenge for the social media environment. Appropriate granularity can be considered as the minimum amount of provenance data necessary to answer provenance queries in a useful and meaningful manner. In the case of a statement appearing in social media, the granularity is a result of the amount and types of provenance data attributes that could be associated with a particular statement.

Rowe [70] argues that "*Social Web users construct digital identity representations which mirror their real-world identities.*" Three tiers are used to define digital identity including: *My Identity, Shared Identity, and Abstracted Identity*. Specific provenance attribute values help to form the *My Identity*. *Shared Identity* and *Abstracted Identity* can be used to help deal with duplicated names and estimate likely provenance attribute values in some cases.

Dai et al. put forth an approach to evaluate data trustworthiness in [27]. Their approach addresses data similarity and data conflict and defines an Item Generation Path that assumes "every source provider and intermediate agent has a unique identifier." While interesting, it appears their path similarity approach would not scale well for social media. The unique identifier is also key for the provenance data approach for social media for which α represents the unique identifier for a social media node.

7.2 Provenance Metrics

Syed Ahsan and Abad Shah present a comprehensive list of twelve metrics for data provenance in [4]. The twelve metrics are: granularity, representation, format, scalability, data core-elements, completeness, accuracy, conformance, timeliness, accessibility, authority, and security. Some metrics are better defined than others and some metrics are more useful than others.

Granularity

The provenance granularity metric is loosely defined as assigning one point for each "metadata element" captured by the provenance scheme. Essentially, the granularity is the amount of detail the provenance scheme will capture about the data/information. Capturing more metadata elements as part of a provenance scheme results in a higher value for the granularity metric. While useful for making general comparisons amongst provenance applications, or tools used for collecting and management of provenance data for the same domain (for example, bioinformatics or business transactions), the granularity metric alone does not address important implementation limitations such as the maximize size of the provenance store. Depending on the data structures used and data elements collected, even a relatively small granularity score may still require a large amount of computational resources. For example, consider the difference between a provenance scheme that captures and records screen shots versus one that captures user name, date, time, and process ID. The latter scheme would have a higher score, but require less computational resources. However, granularity, in and of itself, is an important design consideration for provenance systems [15, 22, 77].

Representation

The purpose of the provenance representation metric is to quantify additional characteristics of the provenance data. This can be defined differently for various provenance applications. In addition to granularity, this metric would capture process information such as the workflow used (such as in the case of e-science). The metric is not well defined enough to be used generally, but could be developed to aid in comparing provenance systems designed for the same domain. Perhaps a better assessment of representation is the amount of space taken to store provenance data. Space is one of only two provenance metrics discussed in [95] and is a common topic in data provenance research [6, 22, 59, 77, 79]. The amount of space required for the provenance data is a simple, important, and practical metric. If the provenance data scheme is so large it cannot be used or implemented, it is worthless.

Format

Ahsan and Shah scale the provenance format metric from 1 to 10. However, they do not detail a process or guidelines for assigning an exact score. The goal is to quantify how searchable the provenance data is. The more machine readable the provenance data is, the higher the score. Even if there were specific guidelines for scoring available for this metric, the metric does not seem useful because it could be simplified by using a Boolean value set to True if the provenance data is represented using widely accepted standards such as eXtensible Markup Language (XML), Resource Description Framework (RDF), or Web Ontology Language (OWL), and False if it is not.

Scalability

The provenance scalability metric is meant to capture the cost of storing and accessing provenance data. Ahsan and Shah's explanation leave much to be desired in specifying specific methodologies for implementing this metric. However, this is an important factor to consider when judging the success of an information provenance approach, or implementation, and should be clarified for information provenance applications. If the information provenance solution cannot be scaled to provide usable provenance information in a reasonable time, the solution is worthless. For social media users, the information provenance solution should be readily accessible from any contemporary communication devices used to access social media information (i.e. personal computer and smart phone).

Core-elements

The provenance data core-elements metric is clearly defined. Core elements are "title, description, subject, data, and unique identifier." The values for this metric range from 1 to 5. The higher the value, the better the provenance data. This metric could be used to compare provenance applications and the quality of provenance data generically. The nice feature of this metric is that it provides some confidence that a system is providing the bare necessities of provenance data. However, this metric will not provide enough information to truly judge the success of information provenance research and would need to be supplemented with additional domain specific elements in order to be most useful.

Completeness

The provenance completeness metric "determines the extent to which provenance metadata gives an ideal representation of the data resource [4]." This metric, adapted from [62], proposes to distinguish between how much provenance metadata is collected by a provenance system or provenance scheme. It is useful for comparing provenance applications/schemes in the same domain. It could also be used to make general judgments about approaches to collecting provenance metadata. Provenance completeness is given as:

$Q_{completeness} = \frac{\sum_{i=1}^{N} P(i)}{N}$ where $P(i) = 1$, if the ith metadata has a non-null value, 0 otherwise.

There is a version of the completeness metric that allows weighting:

$Q_{Wcompleteness} = \frac{\sum_{i=1}^{N} \alpha_i P(i)}{\sum_{i=1}^{N} \alpha_i}$ where α_i is the relative importance of the ith data field.

The raw computation is the same as the information provenance availability function but the meaning and application are different. The provenance completeness metric is designed to compare the metadata used in provenance applications/schemes. The information provenance availability function is used to assess the number of provenance attribute values found during a search. This is an important distinction. For clarity, the provenance completeness metric would be applied to a set of provenance attributes and the value would be the same for any instance of provenance attribute values mapping to the same set of attributes. However, different provenance attribute values may yield very different provenance availability values even when mapped to the same provenance attribute set.

Accuracy

The provenance accuracy metric is meant to provide a quantifying sense of how well the provenance data enable the users to accurately recreate the object represented by the provenance data. This metric, adapted from [62], has clear application for workflows where the goal would be akin to 'given the provenance data, recreate the workflow.' The metric assigns a score of 1 for every 10% of the original key data elements that can be recreated given the provenance data. The maximum value is 100%. Ahsa and Shah convert these scores into relative distances by using:

$$Q_{accuracy} = \frac{1-\sqrt{\sum_{i=1}^{N} d(field_i)^2}}{\sum_{i=1}^{N} d(field_i)} \text{ Given that } \sum_{i=1}^{N} d(field_i) > 0.$$

The smaller the distance value, the better the provenance data can be used to recreate the data object. The accuracy metric does not seem like a reasonable approach because it could be vastly simplified to represent what is really important by using a Boolean value set to True if the provenance data can be used to recreate the workflow or data object and false if it cannot.

Conformance

The provenance conformance metric is proposed to quantify the information provided by the metadata. Adapted from [62], this metric attempts to quantify how much information the provenance data provides. It is described mathematically as:

$$Q_{conformance} = \frac{\sum_{i=1}^{N} Icontent(field_i)}{N}$$

Where N is the number of metadata fields and $Icontent(field_i)$ is the estimation of the amount of unique information contained in the field.

The provenance conformance metric is interesting in theory but practically not very useful for the proposed use of information provenance where the interest is not primarily to capture the provenance of a workflow or process rather than to capture the provenance data for a specific piece of information. Even viewed solely in the context of workflows, this metric does not significantly provide any additional value than using the provenance completeness metric or the provenance accuracy metric.

Timeliness

The provenance timeliness metric attempts to describe how current the provenance data is. Given highly dynamic information environments such as today's online social medial, having current provenance metadata about information is important. The Ahsan and Shah implementation of this metric combines the age of the document, the frequency of use, and the provenance accuracy metric as follows:

$$age = present\_year - publication\_year$$

$$frequency\_of\_use = \frac{times\_retrieved}{toal\_records\_retrieved} \text{ (over a period of a year)}$$

$$Q_{currency} = Q_{accuracy} \times age \times frequency\_of\_use$$

This definition is cumbersome and is not applicable to every provenance application or scheme. A simpler approach would be to define currency as the difference between the current time and the time at which the provenance data was obtained such as:

$$Q_currency = current\_time - time\_provenance\_data\_created/retrieved$$

Redefining the metric in this manner provides for more general use and

better addresses environments where currency might be evaluated frequently (i.e. daily or hourly) such as in the case of today's social media environment. The other aspect of timeliness important to consider for provenance systems is the time required to obtain provenance data of interest [6, 76, 95]. This aspect of timeliness is especially important when considering social media information. If the provenance system takes too long to provide provenance data, the provenance data may be superseded during retrieval or provide little or no value if provided too late to inform a decision that must be made.

Accessibility

The provenance accessibility metric is weakly defined but in simple terms it is a metric that would characterize how easy it is to find or access data resources in a repository. In general, this metric is not helpful and seems outside of the realm of interest. However, accessibility would be an excellent metric to include in order to evaluate approaches to obtaining information provenance in social media. In the social media context, some provenance information may simply be inaccessible due to privacy policy or other constraints. It is easy to envision an accessibility metric that would used to quantify answers to the question of 'given a set of provenance attributes, which ones are accessible and which ones are not.' For example, provenance attributes for a political domain may include name, location, occupation, birth date, ethnicity, etc. Birth date and ethnicity may not be accessible in some bounded social media environments but may be in others. An accessibility metric would nicely supplement to the results of the proposed information provenance availability function.

Authority

The provenance authority metric is described as a ”parameter that determines the trust a user places in the provenance information.” No mathematical definition is proposed and this metric is dependent on so much subjectivity that it is meaningless to apply to provenance data in social media.

Security

The provenance security metric is described but not clearly defined in [4]. However, this metric highlights an issue for provenance systems because under some circumstance it is important to ensure the provenance data itself is protected [25, 43, 59]. A clearly defined metric would help describe and allow comparison between approaches about how secure is the provenance data really is. For example, can the provenance data be modified, spoofed, and protected from unauthorized access? A list of security features implemented by a provenance system could be itemized relatively easy. The sum of the number of features implemented, although simple, would yield a much better defined and usable metric than described in [4].

Almost all of these metric concepts presented by Ahsa and Shah are useful for judging the success of provenance research. However, many of the metrics are not well defined enough or sufficiently standardized to yield measures that can be used generally. Additionally, the metrics do not address some important factors related to provenance data in social media. For example, including a succession metric to quantify whether or not there are breaks in provenance data could also be informative. Classic measures applied to information retrieval (i.e. precision and

recall), may provide additional value for a provenance methodology dependent on search techniques.

7.3 True or False Statements

The goal of finding provenance data about a particular statement is to provide a recipient addition context, and reveal any latent motivations about a particular statement published in social media. Sharing and publishing opinions is a popular use

of social media, and one motivation for revealing provenance data about an opinion statement is to better understand the backdrop for the statement.

Determining whether the statement is true or false is not a primary goal of finding provenance data in social media. However, provenance data certainly should be factored when a recipient questions the verity of a statement and there are some efforts solely dedicated to verifying whether not statements appearing in public (from a variety of media) are true or false.

FactCheck.org[1] employs people to research statements asserted as facts and validate whether the statement is true. FactCheck.org begins with a source of information (a political ad or particular candidate). FactCheck.org is currently processing "hundreds of questions each day" versus the ultimate vision for a Provenance Engine with the ability to process thousands of queries per hour. The reliance on human cognitive processing may provide accurate results but is unable to scale up to begin to address the large number of statements published in social media such as hundreds of millions of tweets published each day. In addition to FactCheck.org,

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

there are other sites and services dedicated to validating or refuting political statements [41] such as PolitiFact.com[2].

Snopes[3] boasts it is the "the definitive Internet reference source for urban legends, folklore, myths, rumors, and misinformation." Similar to FactCheck.org, articles published by the Snopes.com operating owners rely on human cognitive processing. Snopes efforts are primarily focussed on documenting the veracity of urban legends. For political opinions, Snopes works to investigate whether or not the attribution is correct[4]. While Snopes certainly can provide useful information,

the web site does not provide near real-time information about statements such as the provenance data desired for recipient social media users.

Researchers at the Indiana University Center for Complex Networks and Systems Research developed a system named Truthy [5] to track memes[6] in Twitter. The motivation for the Indiana researchers is to study "social epidemics" and to "detect political smears, astroturfing, misinformation, and other social pollution[7]." Unlike the approach to finding provenance data in social media, Truthy focuses on large numbers of tweets. This differs from the vision of finding provenance data in social media which provides a strategy for users to better assess even single statements published in social media.

[2]http://www.politifact.com/
[3]http://www.snopes.com/
[4]http://www.snopes.com/info/faq.asp, accessed on October 19, 2011.
[5]http://truthy.indiana.edu/
[6]Memes are cultural ideas or patterns or behavior.
[7]http://truthy.indiana.edu/about, accessed on October 19, 2011.

Castillo, Mendoza, and Poblete [17] investigated information credibility on Twitter and built a classifier aimed at discerning whether or not messages can be automatically categorized as credible or not. They identify four types of features used to categorize messages including seven user based features. Their work concludes that users "lack the clues that they have in the real world to assess the credibility of the information to which they are exposed." This conclusion supports the motivation for finding provenance data about a statement in social media such that a user will be better able to assess the statement.

Engineers for the popular Ushahidi[8] crisis map application are developing *Swift River*[9] to validate crowdsourced information. Although the proposed solution will likely have a human-in-the-loop to help with validation, engineers are working to implement algorithms that will help process invalid messages.

Research efforts have also focussed on identifying spam in Twitter [88] and investigating how Twitter is used in political activities [94]. Conover et al. [23] examined content and structure to build classifiers to distinguish political affiliation (liberal and conservative) for large number of Twitter users. Although, their approach might be leveraged to find particular provenance attribute values in the future, it is meant for groups versus individuals and is susceptible to errors when users include ambiguous text in statements such as sarcastic remarks.

Computer forensics literature covers a host of related topics that might be leveraged for future work on finding provenance data in social media. These topics

[8] http://www.ushahidi.com/
[9] http://swift.ushahidi.com/

include deception detection, identity theft on the web, face recognition, and other methods in which computational evidence is collected in a systematic matter [55].

Chapter 8

CONCLUSIONS

Social media applications have profoundly changed how people communicate. Consumers of traditional media did not face the same information provenance challenges that today's social media users face. Without provenance data, social media users can have a challenging time discerning latent meaning and bias that may be associated with a piece of information published in social media. Until provenance data is provided explicitly to recipients by social media applications, provenance data needs to be found independently. Leveraging social media to find provenance data about statements made in social media has the potential to address this gap. Provenance data can benefit social media users by exposing latent data upon which users can base judgements about statements that are published in social media.

In addition to the motivating cases previously discussed, this research has exciting implications for addressing contemporary issues facing users and decision makers such as: identifying the source of an online product review to reveal fake reviews, helping to implement a practical cyber genetics [8] capability, and determining the source when no author is evident.

This work presented a framework (provenance paths) for the problem of finding provenance data in social media, puts forth formal definitions, proposes metrics, suggests strategies for finding provenance, and highlights lessons learned in the development of a provenance search application. Additionally, this work establishes the basis that finding provenance data in social media is a viable approach that can be applied to contemporary, popular, social media. The initial results are

encouraging and there is a foundation for future research, but there are important research opportunities and challenges that are left to be addressed.

8.1 Research Opportunities

There are three areas that would benefit from additional research: addressing challenges related to finding and processing provenance attribute data, extending the investigation of provenance paths, and better understanding how time factors into provenance data in social media.

Provenance Attributes

It is clear that there is adequate data available in social media that can be used for provenance attributes. However, there are some important aspects of provenance attributes that could use further work that were revealed during this effort:

- How to ensure the attribute values are correct? In other words, how to validate whether or not the correct attribute values were returned? There are several items to consider, the most basic of which is entity resolution, that is, are the attribute values that are being collected attributable to the same individual? How can it be validated?
- Are the attributes adequately defined for a particular domain? Provenance attributes were defined as being subjectively defined by a particular recipient. However, there seems to be additional work that could be done to help inform a recipient about what attributes are useful. The general demographic attribute set defined in this work is a logical starting point. Determining how

to ensure the right attributes are defined for a particular domain is more theoretical and seems to be an excellent intersection for the social sciences.

