Using Word and Phrase Abbreviation Patterns to Extract Age From Twitter Microtexts

Approved by Supervising Committee:

Dr. Manjeet Rege
Committee Chair, Department of Computer Science

Dr. Cecilia Ovesdotter Alm
Reader, Department of English

Dr. Reynold Bailey,
Observer, Department of Computer Science

Using Word and Phrase Abbreviation Patterns to Extract Age From Twitter Microtexts

by

Nathaniel Moseley

THESIS

Presented to the Faculty of

the B. Thomas Golisano College of Computing and Information Sciences

Department of Computer Science

Rochester Institute of Technology

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

Rochester Institute of Technology

May 20, 2013

Acknowledgments

I wish to thank my friends and mother who provided valuable feedback on the reading of this document through its development, as well as reminding me to keep at the task of writing. I would also like to thank my parents for supporting me throughout my education.

I must extend gigantic thanks to my thesis committee, for their time spent proofreading this document and for invaluable input on the progress of my experimentation and ideas. This would have been much more difficult without their help.

Abstract

The wealth of texts available publicly online for analysis is ever increasing. Much work in computational linguistics focuses on syntactic, contextual, morphological and phonetic analysis on written documents, vocal recordings, or texts on the internet. Twitter messages present a unique challenge for computational linguistic analysis due to their constrained size. The constraint of 140 characters often prompts users to abbreviate words and phrases. Additionally, as an informal writing medium, messages are not expected to adhere to grammatically or orthographically standard English. As such, Twitter messages are noisy and do not necessarily conform to standard writing conventions of linguistic corpora, often requiring special pre-processing before advanced analysis can be done.

In the area of computational linguistics, there is an interest in determining latent attributes of an author. Attributes such as author gender can be determined with some amount of success from many sources, using various methods, such as analysis of shallow linguistic patterns or topic. Author age is more difficult to determine, but previous research has been somewhat successful at classifying age as a binary (*e.g.* over or under 30), ternary, or even as a continuous variable using various techniques.

Twitter messages present a difficult problem for latent user attribute analysis, due to the preprocessing necessary for many computational linguistics analysis tasks. An added logistical challenge is that very few latent attributes are explicitly defined by users on Twitter. Twitter messages are a part of an enormous data set, but the data set must be independently annotated for latent writer attributes not defined through the Twitter API before any classification on such attributes can be done. The actual classification problem is another particular challenge due to restrictions on tweet length.

Previous work has shown that word and phrase abbreviation patterns used on Twitter can be indicative of some latent user attributes, such as geographic region or the Twitter client (iPhone, Android, Twitter website, *etc.*) used to make posts. This study explores if there there are age-related patterns or change in those patterns over time evident in Twitter posts from a variety of English language authors.

This work presents a growable data set annotated by Twitter users themselves for age and other useful attributes. The study also presents an extension of prior work on Twitter abbreviation patterns which shows that word and phrase abbreviation patterns can be used toward determining user age. Notable results include classification accuracy of up to 82.6%, which was 66.8% above relative majority class baseline (ZeroR in Weka) when classifying user ages into 10 equally sized age bins using a support vector machine classifier and PCA extracted features.

Contents

List of Tables

List of Figures

1 Introduction

Discovering latent data about authors of texts is a recognized topic of interest in natural language processing. With the rising popularity of the internet, many texts and recordings that can be subjects of computational linguistics analysis are available online, and online texts themselves represent ever growing corpora. Computational linguistics provides methods to process and analyze these large language collections. Many corpora of internet language have received such attention, such as blog posts, online news, and scientific publications [41, 42].

One of the newest corpora developing and increasingly receiving attention is a set of texts linguists have termed *microblogs*. These include short, often character-limited messages, such as those found on Facebook update messages, SMS cell phone text messages, and Twitter messages. These present an interesting type of linguistic corpus, but often, particularly in the case of Twitter, the texts are noisy and more challenging to work with because of nonstandard language use. In addition, character restrictions prompt authors to develop and use word and phrase abbreviations to convey their messages in fewer characters [18]. Cook and Stevenson identified 12 types of abbreviations often used in SMS messages [11]. Usage of these abbreviations results in messages with a significant percentage of tokens that are out-of-vocabulary (OOV) for the language in which they are written. Such increased linguistic sparsity can make linguistic analysis, such as for context, genre, and topic detection, more difficult to perform [29].

Part of the process of preparing Twitter messages for analysis involves mapping OOV word and phrase abbreviations to in-corpus equivalents. Gouws *et al.* identified 9 word abbreviation patterns used in Twitter messages which accounted for over 90% of the lexical transformations used in a large collection of English tweets [17]. These abbreviation patterns are word-level changes done in order to save characters in a message. The 9 identified abbreviation pattern features are further discussed in subsection 5.4. Several of the types of abbreviation patterns are phonemic changes, substituting a character for a phoneme or whole word. Gouws *et al.* used these patterns to identify the region from which English-writing Twitter users were posting (according to time zone data provided by Twitter) as well as the client (iPhone, Android, Twitter website, *etc.*) used to post the message [17]. Based on Sarawgi *et al.*'s success of using deep syntactic patterns and shallow, token-level linguistic features to identify author gender [42] and Rao *et al.*'s success with identifying user age [39], it is reasonable to assume that such abbreviation patterns can help identify user age on Twitter.

Some prior studies have looked at identifying Twitter user age with relative success, but each had to develop a data set by hand. Because of the required and costly manual annotation and time involved, the collected data sets were relatively small and generally used binary or ternary

classification. Rao *et al.* began with a seed set of users selected from keywords in their account description. They then had annotators choose from the seed set and users connected to them that definitely fit in each of two bins: over or under 30 [39]. While the data were carefully selected for accuracy, such age classification is still a challenging problem. Rao *et al.* only managed 9-14% correct classification over the baseline in evenly split binary classification [39].

Age is a constantly changing real-valued attribute and is not strictly linked to communication preferences. Those discourse patterns that are sometimes indicative of age can not always be relied on either, as discourse style changes with social context. Increased use of informal language can be indicative of an adolescent [37], but the same person may completely change their register when in a conversation with an employer or other adult.

Age and other demographic data are not supplied by Twitter. Users are able to supply a location, but it can be any value they want, and it can be changed at any time. Additionally, users can supply a description, which may or may not hint at demographic data. Geography is best represented via the timezone a user selects to post from or view Twitter from, but more fine-grained location and other demographic data are not available. Demographic data are used a great deal in targeted marketing, which seeks to advertise relevant products to a user based on their demographic information. By only advertising to those demographics who would be most interested in a product, advertisers can save on costs and increase revenues [4]. When such information is not available, it must be determined other ways, such as through text analysis applied to social media services, such as Twitter or Facebook.