- When unexpected attribute values are found, what does that mean? Is it valuable information? How should it be considered and presented to a recipient? For example, location attribute values such as “VEGAS BABY!!”, “Internet”, and “No, where are you?” do not convey the desired geographic location information but do convey sentiment, attitude, or feelings depending on how the attribute values are interpreted. Formal strategies for dealing with attribute values that are unexpected or do not exactly correspond to what was desired need to be developed.

  The manner in which information provenance availability, $r(V)$, is defined does not address the semantics of the provenance data about a statement. This becomes a problem if $r(V)$ is used as the sole criteria for validating the statement. $r(V)$ should be used to help assess how a statement should be considered in light of what is known about the statement itself. As an example, the ability to identify that any particular political party is associated with a statement versus not having any information about a statement enables the recipient to subjectively consider the statement given specific attribute values. $r(V)$ will be most useful to distinguish between similar statements or conflicting statements to indicate which statement might be preferred over the other. The first strategy for helping to deal with the semantics of provenance attributes is including the weighting mechanism in the definition of $r(V)$. Values returned from $r(V)$ also give an important indicator about whether or not any provenance data is available for a particular statement.

The current $r(V)$ and weighting scheme does not provide the type of automatic semantic discernment that would be most valuable to a recipient. One strategy for overcoming this problem directly might be to allow a recipient to define preferred values for provenance attributes that are most important to the recipient. Next, the preferred values could then be compared with the values that are identified during the attribute search using a metric. For example, preferred values could be contrasted to the actual values using edit distance. In such a case, preferred occupations such as *professor*, *lawyer*, and *surgeon*, would be contrasted with other occupations that are not preferred by a recipient such as *drug dealer*.

- How should attribute weights be determined for the provenance availability function? Attribute weights are subjectively determined by the recipient of social media data. A recipient can be an individual, group, or organization. These weights are subjective because: "provenance is context dependent," provenance data elements for one application area may not be valuable to another application area, the quality of provenance is determined from a user perspective [4], and there are multiple perspectives about provenance itself [59]. This is similar to considering trust subjectively when assessing trust across social networks from an individual perspective as noted in [37, 46]. Ahsan and Shah provide additional insight in [4]:

  > "Due to the heterogeneity and distribution of data resources, the usability of data resource for a particular domain depends upon the provenance information attached to the data resource. The content and amount of provenance information in turn is dependent on a

> number of factors such as the domain of use itself, its application within a particular domain and the mechanism of collecting provenance information."

Providing a mechanism for subjective weighting increases the utility of information provenance availability because it enables the computation to be used across domains under a variety of circumstances of interest to different recipients (individuals, groups, or organizations). This is an important ability for use in social media where it could be useful to consider provenance more abstractly (i.e., What ideology supports statement *S*?). In some cases it may be enough to know whether or not the idea being presented is adversarial or complementary toward the recipient. Understanding the nuances of a publication, position, or opinion, could provide an acceptable availability assessment to a recipient in order make a decision about the information under consideration.

For any single domain, the difference in defining weights can impact the usefulness of the computed information provenance availability. Attributes that are most indicative or instill the most confidence in the availability assessment should be weighted more. Attributes that are the most difficult to obtain (but most indicative) should be weighted greater. If the provenance weights are not chosen carefully, high information provenance availability scores will not be meaningful. For example, suppose a recipient is a group of realtors who receive a forwarded microblog message (a retweet) from a colleague that states "the park near Baker elementary is going to be replaced with a mall." The realtors are hypothetically interested in the following provenance

attributes: name, date, town, state, occupation, location, organization, place of employment, and political party. Equal weighting of the attributes would not capture that the political party probably is not the most important attribute for availability in this case. Incorrectly weighting the political party attribute could give a false sense of the value of the provenance information obtained.

- Additional work can be done to test and validate the metrics. Some of the metric concepts defined by Ahsan and Shah, and discussed in the chapter addressing related work, could prove beneficial for provenance data in social media. Specifically, timeliness, accessibility, authority, and security are loosely defined by Ahsan and Shah but the concepts would prove valuable if implemented for social media data. Lastly, the automated analysis was applied to over 5,000 α user names but additional large scale experiments including tens of thousands, or even millions, of users would better represent the hundreds of millions of social media users.

Provenance Paths

Approaches need to be designed to infer provenance data when the path is incomplete. Decision strategies need to be developed to help the recipient authenticate information provided through social media or determine whether or not the information itself can be corroborated via a separate provenance path including accepted social media nodes. In some cases, it may be enough to know whether or not the idea being presented is adversarial, complementary, or unique toward the recipient individual or group. Understanding the nuances of a publication, position, or

opinion, could lend itself to a level of confidence acceptable to a recipient by using only the portion of the provenance path that is available for analysis.

If the actual path is not completely known, it could be difficult to determine whether or not a discarded node contributed to or altered information presented to the recipient. Social media data could be leveraged to estimate likely paths. The nodes and links that are known to the recipient or consequently discovered can be exploited to provide a warning or calculate confidence values using probabilistic mechanisms to determine how the information might be considered.

A dynamic approach like a PE is needed because it is not practical for every recipient to store provenance data about every piece of information. Efficient storage of provenance data can be a challenge [15, 59] and provenance storage can be a limiting factor in an automated provenance system [77]. Additionally, the dynamic approach allows recipients to evaluate provenance paths representative of the dynamic social media environment. Over time, it is possible that the provenance path can change due to new information that becomes available or additional paths may be identified.

Additionally, research concerning how results could be mapped into previously defined structures, ontology definitions, and taxonomies suggested by other researchers such as OPM [59], KP [31], and provenance taxonomy [77] may provide useful insights.

Conducting research to better understand the factors in the social media environment that facilitate or hinder obtaining provenance data in the social media environment will also be important. Other interest include examining whether

or not provenance attribute values can be used as a basis for dealing with other aspects of the problem space. Specifically, might $r(V)$ be a reasonably effective objective function for greedily choosing a provenance path when multiple paths are evident for a specific piece of information? Can $r(V)$ be used to greedily choose the most likely predecessor node when an edge in a path is unclear? Availability might serve as a basis for examining characteristics in social media that could be important factors in estimating a provenance path such as distance between nodes and community structure.

Determining reasonable values of C for a particular domain will require additional effort. Any C value for a particular domain should take into account the information gain provided by a particular attribute as well as any recipient preferences for weighting the counts.

When the search for provenance data moves, or hops, from one social media application to another (such as from Twitter to LinkedIn), the hop should be chosen in a methodical manner. In this work, LinkedIn was chosen as the best site for the first hop based on the assumption that LinkedIn users are more professionally oriented as a user population. Facebook was chosen as the second hop because of its widespread popularity. However, there are additional social media sites that could also be considered, such as Google+[1].

This work also revealed that some social media sites are easier to access than others (for the purpose of searching for and obtaining provenance attribute values). Future work might include strategies for hopping based on the domain. The recipient's accounts may also be a determining factor because of increasing

[1]http://www.google.com/+/learnmore/, accessed on October 19, 2011

privacy and security restrictions. Additionally, access to some social media site APIs is facilitated by user (recipient) credentials.

Beyond the work to identify and assess provenance paths, there are additional questions related to provenance data in social media such as:

- How would provenance paths be valued from different recipients?
- Can provenance paths be identified and leveraged to help influence a group?
- In addition to trust, what other connections can be made between provenance and elements of social media?
- What are the implications for privacy?

Accounting for Time

There is also a temporal factor for provenance attributes and provenance paths that should be explored further. Are the attribute values found the most current attribute values? Did the attribute values change over time, and if so, when, and more importantly *why*? Has a path been used previously, and if so, was the path credible?

8.2 Future Work

The application developed for automated analysis encountered both expected and unexpected challenges. Entity resolution, improved text analysis for entity resolution (and attribute extraction), personalized versus public provenance attribute availability, and leveraging additional web based resources are areas that would likely benefit from additional development.

Entity resolution was an anticipated challenge. For this initial work, location was used to help reconcile duplicates. However, this simple approach will not scale up. Comparing friend networks to identify where there are similarities between a duplicate name on one social media site and another site may provide a useful mechanism for dealing with duplicates. Entity resolution work by other researchers [16, 48] might be leveraged to determine how to incorporate more sophisticated mechanisms into the application. During the manual analysis, in some cases, entities were matched across social media sites by using profile photos. An automated means of face recognition incorporated into an application would also assist in entity resolution. Facial recognition techniques are effective [47, 97] and are implemented commercially for a variety of appliations [56, 81].

The application developed for this effort implemented regular expressions as a simple mechanism for text analysis. More sophisticated text analysis means, such as those enumerated in [44], could be used in the future to assist with entity resolution and attribute extraction.

During the course of this research effort, social media security and privacy was a topic of discussion in many news stories and articles. As a result, social media web sites changed security posture and authentication schemes for APIs limiting the amount of data available. Social media users have easier access to social media data than is available publicly in some circumstances. This implies that accessing and finding provenance data in social media might be best approached from an individual recipient's perspective. Extending the application to leverage new APIs and security protocols should be investigated.

Lastly, search engine results proved beneficial for accessing publicly available profile data. Additional development work might better leverage search results and incorporate other internet sources such as personal web pages. Coupled with more sophisticated text processing, leveraging publicly available web data may yield additional attribute values and serve to validate provenance data.

BIBLIOGRAPHY

[1] R. Adams. John Roberts retirement rumour: A lesson in gossip and the internet. *The Guardian*, March 5, 2010.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734 – 749, june 2005.

[3] N. Agarwal and H. Liu. Blogosphere: research issues, tools, and applications. *SIGKDD Explor. Newsl.*, 10:18–31, May 2008.

[4] S. Ahsan and A. Shah. *Designing Software-Intensive Systems : Methods and Principles*, chapter Quality Metrics for Evaluating Data Provenance, pages 455–473. Information Science Reference (an imprint of IGI Global), 701 E. Chocalate Ave, Suite 200, Hershey, PA 17033, 2009.

[5] G. W. Allport and L. Postman. *The Psychology of Rumor*. Henry Holt and Company, New York, 1947.

[6] M. K. Anand, S. Bowers, T. McPhillips, and B. Ludäscher. Efficient provenance storage over nested data collections. In *EDBT '09: Proceedings of the 12th International Conference on Extending Database Technology*, pages 958–969, New York, NY, USA, 2009. ACM.

[7] E.-A. Baatarjav, S. Phithakkitnukoon, and R. Dantu. Group recommendation system for facebook. *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, pages 211–219, 2010.

[8] B. Bain. DARPA: Calling all cyber geneticists. Defense Systems, January 2010. Retrieved on October 17, 2011.

[9] G. Barbier and H. Liu. Information Provenance in Social Media. In J. Salerno, S. J. Yang, D. Nau, and S.-K. Chai, editors, *The 4th International Conference on Social Computing, Behavioral Modeling, and Prediction (SBP)*, volume 6589 of *Lecture Notes in Computer Science*, pages 276–283, College Park, MD, March 2011. Springer.

[10] G. Barbier, L. Tang, and H. Liu. Understanding online groups through social media. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4):330–338, 2011.

[11] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *Proceedings of the 14th international conference on World Wide Web*, pages 463–470. ACM, 2005.

[12] D. Benoit, D. Slauenwhite, and A. Trudel. A web census is possible. In *International Symposium on Applications and the Internet*, January 2006.

[13] M. Block. Tracing rumor of John Roberts' retirement. National Public Radio, March 2010. accessed on October 19, 2011.

[14] N. M. Bradburn, S. Sudman, and B. Wansink. *Asking Questions*. John Wiley & Sons Inc., 2004.

[15] U. Braun, S. Garfinkel, D. A. Holland, K.-K. Muniswamy-Reddy, and M. I. Seltzer. Issues in automatic provenance collection. In L. Moreau and I. Foster, editors, *Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 2006, Revised Selected Papers*, volume 4145, pages 171–183. Springer-Verlag Berlin Heidelberg, May 2006.

[16] D. G. Brizan and A. U. Tansel. A survey of entity resolution and record linkage methodologies. *Communications of the IIMA*, 6(3):41–50, 2006.

[17] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on Twitter. In *International World Wide Web Conference 2011, March 28April 1, 2011, Hyderabad, India*. Association for Computing Machinery, Inc., March 2011.

[18] A. C.C. *Social Network Data Analytics*. Springer, City, 2011.

[19] D. Centola. The spread of behavior in an online social network experiment. *Science*, 329:1194–1197, 3 September 2010.

[20] S.-K. Chai. Social computing: An opportunity for mathematical sociologists. *The Mathematical Sociologist*, 12(2), 2008-9.

[21] S.-K. Chai, J. J. Salerno, and P. L. Mabry, editors. *Advances in Social Computing*, Lecture Notes in Computer Science. Third International Conference on Social Computing, Behavorial Modeling, and Prediction, SBP 2010, Springer, March 2010.

[22] A. Chapman and H. Jagadish. Issues in building practical provenance systems. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 40(4):38–43, 2007.

[23] M. D. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Menczer. Predicting the political alignment of twitter users. In *3rd IEEE Conference on Social Computing*, October 2011.

[24] S. Coppens, D. Garijo, J. M. Gomez, P. Missier, J. Myers, S. Sahoo, and J. Zhao. Provenance XG Final Report. Final Report XGR-prov-20101214, World Wide Web Consortium (W3C), December 2010.

[25] B. Corcoran, N. Swamy, and M. Hicks. Combining provenance and security policies in a web-based document management system. In *On-line Proceedings of the Workshop on Principles of Provenance (PrOPr)*, Nov. 2007. http://homepages.inf.ed.ac.uk/jcheney/propr/, accessed on October 19, 2011.

[26] D. Crosby. Spread the news: Social media not all it's rumored to be. *The Courier News, A Chicago Sun-Times Publication*, May 6 2011. http://couriernews.suntimes.com/news/5207560-417/spread-the-news-social-media-not-all-that-its-rumored-to-be.html, accessed on 10 May 2011.

[27] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu. An approach to evaluate data trustworthiness based on data provenance. In W. Jonker and M. Petkovic, editors, *Secure Data Management*, volume 5159 of *Lecture Notes in Computer Science*, pages 82–98. Springer Berlin / Heidelberg, 2008.

[28] V. Deolalikar and H. Laffitte. Provenance as data mining: combining file system metadata with content analysis. In *TAPP'09: First workshop on Theory and practice of provenance*, pages 1–10, Berkeley, CA, USA, 2009. USENIX Association.

[29] S. Devlin. Why john roberts is still chief justice. *Missoula Editor*, March 6, 2010.

[30] A. Diana. Social media tops e-mail priority. *Information Week*, 2010. August 2, 2010.

[31] M. S. Fox and J. Huang. Knowledge provenance in enterprise information. *International Journal of Production Research*, 43(20):4471–4492, October 2005.

[32] M. S. Fox and J. Huang. An ontology for static knowledge provenance. In P. Bernus and M. Fox, editors, *Knowledge Sharing in the Integrated Enterprise*, volume 183 of *IFIP International Federation for Information Processing*, pages 203–213. Springer Boston, 2005.

[33] H. Gao, X. Wang, G. Barbier, and H. Liu. Promoting coordination for disaster relief - from crowdsourcing to coordination. In J. Salerno, S. J. Yang, D. Nau, and S.-K. Chai, editors, *The 4th Internatinoal Conference on Social Computing, Behavioral Modeling, and Prediction (SBP)*, volume 6589 of *Lecture*

Notes in Computer Science, pages 197–204, College Park, MD, March 2011. Springer.

[34] W. R. Garner. *Uncertainty and Structure as Psychological Concepts*. John Wiley and Son's, Inc., New York, 1962.

[35] L. Getoor and C. P. Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, 2005.

[36] B. Glavic and K. R. Dittrich. Data Provenance: A Categorization of Existing Approaches. In *BTW '07: 12. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web*, pages 227–241. Verlagshaus Mainz, Aachen, March 2007.

[37] J. Golbeck. Combining provenance with trust in social networks for semantic web content filtering. In L. Moreau and I. Foster, editors, *Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 2006, Revised Selected Papers*, volume 4145, pages 101–108. Springer-Verlag Berlin Heidelberg, May 2006.

[38] J. A. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, University of Maryland, College Park, MD, USA, 2005. Chair-Hendler, James.

[39] L. Grossman. Iran protests: Twitter, the medium of the movement. *Time*, June 17, 2009.

[40] L. Grossman. Twitter can predict stock market. Wired, October 2010. Accessed on October 19, 2011.

[41] P. J. Hane. Political fact-check web sites. Information Today, Inc., October 2007. Accessed on October 18, 2011.

[42] A. Harth, A. Polleres, and S. Decker. Towards a social provenance model for the web. In *2007 Workshop on Principles of Provenance (PrOPr)*, November 2007. Edinburgh, Scotland.

[43] R. Hasan, R. Sion, and M. Winslett. Preventing history forgery with secure provenance. *Trans. Storage*, 5(4):1–43, 2009.

[44] A. Hotho, A. Nürnberger, and G. Paaß. A brief survey of text mining. In *LDV Forum-GLDV Journal for Computational Linguistics and Language Technology*, volume 20, pages 19–62. Citeseer, 2005.

[45] J. Huang and M. S. Fox. Uncertainty in knowledge provenance. In C. Bussler, J. Davies, D. Fensel, and R. Studer, editors, *The Semantic Web: Research and Applications*, volume 3053 of *Lecture Notes in Computer Science*, pages 372–387. Springer Berlin / Heidelberg, 2004.

[46] J. Huang and M. S. Fox. Trust judgment in knowledge provenance. In *Proceedings. Sixteenth International Workshop on Database and Expert Systems Applications*, pages 524 –528, August 2005.

[47] R. Jenkins and A. M. Burton. 100 *Science*, 319(5862):435, January 2008.

[48] J. Jonas. Threat and fraud intelligence, las vegas style. *IEEE Security & Privacy*, 4(6):28 –34, nov.-dec. 2006.

[49] G. C. Kane, R. G. Fichman, J. Gallaugher, and J. Glasier. Community relations 2.0. *Harvard Business Review*, 87(11):45–50, November 2009.

[50] A. M. Kaplan and M. Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59–68, Jan 2009.

[51] I. King, J. Li, and K. T. Chan. A brief survey of computational approaches in social computing. In *IJCNN'09: Proceedings of the 2009 international joint conference on Neural Networks*, pages 2699–2706, Piscataway, NJ, USA, 2009. IEEE Press.

[52] S. Kumar, M.-A. Abbasi, H. Liu, and G. Barbier. Tweettracker: An analysis tool for humanitarian and disaster relief. In *Fifth International AAAI Conference on Weblogs and Social Media*, July 17-21 2011.

[53] H. Lauw, J. C. Shafer, R. Agrawal, and A. Ntoulas. Homophily in the digital world: A livejournal case study. *Internet Computing, IEEE*, 14(2):15 –23, march-april 2010.

[54] D. Lazer, A. Pentland, L. Adamic, S. Aral, A.-L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, and M. V. Alstyne. Computational social science. *Science*, 323:721–723, 2009.

[55] C.-T. Li. *Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions*. Information Science Reference, Hershey, PA, 2010.

[56] S. Li and D. Sarno. Advertisers start using facial recognition to tailor pitches. Los Angeles Times, August 2011. Accessed on October 17, 2011.

[57] M. McArdle. Anatomy of a fake quotation. *The Atlantic*, May 3 2011. http://www.theatlantic.com/national/archive/2011/05/anatomy-of-a-fake-quotation/238257/, accessed on 10 May 2011.

[58] M. Mendoza, B. Poblete, and C. Castillo. Twitter Under Crisis: Can we trust what we RT? In *1st Workshop on Social Media Analytics (SOMA10), Washington DC, USA*. ACM, July 25 2010.

[59] L. Moreau. The foundations for provenance on the web. *Foundations and Trends in Web Science*, 2:99–241, 2009.

[60] L. Moreau, J. Freire, J. Futrelle, R. McGrath, J. Myers, and P. Paulson. The open provenance model: An overview. In J. Freire, D. Koop, and L. Moreau, editors, *Provenance and Annotation of Data and Processes*, volume 5272 of *Lecture Notes in Computer Science*, pages 323–326. Springer Berlin / Heidelberg, 2008.

[61] B. Morrissey. Twitter sees sizable ad business. Adweek, July 2010. Accessed on October 19, 2011.

[62] X. Ochoa and E. Duval. Quality metrics for learning object metadata. In E. Pearson and P. Bohman, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006*, pages 1004–1011, Chesapeake, VA, June 2006. AACE.

[63] T. O'Reilly and S. Milstein. *The Twitter Book*. O'Reilly Media, Inc., Sebastopol, CA, 2009.

[64] B. Parr. Twitter surpasses 200 million tweets per day. Mashable, Inc., June 2011. Retreived on 18 Oct 2011.

[65] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. *Advances in Neural Information Processing Systems*, pages 1425–1432, 2003.

[66] S. M. Paul Groth and S. Munroe. Principles of high quality documentation for provenance: A philosophical discussion. In *Provenance and Annotation of Data: International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers*, volume 4145/2006 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006.

[67] E. Qualman. *Socialnomics*. Wiley, New York, 2009.

[68] R. Reid, E. Pignotti, P. Edwards, and A. Laing. ourspaces: linking provenance and social data in a virtual research environment. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 1285–1288, New York, NY, USA, 2010. ACM.

[69] C. Rovzar. Heres how the rumor that John Roberts is retiring may have gotten started. New York Magazine, March 2010. Accessed, March 4, 2010.

[70] M. Rowe. The credibility of digital identity information on the social web: a user study. In *Proceedings of the 4th workshop on Information credibility*, WICOW '10, pages 35–42, New York, NY, USA, 2010. ACM.

[71] S. Rozsnyai, A. Slominski, and Y. Doganata. Large-scale distributed storage system for business provenance. In *2011 IEEE International Conference on Cloud Computing (CLOUD)*, pages 516 –524, july 2011.

[72] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web, Raleigh, North Carolina, USA*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[73] D. Schuler. Social computing. *Commun. ACM*, 37(1):28–29, 1994.

[74] A. Shahid. Shirley Sherrod, ex-usda worker: White house forced me to resign over fabricated racial controversy. *New York Daily News*, July 20, 2010.

[75] Y. Simmhan and K. Gomadam. Social web-scale provenance in the cloud. In L. Moreau and I. Foster, editors, *Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2010, Troy, NY, USA, June 2010, Revised Selected Papers*, volume 6378. Springer-Verlag Berlin Heidelberg, June 2010.

[76] Y. Simmhan, B. Plale, D. Gannon, and S. Marru. Performance evaluation of the karma provenance framework for scientific workflows. In L. Moreau and I. Foster, editors, *Provenance and Annotation of Data*, volume 4145 of *Lecture Notes in Computer Science*, pages 222–236. Springer Berlin / Heidelberg, 2006.

[77] Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance techniques. Technical Report IUB-CS-TR618, Computer Science Department, Indiana University, Bloomington, IN 47405, 2005.

[78] B. Smith. Twitter to launch political advertising. Politico LLC, September 2011. Retrieved on October 19, 2011.

[79] D. Srivastava and Y. Velegrakis. Intensional associations between data and metadata. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 401–412, New York, NY, USA, 2007. ACM.

[80] J. Sterne. *Social Media Metrics*. The New Rules of Social Media. John Wiley & Sons Inc., 2010.

[81] A. Sternstein. FBI to launch nationwide facial recognition service. Nextgov, October 2011. Retrieved on October 17, 2011.

[82] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, New York, NY, USA, 2009. ACM.

[83] L. Tang and H. Liu. Toward collective behavior prediction via social dimension extraction. *Intelligent Systems, IEEE*, PP(99):1 –1, 2010.

[84] L. Tang, H. Liu, J. Zhang, N. Agarwal, and J. J. Salerno. Topic taxonomy adaptation for group profiling. *ACM Trans. Knowl. Discov. Data*, 1(4):1–28, January 2008.

[85] L. Tang, X. Wang, and H. Liu. Uncoverning groups via heterogeneous interaction analysis. In *Data Mining, 2009. ICDM '09. Ninth IEEE International Conference on*, pages 503 –512, 6-9 2009.

[86] J. Toon. I spy a red balloon: Georgia Tech team wins key insights and a second-place finish in DARPA network challenge. *Research Horizons*, Spring:30–31, 2010.

[87] G. Vaynerchuk. *Crush It!: Why Now Is the Time to Cash in on Your Passion*. HarperCollins, 10 East 53rd Street, New York, NY 10022, 1st edition, 2009.

[88] A. H. Wang. Don't follow me: Spam detection in twitter. In *Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT)*, pages 1 –10, july 2010.

[89] F.-Y. Wang, K. M. Carley, D. Zeng, and W. Mao. Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 22(2):79 –83, March-April 2007.

[90] X. Wang, G. Kannan, and P. Mohapatra. Collusion-resilient quality of information evaluation based on information provenance. In *2011 8th Annual IEEE*

Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pages 395 –403, June 2011.

[91] X. O. Wang, K. Govindan, and P. Mohapatra. Provenance-based information trustworthiness evaluation in multi-hop networks. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pages 1 –5, dec. 2010.

[92] M. Wesch. An Anthropological Introduction to YouTube. Presentation at the Library of Congress/Electronic, June 2008. Contributors include and The Digital Ethnography Working Group at Kansas State University; Accessed on 22 Mar 2010.

[93] B. Womack. Twitters ad revenue may triple to $150 million, emarketer says. Boomberg Bussinessweek, January 2011. Retrieved on October 19, 2011.

[94] A. Younus, M. A. Quresh, F. F. Asar, M. Azam, M. Saeed, and N. Touheed. What do the average twitterers say: A twitter model for public opinion analysis in the face of major political events. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 618 –623, july 2011.

[95] J. Zhang and H. V. Jagadish. Lost source provenance. In *EDBT '10: Proceedings of the 13th International Conference on Extending Database Technology*, pages 311–322, New York, NY, USA, 2010. ACM.

[96] D. Zhao and M. B. Rosson. How and why people twitter: the role that microblogging plays in informal communication at work. In *Proceedings of the ACM 2009 international conference on Supporting group work*, GROUP '09, pages 243–252, New York, NY, USA, 2009. ACM.

[97] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35:399–458, December 2003.

APPENDIX A

INSTITUTIONAL REVIEW BOARD (IRB) EXEMPTION LETTER

ASU Knowledge Enterprise Development

Office of Research Integrity and Assurance

To:	Huan Liu BYENG
From:	Mark Roosa, Chair Soc Beh IRB
Date:	02/18/2011
Committee Action:	**Exemption Granted**
IRB Action Date:	02/18/2011
IRB Protocol #:	1102006062
Study Title:	Provenance in Social Media

The above-referenced protocol is considered exempt after review by the Institutional Review Board pursuant to Federal regulations, 45 CFR Part 46.101(b)(4) .

This part of the federal regulations requires that the information be recorded by investigators in such a manner that subjects cannot be identified, directly or through identifiers linked to the subjects. It is necessary that the information obtained not be such that if disclosed outside the research, it could reasonably place the subjects at risk of criminal or civil liability, or be damaging to the subjects' financial standing, employability, or reputation.

You should retain a copy of this letter for your records.

APPENDIX B

RELATED TERMS

The terms in this section are included for reference, relation, and comparison to the definition of provenance data in social media presented in this work.

- Archiving "To compress one or more files and folders into a single file for backup or transport. Although archived files may remain on the same computer, the term implies data retention, and archived data are typically stored in a secondary location **for backup and historical purposes**"[2]

- Authentication "the process of confirming the correctness of **the claimed identity**"[3]

- Belief dynamics "changes in the beliefs of minds in the data of databases; database updating, theory change, theory revision, belief change, and **belief revision**"[4]

- Biographical identity "is comprised of **documented events which build up over time**, i.e. educational qualifications, marriage, employment history, mortgage accounts, bank accounts, utilities accounts etc."[5]

- Data aggregation "the ability to get a **more complete picture** of the information by analyzing several different types of records at once"[6]

- Data annotation "Researchers do more than produce and consume data: they **comment on it** and refer to it, and to the results of queries upon it"[7]

[2] http://encyclopedia2.thefreedictionary.com/Digital+archive
[3] http://www.sans.org/security-resources/glossary-of-terms/
[4] Hansson, Sven Ove, A Textbook of Belief Dynamics, 1999
[5] http://www.huntingvenus.com/ecart1.htm
[6] www.sans.org/security-resources/glossary-of-terms/
[7] http://www.nesc.ac.uk/esi/events/304/

- Data derivation "the process of **creating a data value from one or more contributing data values** through a data derivation algorithm" [8]

- Data pedigree - "the metadata which uniquely defines data and **provides a traceable path to its origin**"[9]

- Decision quality information enough **correct information** to inform correct decisions or serve as a basis for decisions

- Digital certificate "an electronic "credit card" that establishes **your credentials** when doing business or other transactions on the Web. It is issued by a certification authority. It contains your name, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages and digital signatures), and the digital signature of the certificate-issuing authority so that a **recipient can verify** that the certificate is real"[10]

- Digital signature "a hash of a message that **uniquely identifies the sender** of the message and proves the message hasn't changed since transmission"[11]

- Digital watermarking "process whereby arbitrary information is encoded into an image in such a way as to be imperceptible to image observers. has been proposed as a suitable tool for **identifying the source, creator,** owner, distributor, or authorized consumer of a document or an image"[12]

[8] http://www.geekinterview.com/kb/data-derivation.html
[9] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.138.1145
[10] www.sans.org/security-resources/glossary-of-terms/
[11] Ibid.
[12] Shih, Frank y., Digital Watermarking and Steganography, 2008

- Information attribution "assigning some quality or character to a person or thing"[13] (i.e. **assigning the source of information**)

- Information diffusion "anything that **propagates over a network**"[14]

- Integrity "the need to **ensure** that information has not been changed accidentally or deliberately, and that it is **accurate** and complete."[15]

- Non-repudiation "method by which the sender of data is provided with proof of delivery and the **recipient is assured of the sender's identity**, so that neither can later deny having processed the data."[16]

- Reliability "How can a user (or an automated agent) evaluate the reliability of digital materials? What **data** must be maintained **about the source** of the item and its creator to **facilitate a decision** to trust or not?"[17]

- Trust

  "reliance: **certainty** based on past experience"[18]

  "believe: be **confident** about something"[19]

  "the trait of **believing in the honesty and reliability of others**"[20]

  "determine which permissions and what actions other systems or **users can perform on remote machines**."[21]

[13]http://www.audioenglish.net/dictionary/attribution.htm
[14]http://www.cs.umd.edu/class/spring2008/cmsc828g/Slides/information-diffusion.pdf
[15]www.sans.org/security-resources/glossary-of-terms/
[16]http://www.tsl.state.tx.us/ld/pubs/compsecurity/glossary.html
[17]http://www.wtec.org/loyola/digilibs/02_12.htm
[18]http://wordnetweb.princeton.edu/perl/webwn?s=trust
[19]Ibid.
[20]Ibid.
[21]www.sans.org/security-resources/glossary-of-terms/