As demographic-tagged Twitter data sets are sparse and mostly not available, I contribute and present a novel data set developed to improve future Twitter research. In addition, I report on analysis and processing methods, and I present updating findings regarding user age classification using word and phrase abbreviations found in Twitter messages.

Section 2 gives a background of computational linguistics as it pertains to my topic. Section 3 covers previous work pertaining to Twitter analysis in computational linguistics. Section 4 explains the details of my hypotheses as they relate to existing problems. Section 5 outlines the methodologies that I used in collecting and analyzing data. Section 6 presents an analysis of the collected data set and extracted abbreviation features. Section 7 presents the details of my experimentation and results. Finally, section 8 and section 9 present my conclusions and notes for continuing research on this topic.

2 Background

Computational linguistics was boosted by machine translation efforts in the 1950s, because researchers believed that computers would be able to produce effective translations more quickly than their human counterparts [21]. Today, computational linguistics has many other applications beyond automatic translation. An understanding of language and meaning can allow a computer to more effectively interact with its human operators and can be used for text data analytics. This takes many forms: In advertising, products such as Google's Adsense show topic-centered ads based on page content [23]; In email, spam filters analyze messages for typical real usage and spam usage then automatically flag messages that seem like spam; In intelligent human-computer interaction, a computer can be able to communicate with and adapt to its user more effectively if it knows different ways to present information based on its user's age, education, emotional state, or other attributes.

Computerized linguistic analysis has additional data available when analyzing speech than when analyzing text. In speech, patterns in prosody, filled pauses such as *umm*, and so on can indicate various details about a speaker. Additionally, speech can often be analyzed as part of a conversation. Social factors such as social status (a boss or a child), gender, *etc.* can affect vocabulary and many other linguistic factors in a conversation [14]. In contrast, when analyzing text, especially online texts, those extra speech components are not available, and in many cases, conversational information is not either. This sparsity restricts analysis to certain linguistic patterns.

Most studies analyze data that falls into two categories: shallow or deep linguistic features. These categories are not always consistent between linguists, however. Shallow features often include information that is based on analysis of surface text, such as through tokenization, sentence splitting, Part-of-Speech (PoS) tagging, lemmatization, sentence phrase structures, word frequencies, genre or topic, and formality [16, 35]. Deep linguistic features involve further linguistic analysis of a text. These features are usually context independent and deal with individual words, phrases, and sets of characters. Deep features can include phrase type, voicing, word and sentence length, and word or character n-grams [9, 22]. Different studies assign features to different categories, depending on perspective. Those above are based on the cited works.

Age is an acknowledged factor in language use, as noted by Wagner [49]. Wagner explains that a person's writing and speech patterns change over time as they learn and develop ('age grading') [49]. An individual goes through many stages of language use through childhood, adolescence, and adulthood. In childhood, language is acquired and understanding and conversational-interaction skills are developed. Adolescence marks a period of change in many respects, and a person trying to find their identity socially also explores their identity linguistically. Into adulthood, language

use continues to change, often in response to changes in community language use ('generational change'). "Women have been repeatedly identified as the leaders of generational language change" [49]. The variety of ways that language use changes over time and that language use with respect to age varies differently depending on gender and other individual features make age classification a challenging and attention-deserving problem.

Texts found on the internet represent a gigantic collection for analysis of linguistic changes with respect to age. The simplicity and low cost of writing on the internet allows individuals to publish a prolific library of formal and informal texts. Many people keep blogs, write on message boards and newsgroups, or participate in social networking sites. The types of writing available range from long, scientific writing, which adheres to language standards with standard grammar, syntax, and orthography, to short, informal messages, rich in nonstandard language found on Twitter.

2.1 Twitter and the Character of Tweets

Twitter is a relatively new service, made public in 2006, which allows users to post 140 character *updates*. They can follow other users, such as friends, celebrities, or companies to receive a live digest of those users' updates. Users can engage in public or private conversations with these short messages, forward messages they think their followers will be interested in by *retweeting*, or just post whatever they are doing, thinking, or want to write [27]. Twitter has reported having over 140 million active users and 340 million tweets per day, meaning there is an incredible amount of information and text exchanged on Twitter [26].

The Twitter service provides two special keyword annotations of note. First is the *@-username* construct, often used in a conversation between two users, noting that a tweet is a reply to something another user has said, as a method of bringing a message to the attention of another user, or, in a *retweet*, noting the original author of a quoted phrase or message. An at sign precedes a series of up to 15 alphanumeric characters and underscores which correspond with the username of a Twitter account. The entire token, including the at sign is hyperlinked, pointing to the associated home page of that Twitter account.

The second construct is the *hashtag*, denoted with a '#' symbol, which provides a tagging interface for use in tweets. The hashtag symbol is followed by a keyword or phrase (no spaces) that is relevant to the tweet. The hyperlink created from the hashtag points to a page that lists all other tweets with the same tag. As hashtags can occur anywhere within a tweet, they make the process of cleaning a tweet into a standard language sentence somewhat difficult, as there is no strict rule whether or not the hashtag is a part of a sentence or auxiliary. Hashtags that are most used are generally short. They are frequently abbreviated or are short word phrases with the spaces

removed [12]. Commonly, hashtags at the end of a tweet are dropped from sentence cleaning, and those within sentences are treated as relevant words and have the '#' symbol stripped for analysis purposes. Gimpel *et al.* found 35% of hashtags were treated as words rather than tags [15].

In July 2011, Twitter crossed the one million mark for developer applications registered to use the Twitter API [24]. Twitter provides developers and researchers a robust API with which to interact with accounts and access user information and tweets. Every user defines a username, and optionally a real name, description, and location. Also available are the account's associated timezone and the account creation timestamp. Each tweet is associated with several pieces of information in addition to the message, such as its timestamp, the Twitter client it was posted from, if it was part of a conversation, a retweet, and the count of people who retweeted it. However, Twitter does not elicit other data about the author that might be useful for latent attribute analysis, such as age or other demographic information.

Several corporate entities have published various studies of the demographics on Twitter. Most use data mining techniques to extract a set of demographic features from the defined user attributes, relying on instances where users have published their age, gender, or location as part of their profile or somewhere in their tweets. Others, such as the Pew Research Center, utilize other forms of data collection. In their internet and social media use survey, they used phone interviews to get data on internet and social media (Twitter included) use and demographics [44]. Consumers of this information tend to be in marketing, as companies are always seeking the best way to advertise to their target audiences.