APPENDIX C

SOURCE CODE SAMPLES

```
package ProvenanceAttributes;

import java.io.Serializable;
/**
 *
 * @author gbarbier
 */
public class ProvenanceAttributes implements Serializable {

    public String formalName = "";
    public String TwitterUserNumber = "";
    public String LinkedInID = "";
    public String FacebookID = "";
    public String location = "";
    public String education = "";
    public String occupation = "";
    public String Age = "";
    public String URL =  "";
    public String originalSource = "";
    public String modifier = "";
    public String employer = "";
    public String politicalAffiliation = "";
    public String lobbyAffiliation = "";
    public String specialInterests = "";
    public String convictions = "";
    public String citizenship = "";
    public String ethnicity = "";
    public String gender = "";
    public String traditional_media_source = "";
    public String twitterBio = "";

    //Weigths to use for provenance availability function

    private int formalNameWeight = 1;
    private int locationWeight = 1;
    private int educationWeight = 1;
    private int occupationWeight = 1;
    private int AgeWeight = 1;
    private int employerWeight = 1;
    private int politicalAffiliationWeight = 1;
    private int lobbyAffiliationWeight = 1;
    private int specialInterestsWeight = 1;
    private int convictionsWeight = 1;
    private int citizenshipWeight = 1;
    private int ethnicityWeight = 1;
    private int genderWeight = 1;

    ////////////////////////////////////////////////////////////////
    // Method to compute proveanance availability value
    public double provenanceAvailability() {

        double WeightSum = formalNameWeight + locationWeight +
                educationWeight + occupationWeight + AgeWeight +
                employerWeight + politicalAffiliationWeight +
                lobbyAffiliationWeight + specialInterestsWeight +
                convictionsWeight + citizenshipWeight + ethnicityWeight +
                genderWeight;

        int x_formalName = 1;
        int x_location = 1;
        int x_education = 1;
```

```
    int x_occupation = 1;
    int x_Age = 1;
    int x_employer = 1;
    int x_politicalAffiliation = 1;
    int x_lobbyAffiliation = 1;
    int x_specialInterests = 1;
    int x_convictions = 1;
    int x_citizenship = 1;
    int x_ethnicity = 1;
    int x_gender = 1;

    if (formalName.isEmpty()) {x_formalName = 0;}
    if (location.isEmpty()) {x_location = 0;}
    if (education.isEmpty()) {x_education = 0;}
    if (Age.isEmpty()) {x_Age = 0;}
    if (employer.isEmpty()) {x_employer = 0;}
    if (politicalAffiliation.isEmpty()) {x_politicalAffiliation = 0;}
    if (lobbyAffiliation.isEmpty()) {x_lobbyAffiliation = 0;}
    if (specialInterests.isEmpty()) {x_specialInterests = 0;}
    if (convictions.isEmpty()) {x_convictions = 0;}
    if (citizenship.isEmpty()) {x_citizenship = 0;}
    if (ethnicity.isEmpty()) {x_ethnicity = 0;}
    if (formalName.isEmpty()) {x_formalName = 0;}
    if (gender.isEmpty()) {x_gender = 0;}

    return  ((x_formalName * formalNameWeight) +
            (x_location * locationWeight) +
            (x_education * educationWeight) +
            (x_occupation * occupationWeight) +
            (x_Age * AgeWeight) +
            (x_employer * employerWeight) +
            (x_politicalAffiliation * politicalAffiliationWeight) +
            (x_lobbyAffiliation * lobbyAffiliationWeight) +
            (x_specialInterests * specialInterestsWeight) +
            (x_convictions * convictionsWeight) +
            (x_citizenship * citizenshipWeight) +
            (x_ethnicity * ethnicityWeight) +
            (x_gender * genderWeight))
            /WeightSum;

}  // end method

////////////////////////////////////////////////////////////////
// Method to return all provenance attributes as a string
public String attributesToString() {
    return  "Name: " + formalName + "\n" +
            "Location: " + location + "\n" +
            "Education: " + education + "\n" +
            "Occupation: " + occupation + "\n" +
            "Age: " + Age + "\n" +
            // "URL: " + URL + "\n" +
            // "Orignal Source: " + originalSource + "\n" +
            // "Modifier: " + modifier + "\n" +
            "Employer: " + employer+ "\n" +
            "Political Affiliation: " + politicalAffiliation + "\n" +
            "Lobby Affiliation: " + lobbyAffiliation + "\n" +
            "Special Interests: " + specialInterests + "\n" +
            "Convictions: " + convictions + "\n" +
            "Citizenship: " + citizenship + "\n" +
            "Ethnicity: " + ethnicity + "\n" +
            "Gender: " + gender + "\n";
```

```
            // "Tradition media: " + traditional_media_source + "\n" +
            // "Bio: " + twitterBio + "\n"
}

////////////////////////////////////////////////////////////////
// Method to add new attributes to object only adds attribute
//  values to attributes that do not have previous values
public void addAttributeValues(ProvenanceAttributes newAttributes) {
    if(formalName.equals("") && !(newAttributes.formalName.equals(""))) {
        formalName = newAttributes.formalName;
    }
    if(location.equals("") && !(newAttributes.location.equals(""))) {
        location = newAttributes.location;
    }
    if(education.equals("") && !(newAttributes.education.equals(""))) {
        education = newAttributes.education;
    }
    if(occupation.equals("") && !(newAttributes.occupation.equals(""))) {
        occupation = newAttributes.occupation;
    }
    if(Age.equals("") && !(newAttributes.Age.equals(""))) {
        Age = newAttributes.Age;
    }
    if(URL.equals("") && !(newAttributes.URL.equals(""))) {
        URL = newAttributes.URL;
    }
    if(originalSource.equals("") &&
            !(newAttributes.originalSource.equals(""))) {
        originalSource = newAttributes.originalSource;
    }
    if(modifier.equals("") && !(newAttributes.modifier.equals(""))) {
        modifier = newAttributes.modifier;
    }
    if(employer.equals("") && !(newAttributes.employer.equals(""))) {
        employer = newAttributes.employer;
    }
    if(politicalAffiliation.equals("") &&
            !(newAttributes.politicalAffiliation.equals(""))) {
        politicalAffiliation = newAttributes.politicalAffiliation;
    }
    if(lobbyAffiliation.equals("") &&
            !(newAttributes.lobbyAffiliation.equals(""))) {
        lobbyAffiliation = newAttributes.lobbyAffiliation;
    }
    if(specialInterests.equals("") &&
            !(newAttributes.specialInterests.equals(""))) {
        specialInterests = newAttributes.specialInterests;
    }
    if(convictions.equals("") && !(newAttributes.convictions.equals(""))) {
        convictions = newAttributes.convictions;
    }
    if(citizenship.equals("") && !(newAttributes.citizenship.equals(""))) {
        citizenship = newAttributes.citizenship;
    }
    if(ethnicity.equals("") && !(newAttributes.ethnicity.equals(""))) {
        ethnicity = newAttributes.ethnicity;
    }
    if(gender.equals("") && !(newAttributes.gender.equals(""))) {
        gender = newAttributes.gender;
    }
    if(traditional_media_source.equals("") &&
```

```
                !(newAttributes.traditional_media_source.equals(""))) {
            traditional_media_source = newAttributes.traditional_media_source;
        }
        if(twitterBio.equals("") && !(newAttributes.twitterBio.equals(""))) {
            twitterBio = newAttributes.twitterBio;
        }
    } // end method
} // end class
```

```
package ProvenanceAttributes;

import MediaClients.LinkedInHTMLprocessing;
import MediaClients.TwitterHTMLprocessing;
import MediaClients.BingProcessing;
import MediaClients.FacebookAPIconnection;
import MediaClients.FacebookHTMLprocessing;
import MediaClients.LinkedInAPIconnection;
import MediaClients.TwitterAPIConnection;
import MediaClients.YahooProcessing;
import java.io.IOException;
import java.io.Serializable;
import java.io.UnsupportedEncodingException;
import java.net.MalformedURLException;
import java.util.ArrayList;
import java.util.HashSet;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPathExpressionException;
import oauth.signpost.exception.OAuthCommunicationException;
import oauth.signpost.exception.OAuthExpectationFailedException;
import oauth.signpost.exception.OAuthMessageSignerException;
import oauth.signpost.exception.OAuthNotAuthorizedException;
import org.json.JSONException;
import org.xml.sax.SAXException;

/**
 *
 * @author gbarbier
 */
public class Alpha implements Serializable {
    public String alphaUserName = "";
    public ProvenanceAttributes provAttr = new ProvenanceAttributes();

    public HashSet TwitterFriends = new HashSet();
    public HashSet LinkedInFriends = new HashSet();
    public HashSet FacebookFriends = new HashSet();
    public HashSet MySpaceFriends = new HashSet();

    private static int INFINITY = 10000;

    ////////////////////////////////////////////////////////////
    // constructor method
    public Alpha() {
        alphaUserName = "";
        provAttr.formalName = "";
        provAttr.TwitterUserNumber = "";
        provAttr.LinkedInID = "";
        provAttr.FacebookID = "";
        provAttr.location = "";
        provAttr.education = "";
        provAttr.occupation = "";
        provAttr.Age = "";
        provAttr.URL =  "";
        provAttr.originalSource = "";
        provAttr.modifier = "";
        provAttr.employer = "";
        provAttr.politicalAffiliation = "";
        provAttr.lobbyAffiliation = "";
        provAttr.specialInterests = "";
        provAttr.convictions = "";
        provAttr.citizenship = "";
```

```
    provAttr.ethnicity = "";
    provAttr.gender = "";
    provAttr.traditional_media_source = "";
    provAttr.twitterBio = "";

} //Alpha constructor

    ////////////////////////////////////////////////////////////////////
// Method to search all sources for provenance attributes
// Updates alpha's provenance attributes with the most probable values
public void recipientSearchProvAttr()
        throws IOException,
        XPathExpressionException, SAXException,
        ParserConfigurationException, MalformedURLException,
        OAuthMessageSignerException, OAuthExpectationFailedException,
        OAuthNotAuthorizedException, OAuthCommunicationException,
        UnsupportedEncodingException, JSONException {

    if (!alphaUserName.isEmpty()) {

        getTwitterAttributes();
        getLinkedInAttributes();
        scrapeFacebookAtrributes();
    }

}  // end method

public void getTwitterAttributes()
        throws MalformedURLException,
        IOException, XPathExpressionException,
        SAXException, ParserConfigurationException,
        OAuthMessageSignerException, OAuthExpectationFailedException,
        OAuthNotAuthorizedException, OAuthCommunicationException {

    TwitterAPIConnection user = new TwitterAPIConnection();

    provAttr = user.getTwitterAttributes(alphaUserName);
    TwitterFriends = user.getTwitterFriends(alphaUserName);

} // end method

public void getLinkedInAttributes()
        throws SAXException, ParserConfigurationException, IOException,
        OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
        XPathExpressionException {

       LinkedInAPIconnection user = new LinkedInAPIconnection();
       provAttr.addAttributeValues(user.IdentifyAttributes(
                    provAttr.formalName, provAttr.location));
       LinkedInFriends = user.getLinkedInFriends(provAttr.LinkedInID);

}  //end method

    public void getFacebookAttributes()
        throws SAXException, ParserConfigurationException, IOException,
        OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
        XPathExpressionException, UnsupportedEncodingException,
        MalformedURLException, JSONException {
```

```
        FacebookAPIconnection user = new FacebookAPIconnection();
        provAttr.addAttributeValues(user.IdentifyAttributes(
                    provAttr.formalName, provAttr.location));
        FacebookFriends = user.getFacebookFriends(provAttr.FacebookID);

}  //end method

////////////////////////////////////////////////////////////////////
// Method to search all sources for provenance attributes
// Updates alpha's provenance attributes with the most probable values
public void scrapePubProvAttr()
        throws UnsupportedEncodingException, IOException,
        XPathExpressionException, SAXException,
        ParserConfigurationException {

    if (!alphaUserName.isEmpty()) {

        scrapeTwitterAttributes();
        scrapeLinkedInAtrributes();
        scrapeFacebookAtrributes();

    }

}  // end method

////////////////////////////////////////////////////////
// Method that updates attribute values mined from alpha's
// Twitter profile page
public void scrapeTwitterAttributes() throws UnsupportedEncodingException {

    TwitterHTMLprocessing user = new TwitterHTMLprocessing();

    provAttr = user.ScrapeSingleProfile(
                "http://twitter.com/" + alphaUserName);
} // end method

////////////////////////////////////////////////////
// Method to update attribute values from alpha's
// LinkedIn public profile page
public void scrapeLinkedInAtrributes()
        throws IOException, XPathExpressionException,
        SAXException, ParserConfigurationException {

   LinkedInHTMLprocessing profile = new LinkedInHTMLprocessing();

   provAttr.addAttributeValues(
            profile.ScrapeLIAttributes(IDLinkedInPubProfile()));
}  // end method

////////////////////////////////////////////////////
// Method to identify the LinkedIn public profile page
// most likely associated with alpha
private String IDLinkedInPubProfile()
        throws XPathExpressionException, SAXException,
        ParserConfigurationException, IOException {

    int bestScore = INFINITY;
    int currentScore = 0;
    int bestIndex = 0;

    ArrayList<String> duplicateURLs = new ArrayList<String>();
```

```
    // get list of linked in profile pages
    // most closely associate with alpha's formal name
    // based on bing search query
    duplicateURLs = getAlphaLinkedInURLs();

    // select most likely profile based on the scoring function
    // the lower the profile score is, the more likely the
    // profile is associate with alpha
        for (int i = 0; i < duplicateURLs.size(); i++) {
            currentScore = LIprofileScore(duplicateURLs.get(i));
            if ( currentScore < bestScore){
                bestScore = currentScore;
                bestIndex = i;
            } // end if
        } // end for

    if(duplicateURLs.isEmpty()) {
        return null;
    }
    else {
        return duplicateURLs.get(bestIndex);
    }
}  // end method

////////////////////////////////////////////////
// Method to compute profile page score for a URL
// that is a candidate for alpha's profile page
private int LIprofileScore(String profileURL)
        throws MalformedURLException, IOException,
        XPathExpressionException {

    EditDistance function = new EditDistance();
    LinkedInHTMLprocessing profilePage = new LinkedInHTMLprocessing();

    // calculate socre
    String tempLocation = profilePage.getLinkedInLocation(profileURL);

    // if there a location value is missing return infinity
    if(tempLocation.isEmpty() || provAttr.location.isEmpty()){
        return INFINITY;
    }

    return function.computeEditDistance(provAttr.location, tempLocation);


} // end method

////////////////////////////////////////////////
// Method to return alphs's most likely LinkedIn
// profile pages
private ArrayList<String> getAlphaLinkedInURLs()
        throws XPathExpressionException, SAXException,
        ParserConfigurationException, IOException {

    BingProcessing resultBing = new BingProcessing();
    YahooProcessing resultYahoo = new YahooProcessing();

    ArrayList<String> emptyResult = new ArrayList<String>();

    // use Bing API but limit results using formal name
```

```
    if(provAttr.formalName.isEmpty()){
        return emptyResult;
    }

    return resultBing.getLIBingURLlist(
            resultBing.getBingQuery(provAttr.formalName + " LinkedIn"),
            provAttr.formalName.toLowerCase());
} // end method

    ////////////////////////////////////////////////////
// Method to update attribute values from alpha's
// LinkedIn public profile page
public void scrapeFacebookAtrributes()
        throws IOException, XPathExpressionException,
        SAXException, ParserConfigurationException {

   FacebookHTMLprocessing profile = new FacebookHTMLprocessing();

   provAttr.addAttributeValues(
           profile.ScrapeSingleFBProfile(IDFacebookPubProfile()));
}  // end method

    ////////////////////////////////////////////////////
// Method to identify the LinkedIn public profile page
// most likely associated with alpha
private String IDFacebookPubProfile()
        throws XPathExpressionException, SAXException,
        ParserConfigurationException, IOException {

    int bestScore = INFINITY;
    int currentScore = 0;
    int bestIndex = 0;

    ArrayList<String> duplicateURLs = new ArrayList<String>();

    // get list of linked in profile pages
    // most closely associate with alpha's formal name
    // based on bing search query
    duplicateURLs = getAlphaFacebookURLs();

    // select most likely profile based on the scoring function
    // the lower the profile score is, the more likely the
    // profile is associate with alpha
        for (int i = 0; i < duplicateURLs.size(); i++) {
            currentScore = FBprofileScore(duplicateURLs.get(i));
            if ( currentScore < bestScore){
                bestScore = currentScore;
                bestIndex = i;
            } // end if
        } // end for

    if(duplicateURLs.isEmpty()) {
        return null;
    }
    else {
        return duplicateURLs.get(bestIndex);
    }
}  // end method

//////////////////////////////////////////////////////
// Method to compute Facebok profile page score for a URL
```

```
    // that is a candidate for alpha's profile page
    private int FBprofileScore(String profileURL)
            throws MalformedURLException, IOException,
            XPathExpressionException, SAXException,


        EditDistance function = new EditDistance();
        FacebookHTMLprocessing profilePage = new FacebookHTMLprocessing();

        String tempLocation = profilePage.getFacebookLocation(profileURL);

        if(tempLocation.isEmpty() || provAttr.location.isEmpty()){
            return INFINITY;
        }

        // calculate socre
        return function.computeEditDistance(provAttr.location, tempLocation);

    } // end method

    ////////////////////////////////////////////////////
    // Method to return alphs's most likely LinkedIn
    // profile pages
    private ArrayList<String> getAlphaFacebookURLs()
            throws XPathExpressionException, SAXException,
            ParserConfigurationException, IOException {

        BingProcessing result = new BingProcessing();
        ArrayList<String> emptyResult = new ArrayList<String>();

        // use Bing API but limit results using formal name
        if(provAttr.formalName.isEmpty()){
            return emptyResult;
        }
        return result.getFBBingURLlist(
                result.getBingQuery(provAttr.formalName + " Facebook info"),
                provAttr.formalName.toLowerCase());
    } // end method
} // end class
```

```
package MediaClients;

import java.io.BufferedReader;
import java.io.InputStream;

import oauth.signpost.OAuth;
import oauth.signpost.OAuthConsumer;
import oauth.signpost.OAuthProvider;
import oauth.signpost.basic.DefaultOAuthConsumer;
import oauth.signpost.basic.DefaultOAuthProvider;
import oauth.signpost.exception.OAuthCommunicationException;
import oauth.signpost.exception.OAuthExpectationFailedException;
import oauth.signpost.exception.OAuthMessageSignerException;
import oauth.signpost.exception.OAuthNotAuthorizedException;
import org.openide.windows.IOProvider;
import org.openide.windows.InputOutput;
import ProvenanceAttributes.ProvenanceAttributes;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.StringReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.HashSet;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;
import org.openide.util.Exceptions;
import org.w3c.dom.Document;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;

public class TwitterAPIConnection {

    ////////////////////////////////////////////////////////////////////
    //
    //                          OAuth
    //
    ////////////////////////////////////////////////////////////////////
    String accesstoken;
    String accesssecret;
    static String CONSUMER_KEY = "";
    static String CONSUMER_SECRET = "";
    static String REQUEST_TOKEN_ENDPOINT_URL
          = "http://twitter.com/oauth/request_token";
    static String ACCESS_TOKEN_ENDPOINT_URL
          = "http://twitter.com/oauth/access_token";
    static String AUTHORIZE_WEBSITE_URL
          = "http://twitter.com/oauth/authorize";
    static String VERIFICATION_CODE = "";
    static String ACCESS_TOKEN = "";
    static String SECRET_TOKEN = "";
```

```
OAuthConsumer consumer
            = new DefaultOAuthConsumer(CONSUMER_KEY, CONSUMER_SECRET);
OAuthProvider provider
            = new DefaultOAuthProvider(REQUEST_TOKEN_ENDPOINT_URL,
                                       ACCESS_TOKEN_ENDPOINT_URL,
                                       AUTHORIZE_WEBSITE_URL);

public synchronized String getAccesssecret() {
    return accesssecret;
}

public synchronized void setAccesssecret(String accesssecret) {
    this.accesssecret = accesssecret;
}

public synchronized String getAccesstoken() {
    return accesstoken;
}

public synchronized void setAccesstoken(String accesstoken) {
    this.accesstoken = accesstoken;
}

public void getKey()
{
    try {
        InputOutput io = IOProvider.getDefault().getIO("OAuth getKey",
                                                       true);

       // we do not support callbacks, thus pass OOB

        String authUrl = provider.retrieveRequestToken(consumer,

        io.getOut().println("Now visit:\n" + authUrl +
                "\n... and grant this app authorization");
        io.getOut().println("Enter the PIN code in the text field " +
                            "and <Get Access Tokens>");

    } catch (OAuthNotAuthorizedException ex) {
             Exceptions.printStackTrace(ex);
    } catch (OAuthMessageSignerException ex) {
             Exceptions.printStackTrace(ex);
    } catch (OAuthExpectationFailedException ex) {
             Exceptions.printStackTrace(ex);
    } catch (OAuthCommunicationException ex) {
             Exceptions.printStackTrace(ex);
   } /*catch (IOException ex) {
             Exceptions.printStackTrace(ex);
    } */
}  // end method

public String getTokens(String code)
        throws OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException {

    provider.retrieveAccessToken(consumer, code);

    return consumer.getToken() + "\n" + consumer.getTokenSecret() + "\n";
}  // end method
```

```
////////////////////////////////////////////////////////////////////
//
//                      API calls
//
////////////////////////////////////////////////////////////////////

private String sendQuery(String APIString)
        throws OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
        IOException {

    BufferedReader bRead = null;
    boolean flag = true;
    OAuthConsumer tempconsumer =
            new DefaultOAuthConsumer(CONSUMER_KEY,CONSUMER_SECRET);
    tempconsumer.setTokenWithSecret(ACCESS_TOKEN, SECRET_TOKEN);

    URL url = new URL(APIString);
    HttpURLConnection request = (HttpURLConnection) url.openConnection();

    tempconsumer.sign(request);
    request.connect();

    if(request.getResponseCode()==400
        || request.getResponseCode() == 401
        || request.getResponseCode()==404)
                {
                    flag = false;
                }
    StringBuilder content = new StringBuilder();

    if(flag) {
        bRead = new BufferedReader(
                new InputStreamReader(
                (InputStream) request.getInputStream()));
        String temp = "";
        while((temp = bRead.readLine())!=null)
            {
                content.append(temp);
            }
    } // end if
    request.disconnect();
    return content.toString();
} // end method

public String SearchName(String screenName)
        throws OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
        IOException {

    String requestString =
            "http://api.twitter.com/1/users/lookup.xml?screen_name="

            + screenName;

    return sendQuery(requestString);
} // end method

public String SearchFriends(String screenName, String cursorValue)
        throws OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
```

```
        IOException {

    String requestString =
            "http://api.twitter.com/1/statuses/friends.xml?screen_name="
            + screenName + "&cursor=" + cursorValue;

    return sendQuery(requestString);
} // end method

////////////////////////////////////////////////////////////////
//
//              XML Processing
//
////////////////////////////////////////////////////////////////

// XPATH query to get first name from Twitter GET users/lookup API
private String xFormalName = "//user/name/text()";

// XPATH query to get first name from Twitter GET users/lookup API
private String xID = "//user/id/text()";

// XPATH query to get location from Twitter GET users/lookup API
private String xLocation = "//user/location/text()";

// XPATH query to get interests from Twitter GET users/lookup API
private String xDescription = "//user/description/text()";

// XPATH query to get friends from Twitter GET users/lookup API
private String xFriends = "//user/name/text()";

// XPATH query to get next cursor value from Twitter GET users/lookup API
private String xNextCursor = "//next_cursor/text()";

// regex to search for age in description
private static String agePattern
            = "[Ii] ?['aA]?[mM] ?[aA]? ([0-9][0-9]|100)[ ,.]";
private static String citizenPattern1
            = "[Ii] ?['aA]?[mM] ?[aA]? citizen of (.*?)[ ,.]";
private static String citizenPattern2
            = "[Ii] ?['aA]?[mM] [aA]? (.*?) citizen[ ,.]";

private static int FRIEND_LIMIT = 500;

private XPathFactory factory = XPathFactory.newInstance();
private XPath xpath = factory.newXPath();

private Document XMLStringToDom(String xmlSource)
        throws SAXException, ParserConfigurationException, IOException {

    DocumentBuilderFactory docfactory

    docfactory.setNamespaceAware(true);
    DocumentBuilder builder = docfactory.newDocumentBuilder();

    return builder.parse(new InputSource(new StringReader(xmlSource)));
}  // end method

////////////////////////////////////////////////////////////////
// Method for debugging XPATH results
private String ReturnNodeText(Node node) {
```

```
        String result = null;

        switch (node.getNodeType()) {
            case Node.ELEMENT_NODE:
                result = "<" + node.getNodeName();

                NamedNodeMap map = node.getAttributes();
                for (int i = 0; i < map.getLength(); i++) {
                result += " " + map.item(i).getNodeName() +
                        "=\"" + map.item(i).getNodeValue() + "\"";
                }
                result +=">\n";
                return result;
            case Node.ATTRIBUTE_NODE:
                return node.getNodeName() + "=\"" + node.getNodeValue() + "\"\n";
            case Node.TEXT_NODE:
                return "TEXT NODE "
                        + node.getNodeName() + " "
                        + node.getNodeValue() + "\n";
               // return "TEXT_NODE " + node.getTextContent() + "\n";
            case Node.CDATA_SECTION_NODE:
                return node.getNodeValue() + "\n";
            case Node.PROCESSING_INSTRUCTION_NODE:
                return node.getNodeValue() + "\n";
            case Node.DOCUMENT_NODE:
            case Node.DOCUMENT_FRAGMENT_NODE:
                return node.getNodeName() + "=" + node.getNodeValue() + "\n";
            }
        return result;
    }  // end method

      ////////////////////////////////////////////////////////////
      // Method to get user location
      public String getTwitterLocation(String userName)
            throws MalformedURLException, IOException, XPathExpressionException,
            SAXException, ParserConfigurationException,
            OAuthMessageSignerException, OAuthNotAuthorizedException,
            OAuthExpectationFailedException, OAuthCommunicationException  {

        String location = "";

        Document document = XMLStringToDom(SearchName(userName));

        if(document == null) {
            return location;
        }
            Object result = xpath.evaluate(xLocation,
                    document.getDocumentElement(),
                    XPathConstants.NODESET);

            NodeList nodes = (NodeList) result;
            for (int i = 0; i < nodes.getLength(); i++) {
          //      location += ReturnNodeText(nodes.item(i));
                location += nodes.item(i).getNodeValue();
            }

        return location;
      }  // end method

        public HashSet getTwitterFriends(String userName)
            throws MalformedURLException, IOException, XPathExpressionException,
```

```
        SAXException, ParserConfigurationException,
        OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException  {

    HashSet friends = new HashSet();
    int count = 0;

    String cursorValue = "-1";

    while ((cursorValue.equalsIgnoreCase("0") == false)
           && (count < FRIEND_LIMIT)) {

        String tempXML = SearchFriends(userName, cursorValue);
        if(tempXML.isEmpty()) {
            return friends;
        }

        Document document = XMLStringToDom(tempXML);

        if(document == null) {
            return friends;
        }
            Object result = xpath.evaluate(xFriends,
                    document.getDocumentElement(),
                    XPathConstants.NODESET);

            // increment count for the last 100 friends returnd
            // limit conserves twitter api limitations
            // implementing using the social graph to return 5000
            // ids in one call might be more efficient in the future

            count += 100;

            NodeList nodes = (NodeList) result;
            for (int i = 0; i < nodes.getLength(); i++) {
          //      friends += ReturnNodeText(nodes.item(i));
                friends.add(nodes.item(i).getNodeValue());
            }

            result = xpath.evaluate(xNextCursor,
                        document.getDocumentElement(),
                        XPathConstants.NODESET);

            nodes = (NodeList) result;
                for (int i = 0; i < nodes.getLength(); i++) {
              //      friends += ReturnNodeText(nodes.item(i));
                    cursorValue = nodes.item(i).getNodeValue();
                }

    } // end while

    return friends;
}  // end method

////////////////////////////////////////////////////////////////
// Method to obtain attributes from LinkedIn public profile page
// return provenance attribute object
public ProvenanceAttributes getTwitterAttributes(String userName)
        throws MalformedURLException, IOException, XPathExpressionException,
        SAXException, ParserConfigurationException,
        OAuthMessageSignerException, OAuthNotAuthorizedException,
```

```
    OAuthExpectationFailedException , OAuthCommunicationException  {

ProvenanceAttributes twitterAttributes = new ProvenanceAttributes();

String tempXML = SearchName(userName);
if(tempXML.isEmpty()){
    return twitterAttributes;
}
Document document = XMLStringToDom(SearchName(userName));

if(document == null) {
    return twitterAttributes;
}

// get ID
NodeList nodes = (NodeList) xpath.evaluate(xID,
        document.getDocumentElement(),
        XPathConstants.NODESET);

for (int i = 0; i < nodes.getLength(); i++) {
    twitterAttributes.TwitterUserNumber += nodes.item(i).getNodeValue();
}

// get formal name
nodes = (NodeList) xpath.evaluate(xFormalName,
        document.getDocumentElement(),
        XPathConstants.NODESET);

for (int i = 0; i < nodes.getLength(); i++) {
    twitterAttributes.formalName += nodes.item(i).getNodeValue();
}

// get location

nodes = (NodeList) xpath.evaluate(xLocation,
        document.getDocumentElement(),
        XPathConstants.NODESET);


for (int i = 0; i < nodes.getLength(); i++) {
    twitterAttributes.location += nodes.item(i).getNodeValue();
}

// get descritpion (twitter bio)
nodes = (NodeList) xpath.evaluate(xDescription,
                        document.getDocumentElement(),
                         XPathConstants.NODESET);

for (int i = 0; i < nodes.getLength(); i++) {
    twitterAttributes.twitterBio = nodes.item(i).getNodeValue();
}

// search description for age
Pattern patt = Pattern.compile(agePattern);
Matcher m = patt.matcher(twitterAttributes.twitterBio);
while (m.find()) {
    twitterAttributes.Age = m.group(1);
    }

// search for citizenship
```

```
        patt = Pattern.compile(citizenPattern1);
        m = patt.matcher(twitterAttributes.twitterBio);
        while (m.find()) {
            twitterAttributes.citizenship = m.group(1);
            }
        if(twitterAttributes.citizenship.isEmpty()) {
            patt = Pattern.compile(citizenPattern2);
            m = patt.matcher(twitterAttributes.twitterBio);
            while (m.find()) {
                twitterAttributes.citizenship = m.group(1);
                }
        }
        return twitterAttributes;
    }  // end method
}  // end class
```

```
package MediaClients;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import ProvenanceAttributes.Alpha;
import ProvenanceAttributes.ProvenanceAttributes;
import ProvenanceAttributes.ResearchUsers;
import org.openide.util.Exceptions;
import org.openide.windows.IOProvider;