3 Related Work

A variety of work has been published that focuses on linguistic analysis for author age, much of which focuses on lexical and contextual clues, such as analyzing topic and genre or n-gram patterns. N-gram patterns can refer to several elements of linguistic analysis. On a lexical level, n-grams are groupings of length *n* of word tokens, found adjacently in text. They are also referred to as unigrams, bigrams, trigrams, *etc.* for *n* of *1*, *2*, and *3* respectively. On a character level, n-grams can refer to groupings of adjacent characters within a word, in much the same way as groupings of words. Depending on the approach, special characters may be used as marks at word boundaries. As an example, Cavnar presents trigrams for the word *text*, such as __T, _TE, TEX, EXT, XT_, and T__ [7]. This work, as many others, focus on token analysis. Tokens, as defined for this work, consist of sets of characters, generally separated by spaces in the original text, but not always. Punctuation tokens (those consisting of only punctuation characters) are separated from

adjoining word tokens. Additionally, words recognized as contractions are separated into two word tokens, *e.g.* "shouldn't" → "should" and "n't".

Garera and Yarowsky used linguistic features (amount of speech in conversation, length of utterances, usage of passive tense, *etc.*) for characterizing types of speech in telephone conversations between partners in their research. They found that such sociolinguistic features improved the accuracy of binary attribute classification for speaker age, gender, and native language [14]. Many features that are available in an audio corpus, such as prosody and vocal inflections, are not available in a purely textual corpus, making related classification problems more challenging. Garera and Yarowsky were able to get about 20% improvement over guessing the most common class when classifying phone conversations for age with a binary classifier [14].

Nguyen *et al.* went a step beyond many other studies and classified age as a continuous variable in online texts and transcribed telephone conversations. They found that stylistic, unigram, and part of speech characteristics were all indicative of author age with mean absolute errors between 4.1 and 6.8 years [37].

Rosenthal and McKeown analyzed online behavior associated with blogs (*i.e.* usually larger depth than tweets) and found that behavior (number of friends, posts, time of posts, *etc.*) could effectively be used in binary age classifiers, in addition to linguistic analysis techniques similar to those mentioned above [41].

Similarly, many works investigating linguistic gender and age indicators focus on non-contextual and deeper analysis, such as through statistical language models. A statistical language model is a probability distribution over words, sentences, phrases, or characters in a language. A language model might hold probabilities representing n-grams. Those probabilities can be used in various types of linguistic analysis [40].

With respect to examining another demographic feature, Sarawgi *et al.* explored non-contextual syntactic patterns and morphological patterns to find if gender differences extended further than topic analysis and word usage could indicate. They used probabilistic context-free grammars, token-based statistical language models, and character-level language models, that learn morphological patterns on short text spans. With these, they found that gender is evident in patterns at the character-level, even in modern scientific papers [42].

Much of linguistic analysis that has been completed focuses on formal writing or conversation transcripts, which generally conform to standard English corpora and dialects, syntax, and orthography. Recently, more works have begun to look at new written and online texts which do not tend toward prescriptive standards, including SMS messages and social networking blurbs, such as Facebook and Twitter messages. There are various challenges when trying to analyze these typ-

ically noisy texts. Misspellings, unusual syntax, and word and phrase abbreviations are common in these texts, which many linguistic analysis tools do not deal with.

Rao *et al.* found n-gram and sociolinguistic cues, such as a series of exclamation marks, ellipses, character repetition, use of possessives, *etc.*, in unaltered Twitter messages could be used to determine age (binary: over or under 30), gender, region, and political orientation, similar to works that have focused on more formal writing. These textual sociolinguistic features yielded 20-25% improvements over relative baselines [39]. These improvements are similar to those found in this work. In the best cases, classifiers examined in this work using only numeric abbreviation features performed almost 5% better. Abbreviation features combined with n-gram features showed improvements of as much as 66.8%.

Gimpel *et al.* developed a part-of-speech tagger designed to handle the unique Twitter lexicon by extending the traditionally labeled parts of speech to include new types of text such as emoticons and special abbreviations [15]. Part-of-speech analysis can be used as a part of normalizing noisy text, or the part-of-speech patterns can be used themselves as features for classification.

Some research takes a different approach to noisy text, such as that found on Twitter. Before performing traditional text analysis, noisy texts are often first cleaned or normalized. There are various ways to approach the text normalization problem, such as treating it as a spell-checking problem, a machine translation problem, in which messages are translated from a noisy origin language to a target language, or as an automatic speech recognition (ASR) problem [45]. ASR is often useful for analysis of texts such as SMS, since many of the OOV words are phoneme abbreviations using numbers [17]. Kaufmann and Kalita presented a system for normalizing Twitter messages into standard English. They observed that pre-processing tweets for orthographic modifications and twitter-specific elements (@-usernames and # hashtags) and then applying a machine translation approach worked well [29].

Gouws *et al.* built on top of the techniques of Contractor *et al.* [10] using the pre-processing techniques of Kaufman and Kalita [29] to determine types of lexical transformations used to create OOV tokens in Twitter messages. Such transformations include phonemic character substitutions (“see” $\rightarrow$ “c”; “late” $\rightarrow$ “l8”), dropping trailing characters or vowels (“saying” $\rightarrow$ “sayin”), and phrase abbreviations (“laughing out loud” $\rightarrow$ “lol”). These transformations are discussed further in subsection 5.4. Gouws *et al.* analyzed patterns in usage of these transformations compared to user time zone and Twitter client to see if there was a correlation. The analysis showed that variation in usage of these transformations were correlated with user region and Twitter client [17].

In sum, prior work suggests that text-based age prediction is tenable and leaves room for additional study. This thesis seeks to extend prior work and analyze these transformations with respect

to user age.

4 Hypothesis

There are presently some techniques to determine latent user attributes from general texts, but few that specifically target Twitter messages and their unique corpus characteristics. Of those works that have focused on Twitter messages, they have two main types of shortcomings: (1) they focus on a small set of gathered data from hand-picked users, where latent attributes are determined from limited descriptions on user profiles or key tweeted phrases and are entered by human annotators, as opposed to by the Twitter users providing the information themselves; or (2) they use the full set of Twitter users and messages, but tend to be limited to the latent attributes that are provided through the Twitter API.

Based on these observations, first, I present my solution to these issues through collection of a new, more robust data set where the tweeters themselves label their Twitter feeds with demographic information. Second, based on the work of Gouws *et al.*, I hypothesize that word and phrase abbreviation patterns used to write tweets are indicative of user age, as they are indicative of a user's region and Twitter client [17]. Third and last, I hypothesize that usage of these abbreviations changes as a user ages or spends more time using the Twitter service, similar to the ways in which language changes as a person ages and community language use evolves. I present my experimental analysis of collected data seeking to examine these hypotheses.

5 Pre-Experimental Design and Implementation

The pre-experimental work of this thesis is comprised of four parts: (1) collection of a Twitter data set, described in subsection 5.1; (2) pre-processing of the collected user demographic information, described in subsection 5.2; (3) collection of user tweet data, described in subsection 5.3; and (4) extraction of abbreviation pattern features from the collected data, described in subsection 5.4.

5.1 Data Collection

The first contribution of this thesis is collection of a user-driven Twitter data set, containing at a minimum a user's Twitter username, year of birth, and collected tweet IDs. The data set is populated via a user-friendly web form, shown in Figure 1, via Twitter users entering information to be associated with the account(s) they control. It is assumed that people submitting their information

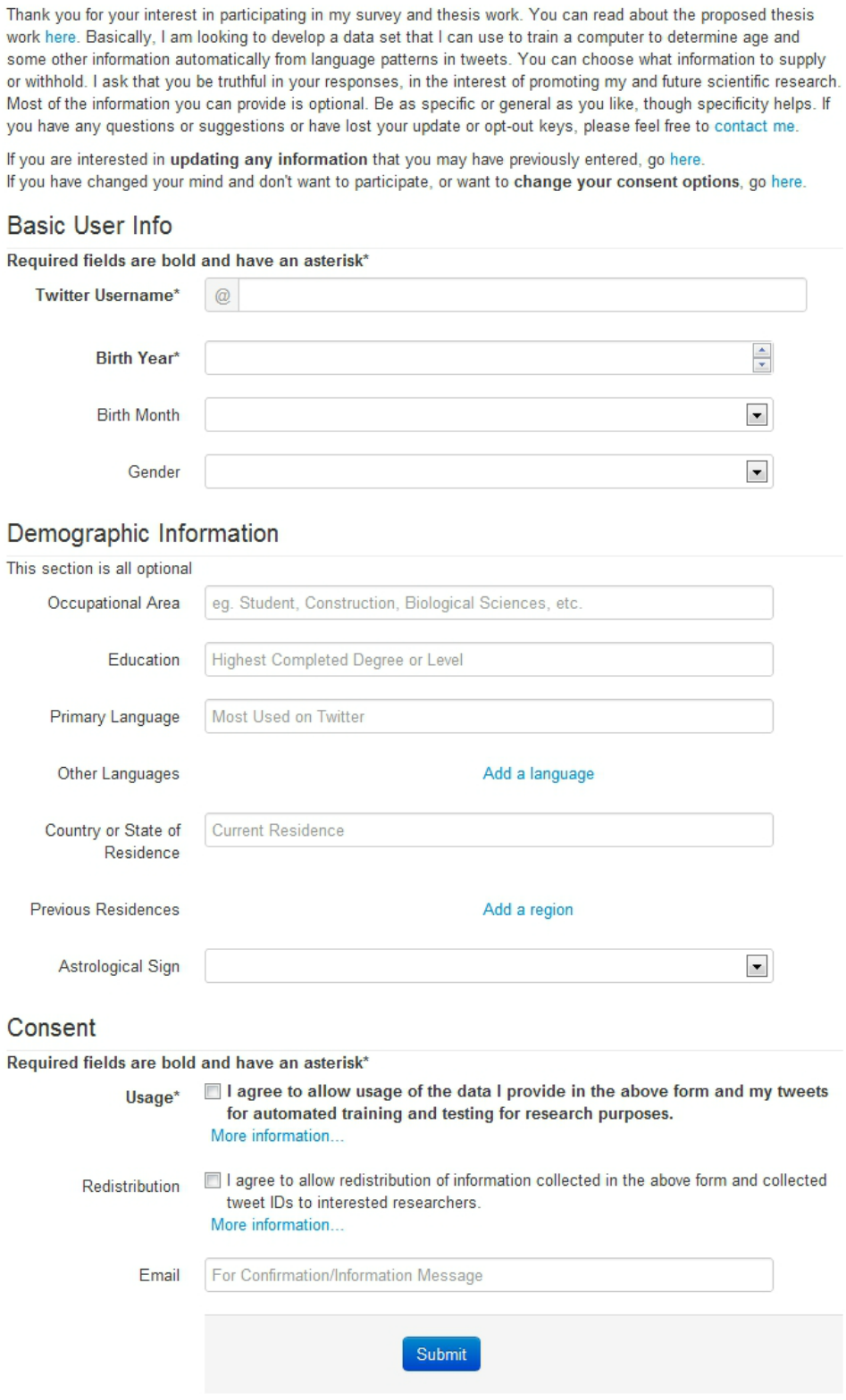
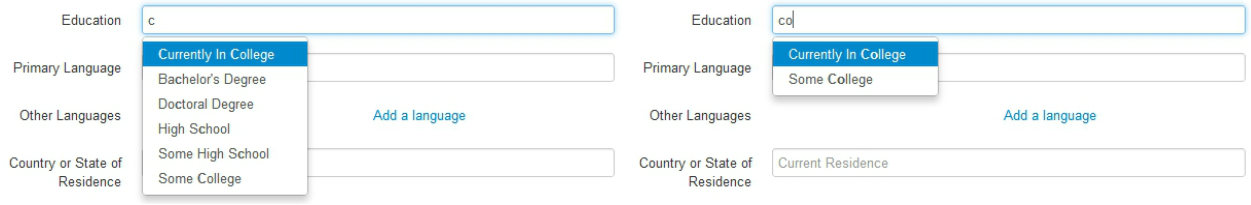


Figure 1: The web form prepared in this study for Twitter users to submit their information



(a) A user types 'c'

(b) A user types 'o' after typing 'c'

Figure 2: The web form suggests options as the user types, sorted alphabetically. The top option in Figure 2a is a user-generated option that has been used more than a threshold number of times. The other options are predefined.

are being truthful in their supplied demographic data, but it is possible that falsified information can be submitted. When the data set is large, the majority of information collected should be truthful, and a submission of falsified information may show as an outlier in some part of the analysis.

The web form explains basic requirements of the data collection and links to pages with more information. Users are informed of the privacy of their data and given the opportunities to allow their data to be redistributed to future researchers, update the information they provide, or opt out entirely from the research. The minimum amount of information collected is Twitter username and year of birth. Twitter users can also supply 8 additional attributes: (1) month of birth, (2) gender, (3) occupational area, (4) highest education level, (5) languages used, (6) regions of residence, (7) astrological sign, and (8) email. Most users appeared willing to include some if not all of this additional information. In the interest of future research, non-identifying demographic information is collected in addition to the age information that will be used for this thesis. Astrological sign is suggested as a control variable, since there has been no support of a scientific link between astrology and personal characteristics [19].

Email is collected for part of the form processing and future automated notifications. On completion of the form, users are shown a confirmation page with links to opt out or to update their information. If they provided an email, they are emailed the same collection of information. Lastly, a Twitter account for this thesis (@NMoseleyThesis) follows the user. In the event that the user's tweets are protected (only authorized users may view their tweets), this allows them to be read for later analysis.

The web form populates a database backend which in turn offers suggestions for the form inputs. As shown in Figure 2, a set of predefined values for each attribute, as well as the most frequently used user-created values, are suggested via JavaScript as a user types in the form. The suggestions help increase initial data coherence. Additionally, each form field only allows certain characters to be entered, according to the type of datum that is expected.