import org.openide.windows.InputOutput;

/**
 *
 * @author gbarbier
 */
public class TwitterHTMLprocessing {

    private static String formalNamePattern
        = "<span class=\"fn\">(.*?)</span>";
    private static String locationPattern
        = "<span class=\"adr\">(.*?)</span>";
    private static String websitePattern
        = "<span class=\"label\">Web</span>\\s<a href=\"(.*?)\"";
    private static String bioPattern = "<span class=\"bio\">(.*?)</span>";
    private static String agePattern
        = "[Ii] ?['aA]?[mM] ?[aA]? ([0-9][0-9]|100)[ ,.]";
    private static String citizenPattern1
        = "[Ii] ?['aA]?[mM] ?[aA]? citizen of (.*?)[ ,.]";
    private static String citizenPattern2
        = "[Ii] ?['aA]?[mM] [aA]? (.*?) citizen[ ,.]";
    private static String occupationPattern
        = "[Ii]+ ?['aA]?[mM] ?[aA]? ([a-zA-Z]{2,})[ ,.]";

    public ProvenanceAttributes ScrapeSingleProfile(String profileURL)
    throws UnsupportedEncodingException {

        HTMLreader pageResult = new HTMLreader();

        ProvenanceAttributes TwitterAttributes = new ProvenanceAttributes();

        String twitterUserPage = pageResult.readHTMLFile(profileURL);

        Pattern patt = Pattern.compile(formalNamePattern);
        Matcher m = patt.matcher(twitterUserPage);
        while (m.find()) {
            TwitterAttributes.formalName = m.group(1);
            if(TwitterAttributes.formalName.contains("\t")){
                TwitterAttributes.formalName =
                   TwitterAttributes.formalName.replace("\t", " ");
            }
       }

        patt = Pattern.compile(locationPattern);
       m = patt.matcher(twitterUserPage);
        while (m.find()) {
            TwitterAttributes.location = m.group(1);
```

```
        }

    patt = Pattern.compile(websitePattern);
    m = patt.matcher(twitterUserPage);
    while (m.find()) {
        TwitterAttributes.URL = m.group(1);
      }

    patt = Pattern.compile(bioPattern);
    m = patt.matcher(twitterUserPage);
    while (m.find()) {
        TwitterAttributes.twitterBio = m.group(1);
        }

    patt = Pattern.compile(agePattern);
    m = patt.matcher(TwitterAttributes.twitterBio);
    while (m.find()) {
        TwitterAttributes.Age = m.group(1);
        }

    patt = Pattern.compile(citizenPattern1);
    m = patt.matcher(TwitterAttributes.twitterBio);
    while (m.find()) {
        TwitterAttributes.citizenship = m.group(1);
        }
    if(TwitterAttributes.citizenship.isEmpty()) {
        patt = Pattern.compile(citizenPattern2);
        m = patt.matcher(TwitterAttributes.twitterBio);
        while (m.find()) {
            TwitterAttributes.citizenship = m.group(1);
            }
    }
    return TwitterAttributes;
} // ScrapeSingleProfile

public String ScrapeReadFile(String dataFile) {
    BufferedReader inputStream = null;

    String profilesData = "";
    InputOutput io = IOProvider.getDefault().getIO("ReadFile", true);

    try {
        try {
            inputStream = new BufferedReader(new FileReader(dataFile));
        } catch (FileNotFoundException ex) {
            io.getErr().println("File not found");
            Exceptions.printStackTrace(ex);
        }
       String line ="";
       int counter = 0;

       try {
            while (((line = inputStream.readLine()) != null)
                    && (counter++ < ResearchUsers.numberToProcess)) {

                Alpha user = new Alpha();


                user.provAttr = ScrapeSingleProfile("http://twitter.com/"
                        + line);
```

```
                    profilesData = profilesData + counter + " " +
                                   "Name: " + user.provAttr.formalName + " " +
                                   "Age: " + user.provAttr.Age + " " +
                                   "Occupation: " +
                                   user.provAttr.occupation + " " +
                                   user.provAttr.location + " " +
                                   user.provAttr.URL + " " +
                                   user.provAttr.occupation + "\n";
                } // end while

            } catch (IOException ex) {
                io.getErr().println("Exception: " + ex);
                Exceptions.printStackTrace(ex);
            }
        } // end try
        finally {
            if (inputStream != null) {
                try {
                    inputStream.close();
                } catch (IOException ex) {
                    io.getErr().println("Exception: " + ex);
                    Exceptions.printStackTrace(ex);
                }
            }
        }
        io.getOut().close();
        io.getErr().close();
     return profilesData;
    } // end ReadFile
}  // class
```

```
package MediaClients;

import ProvenanceAttributes.EditDistance;
import java.io.BufferedReader;
import java.io.InputStream;

import oauth.signpost.OAuth;
import oauth.signpost.OAuthConsumer;
import oauth.signpost.OAuthProvider;
import oauth.signpost.basic.DefaultOAuthConsumer;
import oauth.signpost.basic.DefaultOAuthProvider;
import oauth.signpost.exception.OAuthCommunicationException;
import oauth.signpost.exception.OAuthExpectationFailedException;
import oauth.signpost.exception.OAuthMessageSignerException;
import oauth.signpost.exception.OAuthNotAuthorizedException;
import org.openide.windows.IOProvider;
import org.openide.windows.InputOutput;
import ProvenanceAttributes.ProvenanceAttributes;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.StringReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.HashSet;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;
import org.openide.util.Exceptions;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
/**
 *
 * @author gbarbier
 */
public class LinkedInAPIconnection {

    private static int INFINITY = 10000;

    ////////////////////////////////////////////////////////////////////
    //
    //                      API calls
    //
    ////////////////////////////////////////////////////////////////////

    String accesstoken;
    String accesssecret;
    static String CONSUMER_KEY =
            "";
    static String CONSUMER_SECRET =
            "";
    static String REQUEST_TOKEN_ENDPOINT_URL =
            "https://www.linkedin.com/uas/oauth/requestToken";
    static String ACCESS_TOKEN_ENDPOINT_URL =
            "https://www.linkedin.com/uas/oauth/accessToken";
```

```
static String AUTHORIZE_WEBSITE_URL =
        "https://www.linkedin.com/uas/oauth/authorize";
static String VERIFICATION_CODE = "";

static String ACCESS_TOKEN = "";
static String SECRET_TOKEN = "";

OAuthConsumer consumer
    = new DefaultOAuthConsumer(CONSUMER_KEY,CONSUMER_SECRET);
OAuthProvider provider
    = new DefaultOAuthProvider(REQUEST_TOKEN_ENDPOINT_URL,
                                            ACCESS_TOKEN_ENDPOINT_URL,
                                            AUTHORIZE_WEBSITE_URL);

public synchronized String getAccesssecret() {
    return accesssecret;
}

public synchronized void setAccesssecret(String accesssecret) {
    this.accesssecret = accesssecret;
}

public synchronized String getAccesstoken() {
    return accesstoken;
}

public synchronized void setAccesstoken(String accesstoken) {
    this.accesstoken = accesstoken;
}

public void getKey()
{
    try {
        InputOutput io = IOProvider.getDefault().getIO("OAuth getKey", true);

        String authUrl
            = provider.retrieveRequestToken(consumer, OAuth.OUT_OF_BAND);
        io.getOut().println("Now visit:\n" + authUrl +
                "\n... and grant this app authorization");
        io.getOut().println("Enter the PIN code in the text field " +
                            "and <Get Access Tokens>");

    } catch (OAuthNotAuthorizedException ex) {
             Exceptions.printStackTrace(ex);
    } catch (OAuthMessageSignerException ex) {
             Exceptions.printStackTrace(ex);
    } catch (OAuthExpectationFailedException ex) {
             Exceptions.printStackTrace(ex);
    } catch (OAuthCommunicationException ex) {
            Exceptions.printStackTrace(ex);
   }
}  // end method

public String getTokens(String code)
        throws OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException {

    provider.retrieveAccessToken(consumer, code);

    return consumer.getToken() + "\n" + consumer.getTokenSecret() + "\n";
}  // end method
```

```
private String SendQuery(String APIString)
        throws OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
        IOException {

    BufferedReader bRead = null;
    boolean flag = true;
    OAuthConsumer tempconsumer
       = new DefaultOAuthConsumer(CONSUMER_KEY,CONSUMER_SECRET);
    tempconsumer.setTokenWithSecret(ACCESS_TOKEN, SECRET_TOKEN);

    URL url = new URL(APIString);
    HttpURLConnection request = (HttpURLConnection) url.openConnection();

    tempconsumer.sign(request);
    request.connect();

    if(request.getResponseCode()==400
            || request.getResponseCode()==401
   //         || request.getResponseCode()==403
            ||request.getResponseCode()==404)
                {
                    flag = false;
                }
    StringBuilder content = new StringBuilder();

    if(flag) {
        bRead = new BufferedReader(new InputStreamReader((InputStream)
                request.getInputStream()));

        String temp = "";
        while((temp = bRead.readLine())!=null)
            {
                content.append(temp);
            }
    }  // end if

    request.disconnect();

    return content.toString();
} // end method

////////////////////////////////////////////////////////////////
// Method to search for linked in users via the linked in API
//
public String SearchName(String query)
        throws OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
        IOException {

    String[] tokens = query.split(" ");

    if(tokens.length == 0) {

        return null;

    }
    ArrayList<String> names = new ArrayList<String>();

    names.add(tokens[0]);
```

```
    if(tokens.length > 1) {
        names.add(tokens[1]);
    }
    else {
        names.add("");
    }

        String requestString = "http://api.linkedin.com/v1/people-search"
            + ":(people:(id,first-name,last-name,"
            + "headline,location,num-connections,"
            + "summary,associations,interests,"
            + "three-current-positions"
            + "),num-results)"
            + "?first-name=" + names.get(0)
            + "&last-name=" + names.get(1);

        return SendQuery(requestString);

} // end method

////////////////////////////////////////////////////////////////
// Method to get profile ID based on LinkedIn ID
public String getProfileURL(String profileID)
        throws OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
        IOException {

    String requestString = "http://api.linkedin.com/v1/people/id="
            + profileID + ":public";
        //    + ":(first-name,last-name,headline,location:(name,country))";

    return SendQuery(requestString);
} // end method

public String getPublicProfile(String profileURL)
       throws OAuthMessageSignerException, OAuthNotAuthorizedException,
       OAuthExpectationFailedException, OAuthCommunicationException,
       IOException {

       String requestString = "http://api.linkedin.com/v1/people/url="
            + profileURL;

    return SendQuery(requestString);
} // end method

////////////////////////////////////////////////////////////////////
//
//               XML Processing
//
////////////////////////////////////////////////////////////////////
// XPATH query to get first name from LinkedIn API
private static String xFirstName = "//person/first-name/text()";

// XPATH query to get first name from LinkedIn API
private static String xLastName = "//person/last-name/text()";

// XPATH query to get occupation from LinkedIn API
private static String xID = "//person/id/text()";

// XPATH query to get location from LinkedIn API
```

```
private static String xLocation= "//person/location/name/text()";

// XPATH query to get occupation from LinkedIn API
private static String xPublicProfileURL =
        "//person/site-public-profile-request/url/text()";

// XPATH query to get occupation from LinkedIn API
private static String xStandardProfileURL =
        "//person/site-standard-profile-request/url/text()";

// XPATH query to get occupation from LinkedIn API
private static String xOccupation = "//p[@class='title']/text()";

// XPATH query to get education from LinkedIn API
private static String xEducation
            = "//dd[@class='summary-education']/ul/li/text()";

// XPATH query to get interests from LinkedIn API
private static String xInterests = "//dd[@class='interests']/p/text()";

// XPATH query to get groups from LinkedIn API
private static String xGroups = "//dd[@class='pubgroups']/p/text()";

// XPATH query to get associaitons from LinkedIn API
private static String xAssociations
            = "//div[@class='group-data']/a/strong/text()";

private XPathFactory factory = XPathFactory.newInstance();
private XPath xpath = factory.newXPath();

//////////////////////////////////////////////////////////////////////
private Document XMLStringToDom(String xmlSource)
        throws SAXException, ParserConfigurationException, IOException {

    DocumentBuilderFactory docfactory = DocumentBuilderFactory.newInstance();
    docfactory.setNamespaceAware(true);
    DocumentBuilder builder = docfactory.newDocumentBuilder();
    if(xmlSource.isEmpty()) {
        return null;
    }
    return builder.parse(new InputSource(new StringReader(xmlSource)));
}  // end method

//////////////////////////////////////////////////////////////////
// Method to select most probable LinkedIn profile and return provenance
// attributes
//
public ProvenanceAttributes IdentifyAttributes(
        String formalName, String location)
        throws SAXException, ParserConfigurationException,
        IOException, OAuthMessageSignerException, OAuthNotAuthorizedException,
        OAuthExpectationFailedException, OAuthCommunicationException,
        XPathExpressionException{

    ProvenanceAttributes tempAttr = new ProvenanceAttributes();

    int bestScore = INFINITY;
    int currentScore = 0;
    int bestIndex = 0;
    String profileURL = "";
```

```
// get list of linked in users
// most closely associate with alpha's formal name

Document document = XMLStringToDom(SearchName(formalName));

if(document == null) {
    return tempAttr;
}

// select most likely profile based on the scoring function
// the lower the profile score is, the more likely the
// profile is associate with alpha

NodeList nodes = (NodeList) xpath.evaluate(xLocation,
        document.getDocumentElement(),
        XPathConstants.NODESET);

if(nodes.getLength() == 0) {
    return tempAttr;
}

for (int i = 0; i < nodes.getLength(); i++) {
    String tempLocation = nodes.item(i).getNodeValue();
    tempLocation = tempLocation.replace(" Area", "");
    tempLocation = tempLocation.replace("Greater ", "");
    currentScore = LocationScore(location, tempLocation);
        if ( currentScore < bestScore){
            bestScore = currentScore;
            bestIndex = i;
        } // end if
}  // end for

// get attributes

// get ID
nodes = (NodeList) xpath.evaluate(xID,
        document.getDocumentElement(),
        XPathConstants.NODESET);

if(nodes.getLength() == 0) {
    return tempAttr;
}

tempAttr.LinkedInID += nodes.item(bestIndex).getNodeValue();

// using id get public profile url

String tempXML = getProfileURL(tempAttr.LinkedInID);
if(tempXML.isEmpty()) {
    return tempAttr;
}
document = XMLStringToDom(tempXML);

nodes = (NodeList) xpath.evaluate(xPublicProfileURL,
        document.getDocumentElement(),
        XPathConstants.NODESET);

if(nodes.getLength() == 0) {
    return tempAttr;
}
```

```
        for (int i = 0; i < nodes.getLength(); i++) {
            profileURL = nodes.item(i).getNodeValue();
        }

        // scrape informaiton from public profile

        LinkedInHTMLprocessing scraper = new LinkedInHTMLprocessing();

        tempAttr = scraper.ScrapeLIAttributes(profileURL);

        return tempAttr;
    }  // end method

    ////////////////////////////////////////////////
    // Method to compute profile score for a LinkedIn user
    // that is a candidate for alpha's user page based on location
    // This method should be combined with other metrics to yield a score
    private int LocationScore(String twitterLocation, String LinkedInLocation)

    {
        EditDistance function = new EditDistance();
        // if there si no location return infinity
        if(LinkedInLocation.isEmpty()){
            return INFINITY;
        }
        return function.computeEditDistance(twitterLocation, LinkedInLocation);
    } // end method
} // end class
```

```
package MediaClients;

import ProvenanceAttributes.ProvenanceAttributes;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;
import org.openide.util.Exceptions;