Figure 3: The web form highlights input errors and displays an associated message, requiring the user to correct errors before the form will submit.

Two checkboxes are included in the form which have brief sentences indicating user consent. Adjoining links that point to a more in-depth explanation also show the same explanation on mouseover, if the user has JavaScript enabled. The first checkbox is required to be selected for submission of the form. It explains that the data collected will be used for autonomous analysis and will not be sold or redistributed except when the user checks the second consent box. Checking the second checkbox acknowledges that the user is willing to allow the data collected to be redistributed to interested researchers in the future. It defines collected data to include all entries in the web form, as well as collected tweet identifiers. The tweets themselves cannot be redistributed, but the identifiers produced by Twitter can be, as per the Twitter data use policy [25].

The web form is cross-browser compatible, including on mobile platforms, and performs the same with or without JavaScript enabled. Some added functionality, such as the above input suggestions and consent information mouseover, is not available without JavaScript. All value checking is still done server-side, whether or not value checking is done with JavaScript. However, when JavaScript is enabled, users are required to enter correct values and are informed of their errors, as shown in Figure 3.

Data were solicited from a range of sources on the internet, as well as through QR code fliers around the city of Rochester. Pages linked to from the web form offer social networking buttons, allowing users to suggest to their contacts that they should also participate. In future research, soliciting participation from public figures with many followers could potentially dramatically boost the number of Twitter users who provide access to their tweets. Statistics about the collected data are discussed in section 6.

5.2 Demographic Data Pre-Processing

Before the collected user data could be used in analysis, each of the 10 values (section 5.1) had to be pre-processed to ensure that data values are consistent and useful. Username, birth year, month, gender, astrological sign, and email did not need any additional pre-processing. The web form and database backend ensures that each of those fields has a valid value. The username must be registered with Twitter, the birth year must be within the last 120 years, the email must be a valid

format, and the rest must be empty or contain one of the relevant pre-specified type values.

Occupation, education, languages, and regions needed special pre-processing for graphing and other analysis, since users are allowed to write in any value that conforms to the character level constraints. Unique values were most frequently included in education and region where users wrote phrases or sentences explaining their educational history or history of where they had lived. Additionally, each of these values needed to be generalized in order to be most useful. Some Twitter users included cities in their submitted regions, for example, and these were generalized to the respective state or country region. Highest educational level completed was generalized to general degree names (*e.g. Master's Degree*, *Doctoral Degree*), college, and high school. For data set redistribution, these values are left unaltered, as they were collected with future research in mind, and any pre-processing should be done with the full data set, in the context of the research being conducted. As most of the demographic data collected was not used in this work's experiments, limited analysis of the demographic data was done for this thesis to see how the collected data compares to other published statistics. Some of the results are shown in Figure 7 and are discussed in subsection 6.2.

5.3 Tweet Loading

As part of the web form submission and in order to download users' tweets, the users must be followed on Twitter. Following users as they participate via the web form and downloading their tweets is accomplished by using three programs developed as part of this work. All use Java and the Twitter4J library for interfacing with Twitter [50].

The first program is a library to handle authentication using the thesis Twitter account. Twitter uses an interactive OAuth authentication system [13], which requires authenticating applications to ask the program operator to visit a web address. Upon opening the web address, the operator is shown a pin to enter in the application, allowing it to authenticate with Twitter using the operator's Twitter account. Once the application has authenticated, it is possible to store the generated keys and avoid authenticating in the future. The library handles storage and retrieval of the keys, or generating them if they can not be found.

The second application handles following participating Twitter users and updating database fields to reflect follow status. It is run after a user submits the web form shown in Figure 1 and described in subsection 5.1. First, after a new participant's information is entered in the database, the application queries the database for users marked as new or pending. Next, it queries Twitter for users followed by the thesis account (this returns users as followed or pending). Using set logic, the application determines what users need to be followed on Twitter and follows them.

Feature Name	Example	Present Work	Gouws *et al.*
Single Character	(“see” → “c”)	1.0 %	29.1%
Word End	(“why” → “y”)	1.1 %	18.8%
Drop Vowels	(“should” → “shld”)	11.2 %	16.4%
Word Begin	(“schedule” → “sched”)	8.0 %	9.0%
You to U	(“your” → “ur”)	1.7 %	8.3%
Drop Last Char	(“saying” → “sayin”)	3.2 %	7.0%
Repeat Letter	(“food” → “fooooooood”)	3.0 %	5.5%
Contraction	(“birthday” → “b’day”)	70.7 %	5.0%
Th to D	(“this” → “dis”)	1.0 %	1.0%

Table 1: Feature type names and examples with a comparison of relative percentages found in the collected data set to the work of Gouws *et al.* and ordered by the frequencies found by Gouws *et al.* [17]. Percentages above reflect the percentage of abbreviation features identified that belong to each class (excluding unidentified abbreviation patterns).

With the same set logic, it also determines what users in the database need to have their followed status updated and executes the updates. The followed status can either be unfollowed (new users), pending (requires user approval to follow), or followed. This logic is used to reduce the number of Twitter API queries and to ensure that the following program can be run without any other checks.

The last application is run manually to load new tweets. Using Twitter4J, as many tweets as possible are downloaded from Twitter and stored in compressed files along with some metadata that allows tweet downloading to be continued later. Because the Twitter API only allows the most recent 3200 tweets, give or take a few, to be downloaded, the timestamps of the collected tweets will only go back so far. The oldest collectable tweet corresponds inversely to the rate at which a person tweets. For a person who only tweets about once a day, their tweets can be collected from 9 years ago (if Twitter were that old). The most prolific participant tweets about 15 times per day. As a result, that participant’s tweets can only be collected to around 7 months prior to the date of first collection.

5.4 Abbreviation Features and Extraction

In order to develop a model for predicting Twitter user age, the collected tweets are analyzed for abbreviation features. The abbreviation features used are those found by Gouws *et al.* to be most frequent in an overall Twitter corpus [17]. Those features are descriptively titled *single character*, *word end*, *drop vowels*, *word begin*, *you to u*, *drop last character*, *repeat letter*, *contraction*, and *th to d*. The usage frequency of abbreviation patterns found in the collected data set are compared to those found by Gouws *et al.* in Table 1. Features are described with more detail in Figure 4 and

are discussed in relation to the collected data set below.

The tweets collected using the Java framework described in subsection 5.2 were fed to a python framework for further analysis. The text of each tweet was given to the cleanser framework developed by Gouws *et al.*, which attempted to text-normalize each tweet into a standard English sentence. Different stages of the text normalization utilized functions from the python Natural Language Toolkit (NLTK) framework [33] and the SRI Language Modeling Toolkit (SRILM) [46]. The algorithm is outlined in Figure 5. (1) For each tweet, remove any series of punctuation determined to be emoticons, as well as HTML bracket artifacts, should they exist. A simple regular expression approach is taken to recognizing emoticons, as the problem is in itself quite difficult, as outlined by Bedrick [5]. Because of this difficulty, only a small subset of all emoticons could be correctly identified, so such features are not included in this work's analysis. (2) Tokenize each tweet into individual word tokens and punctuation. The NLTK *tokenize.punkt* library is used for tokenizing sentences, as it is effective at separating words and punctuation, as well as separating contraction words into multiple tokens, *e.g.* "shouldn't" $\rightarrow$ "should n't". (3) Generate substitution candidates for each OOV token using a string subsequence kernel [32]. Each candidate is paired with a probability used as an evaluation of the similarity to the original OOV token. Tokens that are not OOV are assigned a substitution candidate the same as the original token and a probability of 1. Probabilities are generated by the SRILM *ngram* program using n-gram language models based on Gouws *et al.*'s LA Times corpus [17] and Han and Baldwin's Twitter corpus [18]. (4) A word mesh (a confusion network that can be translated into a probabilistic finite-state grammar) is generated from the list of candidates and probabilities, which is given to the *lattice-tool* program of SRILM to decode into a most likely cleaned sentence, consisting of the candidates with the lowest expected word error. (5) The uncleaned original and tokenized texts are recorded, along with a list of pairs consisting of an OOV token and its generated substitution. Non-OOV tokens are retained as part of the tokenized text, but since they are not abbreviated, they are not recorded in the substitution pairs, as the pairs are used for abbreviation feature generation.

The abbreviation features are determined on a per-tweet level, based on a per-token analysis using the algorithm outlined in Figure 6. The input is the list of token and substitution pairs generated by the algorithm above. (1) For each tweet, each token and substitution pair are passed to an abbreviation-finding function. (2) The function applies a series of regular expressions and substring checks, which correspond to each of the defined abbreviation features. (3) Each token pair is thereby assigned an abbreviation feature classification representing which abbreviation type it matches. (4) The tweet's set of token abbreviation feature classifications are consolidated into a single percentage feature vector for each tweet. The values in the vector reflect the percentage

Single Character

Replace a word with a single character. This is often a phonemic transliteration, such as "see" → "c" or "to" → "2". This is overridden by other feature classifications.

Word End

Drop all characters except a substring at the end of the identified replacement word, such as in "why" → "y" or "them" → "em".

Drop Vowels

Vowels make up approximately 38% to 40% of English words [2]. Dropping vowels is often used to shorten words. This feature is defined as elision of one or more orthographic vowels in a word, *e.g.* "could" → "cld" or "interesting" → "intersting" (elision of a single 'e').

Word Begin

Drop all characters except a substring at the beginning of the identified replacement word, such as in "undergraduate" → "undergrad" or "schedule" → "sched".

You to U

"You" is sometimes abbreviated as "u" in various pronoun-based words, such as you, your, you're, etc.

Drop Last Character

The last character can often be omitted without affecting the reading of a word, as in the gerundive (-ing) case in English as in "saying" → "sayin" or in "what" → "wha".

Repeat Letter

Sometimes letters are repeated rather than omitted. This can be to communicate emphasis or emotion, such as in "food" → "fooooooood" or "amazing" → "amaaaaazinggg".

Contraction

A traditional space saving method of abbreviating two words as one. Due to the difficulty of detecting compounds that do not utilize an apostrophe, this feature only describes words that are contractions using an apostrophe, such as nonstandard "breakfast" → "b'fast" or standard "could not" → "couldn't".

Th to D

Some words are perceived similarly when substituting the letter "d" for a "th". This feature could include "the" → "da", as a special case, but this was not considered, as it did not appear in the collected data set. Some cases that did appear in the data set include "that" → "dat" and "this" → "dis".

Figure 4: Description of the nine abbreviation features from Gouws *et al.*. Each word pair was assigned one feature type classification. Some feature types overlap, such as *drop last character* and *word begin*. In these cases, the more specific classification was assigned (*drop last character*).

```
Data: tweet text
Result: normalized tweet text
begin generate sentence candidates
    Remove emoticons and HTML artifacts
    tokens ← Tokenize sentence using NLTK + customization
    candidates ← foreach token in tokens do Generate substitution candidates and
    probabilities
        if OOV_but_valid(token) then
            return token, 1.0
        end
        return list of substitution candidates and probabilities for token
    end
    lattice ← generate confusion network for candidates
    replacements ← generate lowest word error sentence from lattice
    return replacements
end
```

Figure 5: Text cleanser algorithm provided by Gouws *et al.* [17]. This work added some customization in tokenization and small fixes, but otherwise the algorithm is the same.

of tokens in the tweet which utilize each abbreviation type. (5) The percentages are further generalized to a boolean vector, which describes if a given abbreviation feature type was used at all in a tweet. Equivalent experiments were run using the percentage vectors and the boolean vectors and compared. Studies have found that enough information can often be found in a single tweet to do effective binary classification, such as on gender [6]. By comparing the results of classification using the percentage and boolean vectors, it can be determined how much abbreviation feature information is necessary for a good classification. Additionally, combining boolean and percentage features with word n-gram features, as well as best first feature selection or principal component analysis feature extraction, was found to further improve classification results.

6 Data Set Analysis

A total of 72 Twitter users supplied their demographic information. Of those 72 participants, 66 had tweets. This is in part due to several users having no tweets, and some who disappeared from Twitter after submitting the web form data. This means 8% of participating users have no tweets. According to Beevolve Technologies, as many as 25% of users have never tweeted [48]. In the present data set, the average number of tweets submitted by a user is 1538. Beevolve Technologies presents separate figures for tweet frequencies based on gender, which, when combined, give an

```
Data: tokenized tweet array
Result: abbreviation feature vectors (percents and booleans)
begin
    length ← len(tweet array)
    counts ← vector(0,10)
    foreach token pair in tweet array do
        token ← token pair[0]
        replacement ← token pair[1]
        begin get abbreviations for token pair
            if token = replacement.replace("you", "u") then
                type ← "you to u"
            else if token = replacement.replace("aeiou", "") or token =
            replacement.replace("aeiouy", "")
            then
                type ← "drop vowels"
            else if token = replacement.substr(0, len(replacement) - 1) then
                type ← "drop last character"
            else if replacement = de_repeat(token) then
                type ← "repeat letter"
            else if token = replacement.endsWith(token) then
                type ← "word end"
            else if token = replacement.bullettsWith(token) then
                type ← "word begin"
            else if is_contraction(token, replacement) then
                type ← "contraction"
            else if token = replacement.replace("th", "d") then
                type ← "th to d"
            else if token in replacement and len(token) = 1 then
                type ← "single character"
            else
                type ← "unknown"
            end
        end
        increment counts for type
    end
    percents ← counts/length
    booleans ← counts > 0
end
```

Figure 6: Abbreviation feature assignment algorithm. The classifications are assigned such that features which are subsets of other features are assigned first. Some more specific features are subsets of other features. For example, any *drop last character* feature is also a *word begin* feature.