import org.w3c.dom.Document;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.tidy.Tidy;

/**
 * code adapted from jwei512's public example
 * http://thinkandroid.wordpress.com/2010/01/05/using-xpath-and-html-
 * cleaner-to-parse-html-xml/
 * @author gbarbier
 */
public class LinkedInHTMLprocessing {

    // XPATH query to get first name from LinkedIn public profile page
    private static String xFirstName = "//span[@class='given-name']/text()";

    // XPATH query to get first name from LinkedIn public profile page
    private static String xLastName = "//span[@class='family-name']/text()";

    // XPATH query to get occupation from LinkedIn public profile page
    private static String xOccupation = "//p[@class='title']/text()";

    // XPATH query to get location from LinkedIn public profile page
    private static String xLocation = "//dd[@class='locality']/text()";

    // XPATH query to get education from LinkedIn public profile page
    private static String xEducation
        = "//dd[@class='summary-education']/ul/li/text()";

    // XPATH query to get interests from LinkedIn public profile page
    private static String xInterests = "//dd[@class='interests']/p/text()";

    // XPATH query to get groups from LinkedIn public profile page
    private static String xGroups = "//dd[@class='pubgroups']/p/text()";

    // XPATH query to get associaitons from LinkedIn public profile page
    private static String xAssociations
            = "//div[@class='group-data']/a/strong/text()";

    private XPathFactory factory = XPathFactory.newInstance();
    private XPath xpath = factory.newXPath();


    ////////////////////////////////////////////////////
    // Method to convert URL to DOM for XPATH processing
    private Document URLToDom(String htmlsource)
```

```
        throws MalformedURLException, IOException {

    Tidy tidy = new Tidy();
    URLConnection conn = null;
    URL url = null;
    InputStreamReader in = null;

     try
    {
        url = new URL(htmlsource);
    }
    catch ( MalformedURLException ex)
    {
         Exceptions.printStackTrace(ex);
         return null;
    }
    try
    {
        conn = url.openConnection();
        HttpURLConnection huc = (HttpURLConnection) conn;
        huc.setRequestProperty("User-Agent",
        "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0) Gecko/20100101 Firefox/4.0");
        if(huc.getResponseCode()==400||huc.getResponseCode()==404)
              {
                  return null;
              }
    }
    catch (IOException ex) {
        Exceptions.printStackTrace(ex);
        return null;
    }

    in = new InputStreamReader(conn.getInputStream());

    tidy.setQuiet(true);
    tidy.setShowWarnings(false);

    return tidy.parseDOM(in, null);

}  // end method

////////////////////////////////////////////////////////////////
// Method for debugging XPATH results
private String getNodeText(Node node) {

    String result = null;

    switch (node.getNodeType()) {
        case Node.ELEMENT_NODE:
            result = "<" + node.getNodeName();

            NamedNodeMap map = node.getAttributes();
            for (int i = 0; i < map.getLength(); i++) {
            result += " " + map.item(i).getNodeName() +
                    "=\"" + map.item(i).getNodeValue() + "\"";
            }
            result +=">\n";
            return result;
        case Node.ATTRIBUTE_NODE:
            return node.getNodeName() + "=\"" + node.getNodeValue() + "\"\n";
        case Node.TEXT_NODE:
```

```
                return "TEXT NODE "
                        + node.getNodeName() + " "
                        + node.getNodeValue() + "\n";
               // return "TEXT_NODE " + node.getTextContent() + "\n";
            case Node.CDATA_SECTION_NODE:
                return node.getNodeValue() + "\n";
            case Node.PROCESSING_INSTRUCTION_NODE:
                return node.getNodeValue() + "\n";
            case Node.DOCUMENT_NODE:
            case Node.DOCUMENT_FRAGMENT_NODE:
                return node.getNodeName() + "=" + node.getNodeValue() + "\n";
            }
        return result;
    }  // end method

    ////////////////////////////////////////////////////////////////
    // Method to get lcation from a LinkedIn public profile page
    public String getLinkedInLocation(String profilepage)
            throws MalformedURLException, IOException, XPathExpressionException  {
        String location = "";

        Document document = URLToDom(profilepage);
        if(document == null) {
            return location;
        }
        Object result;
        try {
        result = xpath.evaluate(xLocation,
                    document.getDocumentElement(),
                    XPathConstants.NODESET);
        }
        catch (XPathExpressionException ex) {
            return location;
        }
        catch (ArrayIndexOutOfBoundsException ex){
            return location;
        }
        NodeList nodes = (NodeList) result;

        if(nodes.getLength() == 0) {
            return location;
        }

        for (int i = 0; i < nodes.getLength(); i++) {
             location += nodes.item(i).getNodeValue();
        }
       location = location.replace(" Area", "");
       location = location.replace("Greater ", "");

       return location;
    }  // end method

    ////////////////////////////////////////////////////////////////
    // Method to scrape attributes from LinkedIn public profile page
    // return provenance attribute object
    public ProvenanceAttributes ScrapeLIAttributes(String profilepage)
            throws MalformedURLException, IOException, XPathExpressionException  {

        ProvenanceAttributes LinkedInAttributes = new ProvenanceAttributes();

        // if no URL associated with alpha return empty attribute set
```

```
 if(profilepage == null) {
     return LinkedInAttributes;
 }

 Document document = URLToDom(profilepage);

 if(document == null) {
     return LinkedInAttributes;
 }

NodeList nodes;

try {
// get first name
nodes = (NodeList) xpath.evaluate(xFirstName,
         document.getDocumentElement(),
         XPathConstants.NODESET);

 // get special interests
 for (int i = 0; i < nodes.getLength(); i++) {
     LinkedInAttributes.formalName = nodes.item(i).getNodeValue();
 }

nodes = (NodeList) xpath.evaluate(xLastName,
         document.getDocumentElement(),
         XPathConstants.NODESET);


 for (int i = 0; i < nodes.getLength(); i++) {
     LinkedInAttributes.formalName += " " +nodes.item(i).getNodeValue();
 }

 // get occupation
 nodes = (NodeList) xpath.evaluate(xOccupation,
                        document.getDocumentElement(),
                         XPathConstants.NODESET);

 for (int i = 0; i < nodes.getLength(); i++) {
     LinkedInAttributes.occupation = nodes.item(i).getNodeValue();
 }
 // split occupation and employer
 String[] Occ_Employer = LinkedInAttributes.occupation.split(" at ");
 LinkedInAttributes.occupation = Occ_Employer[0];
 if(Occ_Employer.length > 1) {
    LinkedInAttributes.employer = Occ_Employer[1];
 }

 // get location
nodes = (NodeList) xpath.evaluate(xLocation,
         document.getDocumentElement(),
         XPathConstants.NODESET);

 for (int i = 0; i < nodes.getLength(); i++) {
     LinkedInAttributes.location += nodes.item(i).getNodeValue();
 }

 LinkedInAttributes.location =
         LinkedInAttributes.location.replace(" Area", "");
 LinkedInAttributes.location =
         LinkedInAttributes.location.replace("Greater ", "");
```

```
        nodes = (NodeList) xpath.evaluate(xEducation,
                                document.getDocumentElement(),
                                 XPathConstants.NODESET);
        // get education
        for (int i = 0; i < nodes.getLength(); i++) {
            LinkedInAttributes.education += nodes.item(i).getNodeValue();
        }

        // get groups an map to political attribute
        nodes = (NodeList)xpath.evaluate(xGroups,
                document.getDocumentElement(),
                XPathConstants.NODESET);

        for (int i = 0; i < nodes.getLength(); i++) {
            LinkedInAttributes.politicalAffiliation += nodes.item(i).getNodeValue();
        }

        // get associations and map to lobby affiliations
        nodes = (NodeList) xpath.evaluate(xAssociations,
                document.getDocumentElement(),
                XPathConstants.NODESET);

        for (int i = 0; i < nodes.getLength(); i++) {
            LinkedInAttributes.lobbyAffiliation += nodes.item(i).getNodeValue();
        }

       // get interests and map to special interests
       nodes = (NodeList)xpath.evaluate(xInterests,
                document.getDocumentElement(),
                XPathConstants.NODESET);

        for (int i = 0; i < nodes.getLength(); i++) {
            LinkedInAttributes.specialInterests += nodes.item(i).getNodeValue();
        }
        } // end try
       catch (Exception ex)
       {
           return LinkedInAttributes;
       }
        return LinkedInAttributes;
    }  // end method
}  // end class
```

```
package MediaClients;

import ProvenanceAttributes.EditDistance;
import ProvenanceAttributes.ProvenanceAttributes;
import java.io.UnsupportedEncodingException;
import java.net.MalformedURLException;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPathExpressionException;

import org.json.JSONException;
import java.io.IOException;

import java.util.HashSet;
import org.json.JSONArray;
import org.json.JSONObject;

import org.openide.util.Exceptions;
import org.xml.sax.SAXException;

public class FacebookAPIconnection {

    private static int INFINITY = 10000;

    ////////////////////////////////////////////////////////////////////
    //
    //                      API calls
    //
    ////////////////////////////////////////////////////////////////////

    String accesstoken;
    String accesssecret;
    private static String CONSUMER_KEY = "";
    private static String CONSUMER_SECRET =  "";

    private String APPLICATION_ID = "";

    private String REDIRECT_URI
            = "http://www.facebook.com/connect/login_success.html";

    private String REQUEST_TOKEN_ENDPOINT_URL =

            "https://graph.facebook.com/oauth/authorize?"
            + "client_id=" + APPLICATION_ID
            + "&redirect_uri=http://www.facebook.com/connect/login_success.html"
            + "&response_type=token";
       //     + "&response_type=user_agent&display=popup";

    private String FIELDS
       = "&fields=id,name,gender,birthday,email,website,hometown,location,
             timezone,religion,political,relationship_status,
             interested_in,meeting_for,bio,quotes,about,link";

    static String ACCESS_TOKEN = "";

    ////////////////////////////////////////////////////////////////////
    // Method to open a browser and allow user to copy access token
    // used to search Facebook API values
    public void DisplayToken()
            throws UnsupportedEncodingException, MalformedURLException,
            IOException, SAXException, ParserConfigurationException,
                                                    XPathExpressionException {
```

```
      FacebookHTMLprocessing scraper = new FacebookHTMLprocessing();

    scraper.CobraScrape(REQUEST_TOKEN_ENDPOINT_URL);
    Runtime.getRuntime().exec(
            "rundll32 url.dll,FileProtocolHandler "
               + REQUEST_TOKEN_ENDPOINT_URL);
} // end method

////////////////////////////////////////////////////////////////
// Method to search for Facebook users via the API
//
public ProvenanceAttributes IdentifyAttributes(String formalName,
        String location)
        throws UnsupportedEncodingException, MalformedURLException,
        IOException, JSONException, XPathExpressionException,
        SAXException, ParserConfigurationException {

    HTMLreader reader = new HTMLreader();
    FacebookHTMLprocessing scraper = new FacebookHTMLprocessing();
    ProvenanceAttributes tempAttr = new ProvenanceAttributes();

    JSONObject response;
    int bestScore = INFINITY;
    int currentScore = 0;
    int bestIndex = 0;

    // search Facebook API for formal name
    String[] tokens = formalName.split(" ");

    String requestString = "https://graph.facebook.com/search?q=";
    String query = "";
    for (int x=0; x < tokens.length; x++) {
        query += tokens[x];
        if((x+1) < tokens.length) {
             query +=  "%";
        }
    }
    requestString += query + "&type=user&access_token=" + ACCESS_TOKEN;

    String resultString = reader.HTTPSreadHTMLFile(requestString);

    if(resultString.equalsIgnoreCase("Bad Request")) {
        Exception ex = null;
        Exceptions.printStackTrace(ex);
    }
    response = new JSONObject(resultString);

    // if user search was not successful check page list

    if(response.optJSONArray("data").length() == 0) {

        requestString += query + "&type=page&access_token=" + ACCESS_TOKEN;

        resultString = reader.HTTPSreadHTMLFile(requestString);

        if(resultString.equalsIgnoreCase("Bad Request")) {
            Exception ex = null;
            Exceptions.printStackTrace(ex);
        }

        response = new JSONObject(resultString);
```

```
    } // end if

    // find best match from search results

    JSONArray searchlist = response.optJSONArray("data");

    for(int i = 0; i < searchlist.length(); i++) {
        String tempID = searchlist.optJSONObject(i).optString("id");

        String profilePage = getProfileURL(tempID);

        String profile = getFullUserProfile(tempID);

        currentScore = LocationScore(location,
                scraper.getFacebookLocation(profilePage));

        if ( currentScore < bestScore){
                bestScore = currentScore;
                bestIndex = i;
            } // end if
    }  // end for

    if(bestScore < INFINITY) {
        tempAttr = scraper.ScrapeSingleFBProfile(
                getProfileURL(
                searchlist.optJSONObject(
                bestIndex).optString("id")));
    } // end if
    return tempAttr;
} // end method

////////////////////////////////////////////////////////////////
//
// Method to get facebook profile url
//
public String getProfileURL(String profileID)
        throws IOException, JSONException {

    HTMLreader reader = new HTMLreader();
    JSONObject response;

    String requestString = "https://graph.facebook.com/"
           + profileID
           + "&access_token=" + ACCESS_TOKEN;

     String temp = reader.HTTPSreadHTMLFile(requestString);

     if(temp.isEmpty()) {
         return "";
     }
     response = new JSONObject(temp);

     return response.optString("link") + "?sk=info";

} // end method

////////////////////////////////////////////////////////////////
//
// Method to get facebook profile
//
```

```
public String getBriefProfile(String profileID)
        throws IOException, JSONException {

    HTMLreader reader = new HTMLreader();
    JSONObject response;

    String requestString = "https://graph.facebook.com/"
           + profileID
           + "&access_token=" + ACCESS_TOKEN;// + FIELDS;

     String temp = reader.HTTPSreadHTMLFile(requestString);
     response = new JSONObject(temp);

     return response.optString("link");

} // end method

//////////////////////////////////////////////////////////////
//
// Method to get facebook profile (includes fields)
//
public String getFullUserProfile(String profileID)
        throws IOException, JSONException {

    HTMLreader reader = new HTMLreader();
    JSONObject response;

    String requestString = "https://graph.facebook.com/"
           + profileID
           + "&access_token=" + ACCESS_TOKEN; // + FIELDS;

     String temp = reader.HTTPSreadHTMLFile(requestString);

     if(temp.isEmpty()) {
         return "";
     }

     if(!temp.startsWith("{")) {
         return "";
     }
     response = new JSONObject(temp);

     return response.optString("link");

} // end method

//////////////////////////////////////////////////////////////////////
//
//
public String getJSONAttributes(String input)
           throws UnsupportedEncodingException,
           MalformedURLException, IOException {

       HTMLreader reader = new HTMLreader();
       String result = "";
       JSONObject resp;

    try {
        resp = new JSONObject(reader.HTTPSreadHTMLFile(
                "https://graph.facebook.com/villaraigosa?access_token="
                                    + ACCESS_TOKEN));
```

```
            JSONArray search =
                    resp.optJSONObject(
                    "SearchResponse").optJSONObject(
                    "Web").optJSONArray("Results");
            int x = search.length();
            JSONObject searchresponse = new JSONObject();
            searchresponse = search.optJSONObject(0);

            String title = searchresponse.optString("Title");
            String description = searchresponse.optString("Description");
            String url = searchresponse.optString("Url");
            String durl = searchresponse.optString("DisplayUrl");
            String datetime = searchresponse.optString("DateTime");