Total Participants	72
Participants With Tweets	66
Tweets	101,496
Tokens	1,655,326
Word Tokens	1,417,968
Punctuation Tokens	237,358

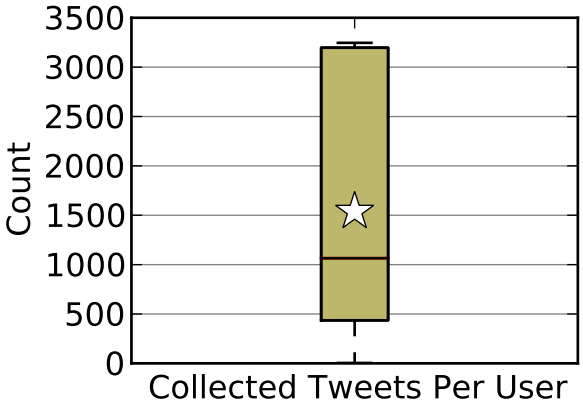
Table 2: Basic information about the present data set

average for tweets per person of 590, much lower than in the collected data set [48]. Basic descriptive data set information is shown in Table 2. From the participants, over a hundred thousand tweets were collected, comprising 1.65 million tokens. Of those tokens, about 1.4 million were words and around two hundred thousand were punctuation. More detailed information about the collected tweets from the 66 users that had tweets follows in subsection 6.1. Information about the demographic data collected as part of the data set (all 72 users) is in subsection 6.2.

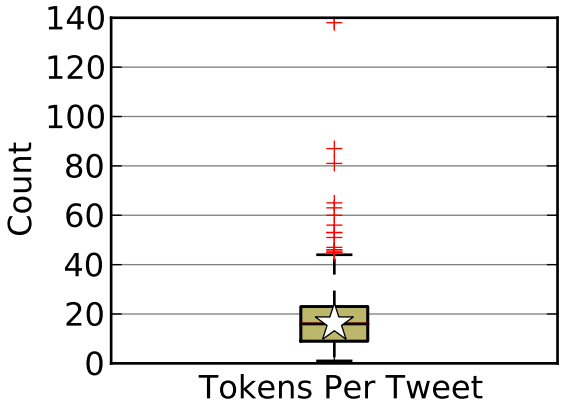
6.1 Tweet Information

The number of tweets collected for participating users varies widely, as shown in Figure 7a. As discussed in subsection 5.2, only about the most recent 3200 tweets can be collected from a user. The average of 1538 tweets collected per user and median of 1065 suggest that for as many users for whom the number of tweets downloaded reached the maximum number (3200), there were equally as many who had published under a thousand tweets. Additionally, several users only contributed a few tweets, and a few contributed none.

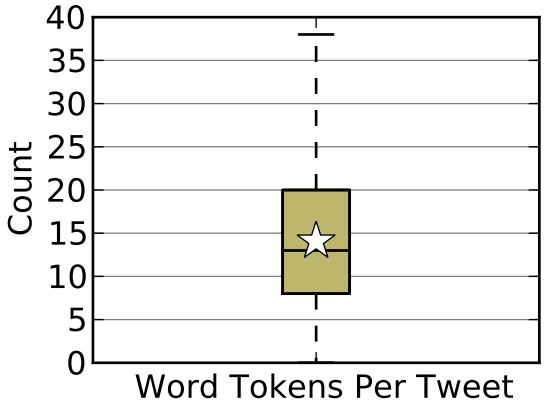
On the tweet level, most tweets had a token count in the teens or low twenties, as shown in Figure 7b. This token count includes both word tokens and punctuation tokens. A punctuation token includes standard clausal punctuation, such as commas and periods, as well as emoticons and other only-punctuation elements in tweets. Additionally, the token count is based on tokenization by the NLTK punkt tokenizer, which splits contractions into two word tokens, so a space-based token count would have been a bit lower. Word tokens are the set of unigram tokens left over when the punctuation tokens are removed. As indicated by Figures 7b and 7c, an average tweet might be a single sentence, with two punctuation marks and 14 word tokens. The most extreme outlier consists of a three character interjection (“YAY”), followed by 137 exclamation points. The tokenizer splits standard punctuation, so this greatly increased the token count by splitting each of the 137 exclamation marks into its own token. The amount of punctuation tokens in a tweet is generally below 5, suggesting that users avoid punctuation, except where necessary. Token counts



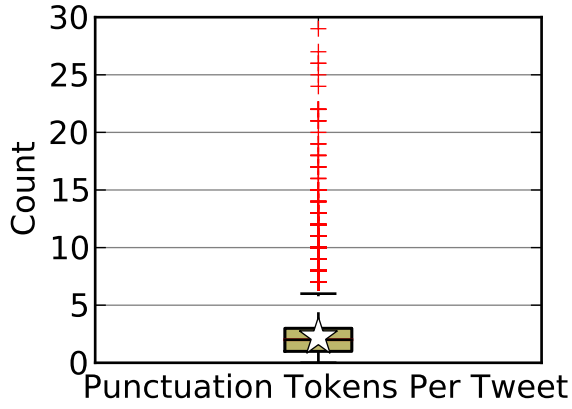
(a) **Average**: 1538
Median: 1065



(b) **Average**: 16.3
Median: 16



(c) **Average**: 14
Median: 13



(d) **Average**: 2.24
Median: 2

Figure 7: Tweet and token distributions. Stars mark the average values. The bar in the middle of the box marks the median value (50th percentile), and the box extends to the edge of the 25th and 75th percentiles. Tokens is a count of the total tokens of a tweet, as defined in subsection 6.1. Word token counts exclude tokens made up entirely of punctuation characters. The top 4 points (137, 80, 48, and 47) are omitted from Figure 7d for readability.

Type	Min	Max	Mean
All Tokens	1	138	16.3
Words	1	35	14.6
Punctuation	1	137	2.6

Table 3: Per-tweet minimum, maximum, and mean counts for types of tweet tokens

		All Tokens
Minimum	(1)	Glargleargleblaaaaarq[a]
Maximum	(138)	YAY!!!**[134 more !]**
Mean	(16)	I called for a shuttle half an hour ago. Wtf. I am cold.
		Word Tokens
Minimum	(1)	Nononono.[a]
Maximum	(35)	So if I do the pinch hit for Tari I'll have about 5500 words due in Nov - TWRB is 3k, pinch hit is 1k, Avenger fest is 1k or 1.5 and K/S
Mean	(14)	Father daughter swim: 9-yr old Claire swam 2050 yds in 1 hr, congrats!
		Punctuation Tokens
Minimum	(1)	Today I learned my bathroom door knob is broken and if I close the door I can't get out and have to use the emergency call button. -_-[b]
Maximum	(137)	YAY!!!**[134 more !]**
Mean	(2)	I feel that I've been lured into a trap. There will be no cake.