            result += " " + title + " " + description + " " + url + " "
                    + durl + " " + datetime;
            } catch (JSONException ex) {
                     Exceptions.printStackTrace(ex);
            }
            return result;
    } // end method


    ////////////////////////////////////////////////
    // Method to compute profile score for a Facebook user
    // that is a candidate for alpha's user page based on location
    // This method should be combined with other metrics to yield a score
    private int LocationScore(String twitterLocation, String fbLocation)

    {

        EditDistance function = new EditDistance();

        // if there si no location return infinity
        if(fbLocation.isEmpty()){
            return INFINITY;
        }
        return function.computeEditDistance(twitterLocation, fbLocation);
    } // end method
}  // end class
```

```
package MediaClients;

import ProvenanceAttributes.ProvenanceAttributes;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.security.Policy;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathFactory;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import org.lobobrowser.html.HttpRequest;
import org.openide.util.Exceptions;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.html2.HTMLElement;
import org.w3c.tidy.Tidy;
import java.io.Reader;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.lobobrowser.html.UserAgentContext;
import org.lobobrowser.html.domimpl.HTMLDocumentImpl;
import org.lobobrowser.html.parser.DocumentBuilderImpl;
import org.lobobrowser.html.parser.HtmlParser;
import org.lobobrowser.html.parser.InputSourceImpl;
import org.lobobrowser.html.test.SimpleUserAgentContext;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.html2.HTMLCollection;
import org.xml.sax.SAXException;
/**
 *
 * @author gbarbier
 */
public class FacebookHTMLprocessing {

    // XPATH query to get first name from Facebook public profile page
    private static String xFirstName = "//span[@class='given-name']/text()";

    // XPATH query to get first name from Facebook public profile page
    private static String xLastName = "//span[@class='family-name']/text()";

    // XPATH query to get occupation from Facebook public profile page
    private static String xOccupation = "//p[@class='title ']/text()";

    // XPATH query to get location from Facebook public profile page
    private static String xLocation = "//dd[@class='locality ']/text()";

     // XPATH query to get education from Facebook public profile page
```

```
private static String xEducation
     = "//dd[@class='summary-education']/ul/li/text()";

// XPATH query to get groups from Facebook public profile page
private static String xPolitical = "//dd[@class='pubgroups']/p/text()";

// XPATH query to get associaitons from Facebook public profile page
private static String xAssociations
     = "//div[@class='group-data']/a/strong/text()";

    // XPATH query to get interests from Facebook public profile page
private static String xInterests = "//dd[@class='interests']/p/text()";

// set up xpath
private XPathFactory factory = XPathFactory.newInstance();
private XPath xpath = factory.newXPath();

//////////////////////////////////////////////////////
// Method to convert URL to DOM for XPATH processing
private Document jtidyURLToDom(String htmlsource)
        throws MalformedURLException, IOException {

    Tidy tidy = new Tidy();
    URLConnection conn = null;
    URL url = null;
    InputStreamReader in = null;

     try
    {
        url = new URL(htmlsource);
    }
    catch ( MalformedURLException ex)
    {
         Exceptions.printStackTrace(ex);
         return null;
    }
    try
    {
                conn = url.openConnection();
                HttpURLConnection huc = (HttpURLConnection) conn;
                huc.setRequestProperty("User-Agent", "Mozilla/4.5");
                if(huc.getResponseCode()==400||huc.getResponseCode()==404)
                {
                     return null;
                }
    }
    catch (IOException ex) {
        Exceptions.printStackTrace(ex);
        return null;
    }

    in = new InputStreamReader(conn.getInputStream());

    tidy.setQuiet(true);
    tidy.setShowWarnings(false);

    return tidy.parseDOM(in, null);

}  // end method

//////////////////////////////////////////////////////////////
```

```
  // Method for debugging XPATH results
  private String getNodeText(Node node) {

      String result = "";

      switch (node.getNodeType()) {
          case Node.ELEMENT_NODE:
              result = "<" + node.getNodeName();

              NamedNodeMap map = node.getAttributes();
              for (int i = 0; i < map.getLength(); i++) {
              result += " " + map.item(i).getNodeName() +
                      "=\"" + map.item(i).getNodeValue() + "\"";
              }
              result +=">\n";
              return result;
          case Node.ATTRIBUTE_NODE:
              return node.getNodeName() + "=\""
                              + node.getNodeValue() + "\"\n";
          case Node.TEXT_NODE:
              return "TEXT NODE "
                      + node.getNodeName() + " "
                      + node.getNodeValue() + "\n";
          case Node.CDATA_SECTION_NODE:
              return node.getNodeValue() + "\n";
          case Node.PROCESSING_INSTRUCTION_NODE:
              return node.getNodeValue() + "\n";
          case Node.DOCUMENT_NODE:
          case Node.DOCUMENT_FRAGMENT_NODE:
              return node.getNodeName() + "=" + node.getNodeValue() + "\n";
          }
      return result;
  }  // end method

  //////////////////////////////////////////////////////////////
  // Method to get lcation from a Facebook public profile page
  public String getFacebookLocation(String profilepage)
          throws MalformedURLException, IOException, XPathExpressionException,
          SAXException, ParserConfigurationException  {

      ProvenanceAttributes FacebookAtrributes = new ProvenanceAttributes();

      FacebookAtrributes = ScrapeSingleFBProfile(profilepage);

      return FacebookAtrributes.location;
  }  // end method

  //////////////////////////////////////////////////////////////
  // Method to scrape attributes from Facebook public profile page
  // return provenance attribute object, using XPATH

  public ProvenanceAttributes ScrapeFacebookAttributes(String profilepage)
          throws MalformedURLException, IOException,
          XPathExpressionException  {

      ProvenanceAttributes FacebookAttributes = new ProvenanceAttributes();

      // if no URL associated with alpha return empty attribute set
      if(profilepage == null) {
          return FacebookAttributes;
      }
```

```
 Document document = jtidyURLToDom(profilepage);

 if(document == null) {
     return FacebookAttributes;
 }

NodeList nodes = (NodeList) xpath.evaluate(xFirstName,
         document.getDocumentElement(),
         XPathConstants.NODESET);

 // get special interests
 for (int i = 0; i < nodes.getLength(); i++) {
     FacebookAttributes.formalName = nodes.item(i).getNodeValue();
 }

nodes = (NodeList) xpath.evaluate(xLastName,
         document.getDocumentElement(),
         XPathConstants.NODESET);

 // get special interests
 for (int i = 0; i < nodes.getLength(); i++) {
     FacebookAttributes.formalName += " " +nodes.item(i).getNodeValue();
 }

 // get occupation
 nodes = (NodeList) xpath.evaluate(xOccupation,
                         document.getDocumentElement(),
                          XPathConstants.NODESET);

 for (int i = 0; i < nodes.getLength(); i++) {
     FacebookAttributes.occupation = nodes.item(i).getNodeValue();
 }
 // split occupation and employer
 String[] Occ_Employer = FacebookAttributes.occupation.split(" at ");
 FacebookAttributes.occupation = Occ_Employer[0];
 if(Occ_Employer.length > 1) {
    FacebookAttributes.employer = Occ_Employer[1];
 }

 // get location
nodes = (NodeList) xpath.evaluate(xLocation, // *** gets location
         document.getDocumentElement(),
         XPathConstants.NODESET);

 // get special interests
 for (int i = 0; i < nodes.getLength(); i++) {
     FacebookAttributes.specialInterests += nodes.item(i).getNodeValue();
 }

 nodes = (NodeList) xpath.evaluate(xEducation,
                         document.getDocumentElement(),
                          XPathConstants.NODESET);
 // get education
 for (int i = 0; i < nodes.getLength(); i++) {
     FacebookAttributes.education += nodes.item(i).getNodeValue();
 }

 // get groups an map to political attribute
 nodes = (NodeList)xpath.evaluate(xPolitical,
         document.getDocumentElement(),
```

```
            XPathConstants.NODESET);

    for (int i = 0; i < nodes.getLength(); i++) {
        FacebookAttributes.politicalAffiliation
                            += nodes.item(i).getNodeValue();
    }

    // get associations and map to lobby affiliations
    nodes = (NodeList) xpath.evaluate(xAssociations,
            document.getDocumentElement(),
            XPathConstants.NODESET);

    for (int i = 0; i < nodes.getLength(); i++) {
        FacebookAttributes.lobbyAffiliation += nodes.item(i).getNodeValue();
    }

   // get interests and map to special interests
   nodes = (NodeList)xpath.evaluate(xInterests,
            document.getDocumentElement(),
            XPathConstants.NODESET);

    for (int i = 0; i < nodes.getLength(); i++) {
        FacebookAttributes.specialInterests += nodes.item(i).getNodeValue();
    }
    return FacebookAttributes;
}  // end method

////////////////////////////////////////////////////////////////////////////
//
//                        regex
//
////////////////////////////////////////////////////////////////////////////

private String locationPattern
    = "\\>Work Info.*?Location.*?u003cdd\\>(.*?)\\\\";
private String livesinPattern ="(Lives in)";
private String politicalViewsPattern
    = "Political Views.*?data_field\\\\\"\\>(.*?)\\\\";
private String politicalPattern = "\\>Party.*?mls\\\\\"\\>(.*?)\\\\";
private String agePattern = "Birthday.*?data_field\\\\\"\\>(.*?)\\\\";
private String convictionPattern
    = "\\>Religious Views.*?data_field\\\\\"\\>(.*?)\\\\";
private String genderPattern = "Gender.*?data_field\\\\\"\\>(.*?)\\\\";

////////////////////////////////////////////////////////////////////
// Method to scrape public FB profile page using regex
public ProvenanceAttributes ScrapeSingleFBProfile(String profileURL)
            throws UnsupportedEncodingException, MalformedURLException,
            IOException, SAXException,
            ParserConfigurationException,
            XPathExpressionException {

    HTMLreader pageResult = new HTMLreader();

    ProvenanceAttributes FacebookAtrributes = new ProvenanceAttributes();

    String FBUserPage = pageResult.readHTMLFile(profileURL);

    if(FBUserPage == null){
        return FacebookAtrributes;
    }
```

```
    Pattern patt = Pattern.compile(locationPattern);
    Matcher m = patt.matcher(FBUserPage);
    while (m.find()) {
        FacebookAtrributes.location = m.group(1);
        }

    if(FacebookAtrributes.location.isEmpty()) {
        patt = Pattern.compile(livesinPattern);
        m = patt.matcher(FBUserPage);
        while (m.find()) {
            FacebookAtrributes.location = m.group(1);
        }
    } // end if

    patt = Pattern.compile(politicalPattern);
    m = patt.matcher(FBUserPage);
    while (m.find()) {
        FacebookAtrributes.politicalAffiliation = m.group(1);
        }

    patt = Pattern.compile(agePattern);
    m = patt.matcher(FBUserPage);
    while (m.find()) {
        FacebookAtrributes.Age = m.group(1);
        }

    patt = Pattern.compile(politicalViewsPattern);
    m = patt.matcher(FBUserPage);
    while (m.find()) {
        FacebookAtrributes.convictions = m.group(1);
        }

    patt = Pattern.compile(convictionPattern);
    m = patt.matcher(FBUserPage);
    while (m.find()) {
        FacebookAtrributes.convictions
            = FacebookAtrributes.convictions + ", " + m.group(1);
        }

    patt = Pattern.compile(genderPattern);
    m = patt.matcher(FBUserPage);
    while (m.find()) {
        FacebookAtrributes.gender = m.group(1);
        }

    return FacebookAtrributes;
} // ScrapeSingleProfile

////////////////////////////////////////////////////////////////
//
//
public void CobraScrape(String inURL)
       throws MalformedURLException, IOException, SAXException,
          ParserConfigurationException, XPathExpressionException {

    // UserAgentContext uacontext = new SimpleUserAgentContext();
    UserAgentContext uacontext = new SimpleUserAgentContext();
    ((SimpleUserAgentContext)uacontext).setExternalCSSEnabled(true);
    ((SimpleUserAgentContext) uacontext).setScriptingEnabled(true);
    ((SimpleUserAgentContext) uacontext).setUserAgent(
```

```
            "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0) Gecko/20100101 Firefox/4.0");
        String platform = ((SimpleUserAgentContext) uacontext).getPlatform();
        ((SimpleUserAgentContext) uacontext).setCookie(null, "");

        DocumentBuilderImpl builder = new DocumentBuilderImpl(uacontext);
        URL url = new URL(inURL);
        StringBuilder page = new StringBuilder();

        InputStream in = url.openConnection().getInputStream();
        try {
            Reader reader = new InputStreamReader(in, "UTF-8");
            InputSourceImpl inputSource = new InputSourceImpl(reader, inURL);

            Document d = builder.parse(inputSource);
            HTMLDocumentImpl document = (HTMLDocumentImpl) d;
            String innerhtml = document.getInnerHTML();
            String innertext = document.getInnerText();
            String namespaceURI = document.getNamespaceURI();
            String textcontent = document.getTextContent();
            String title = document.getTitle();
            String cookie = document.getCookie();
            String something = document.getReferrer();
            String baseURI = document.getBaseURI();
            HTMLElement body = document.getBody();
            NodeList childnodes = document.getChildNodes();
            String name = document.getNodeName();
            String nodeval = document.getNodeValue();
            HTMLCollection anchors = document.getAnchors();
            NodeList ele = document.getElementsByName("Script");
            String domain = document.getDomain();
            String encoding = document.getInputEncoding();
            String refer = document.getReferrer();
            String xmlsource = document.getXmlEncoding();
            String version = document.getXmlVersion();

            HTMLCollection images = document.getImages();
            int length = images.getLength();
            for(int i = 0; i < length; i++) {
                System.out.println("- Image#" + i + ": " + images.item(i));
            }
        } finally {
            in.close();
        }
    } // end method
}  // end class
```

```
package ProvenancePath;

import MediaClients.TwitterAPIConnection;
import ProvenanceAttributes.Alpha;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.MalformedURLException;
import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPathExpressionException;
import oauth.signpost.exception.OAuthCommunicationException;
import oauth.signpost.exception.OAuthExpectationFailedException;
import oauth.signpost.exception.OAuthMessageSignerException;
import oauth.signpost.exception.OAuthNotAuthorizedException;
import org.json.JSONException;
import org.xml.sax.SAXException;
/**
 *
 * @author gbarbier
 */
public class FindPath {

    private static String recipientNamePattern = "(.*?) {1}";
    private static String alphaNodePattern = "RT @(.*?):";

    public ArrayList<Alpha> ProcessRT(String tweetText)
            throws MalformedURLException, IOException, XPathExpressionException,
            SAXException, ParserConfigurationException,
            OAuthMessageSignerException, OAuthNotAuthorizedException,
            OAuthExpectationFailedException, OAuthCommunicationException,
            UnsupportedEncodingException, JSONException {

        ArrayList<Alpha> path = new ArrayList<Alpha>();
        Alpha tempUser = new Alpha();
        ArrayList<String> alphaNodes = new ArrayList<String>();

        Pattern patt = Pattern.compile(recipientNamePattern);
        Matcher m = patt.matcher(tweetText);

        // get first result which will be sender's twitter username
        m.find();
        tempUser.alphaUserName = m.group(1);

        tempUser.recipientSearchProvAttr();
        path.add(tempUser);
        patt = Pattern.compile(alphaNodePattern);
        m = patt.matcher(tweetText);
        while (m.find()) {
            alphaNodes.add(m.group(1));
        }
        for(int i = 0; i < alphaNodes.size(); i++) {
            Alpha temp = new Alpha();
            temp.alphaUserName = alphaNodes.get(i);
            temp.recipientSearchProvAttr();
            path.add(temp);
        }
        return path;
    } // end method
}
```