Table 4: Tweet examples from the collected data set. Since actual counts are integer value, and means are floating point, examples below include floor values of the mean.

[a]Made up words and emotionally illustrative utterances can be difficult to effectively pair with a cleaned representation and subsequently label with features, as there is generally no equivalent pair already encountered for use by the normalization algorithm.

[b]Terminating hashtags, username @-references, and emoticons are not counted toward punctuation or word token counts. Aphostrophes are included as part of contraction word tokens, while clause-separating punction is included in punctuation counts.

Feature Name	Difference between Gouws *et al.* and present work
Single Character	28.1 %
Word End	17.7 %
Drop Vowels	5.2 %
Word Begin	1.0 %
You to U	6.6 %
Drop Last Char	3.8 %
Repeat Letter	2.5 %
Contraction	**−65.7 %**
Th to D	0.0 %

Table 5: Feature names and the difference of relative percentages found in the collected data set and the work of Gouws *et al.* [17]. Differences shown are equivalent to subtracting the present work's results from those of Gouws *et al.* in Table 1. A notable difference is for the Contraction feature (bolded), which accounted for a larger percentage of the detected word and phrase abbreviations in the present work than in the work of Gouws *et al.*. In all other cases, the percentages in the present work were lower than those in the work of Gouws *et al.*.

and examples of average tweets and outliers are shown in Table 3 and Table 4, respectively.

As shown in Table 5, the distribution of abbreviation pattern features in the collected data set is very different from those reported by Gouws *et al.* [17]. While Gouws reported 90% coverage with the 9 defined abbreviation types with a large Twitter data set, those types only cover 43% of the found abbreviation patterns in this data set. There are several reasons for this that could be contributing factors, most notably that the algorithmic definition of the abbreviation patterns may not have been consistent between the work of Gouws *et al.* and this work. Additionally, the data set primarily captures people who have completed some level of college (see subsection 6.2 and Figure 8c). These more educated persons appear at a higher rate in the collected data set than other studies have indicated [44]. Many collected tweets are written in mostly standard English with standard English syntax. Newer slang and various context-specific tokens, which may be considered standard to a human reading or writing the collected tweets, would not have shown up in the LA Times corpus or Han and Baldwin's tweet corpus used to train the sentence normalizer used in this study. As such, many tokens are replaced with unnecessary substitutions, and decoding the lattice into a normalized sentence will augment tokens around any that were considered OOV. This creates anomalous abbreviation patterns that do not fit into the defined categories at a higher rate than a more general Twitter corpus, such as that used by Gouws *et al.*, in which users utilize word and phrase abbreviations more frequently.

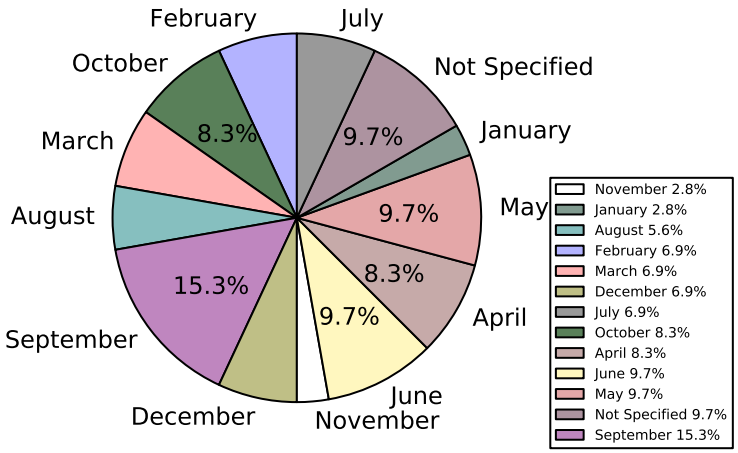
6.2 Demographic Information

The collected user demographic information, while from a small set of users, presents an interesting picture of Twitter users that is consistent with some published figures. In the collected data set, users self-labeled as 51.4% female and 40.3% male (see Figure 8d). If we divide those who withheld gender information evenly between the two groups, we get about 55.6% female, 44.4% male. This is quite consistent with various demographic studies, which suggest that Twitter has a slightly higher female user base than male. The Pew Research Institute reported Twitter users as 53.5% female, 46.5% male [44], while Pingdom reported a higher rate of female usage at 60% female, 40% male [1].

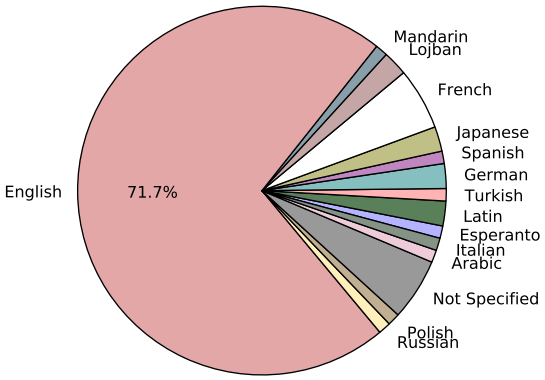
The month data collected is similarly consistent with other general demographic data. It has been established that there are more births in the months of September and August. A random sampling of a population statistically would yield a larger number of people born in those months than others [36]. This is reflected reasonably in the collected data, shown in Figure 8a. 15.3% of participants self-labeled as being born in September. Additionally, as is common, fewer people were born November through January. Those months encompass a total of 12.5% of the participants (90.3% of whom provided their month of birth). While the dates associated are not identical, a similar effect is seen in the reported astrological signs (see Figure 8e). Virgo (mostly consists of birth dates in the month of September) comprises 12.5% of participants, and Sagittarius, Scorpio, and Capricorn (those closest to the months of November through January) are a total of 9.7%. Just over 70% of people labeled their astrological sign.

Participants were also allowed to specify their highest level of education completed. After preprocessing the variety of responses, it becomes evident that the collected data set consists mostly of educated persons. 82% of participants reported that they had completed some or all of a college stay. 7% reported being in high school, and 11% did not provide education information. According to the Pew Research Institute, 6.25% of Twitter users over 18 have not completed high school and 26.9% have completed high school, but no college [44], which reflects a much higher rate of lower education than that reported in the collected data set. The Pew Research Institute also stated that 25.9% of users were in college, while 40.2% had completed college [44]. In comparison to the collected data set, 22% were in college and 60% had completed college. Again, the amount of educational preparation of the users in the data set is generally higher and more homogeneous than a general populous on Twitter. This could have various effects on the writing styles observed for these users. As noted in subsection 6.1, this could be a contributing reason for the lower usage of the word and phrase abbreviations that this research uses as features.

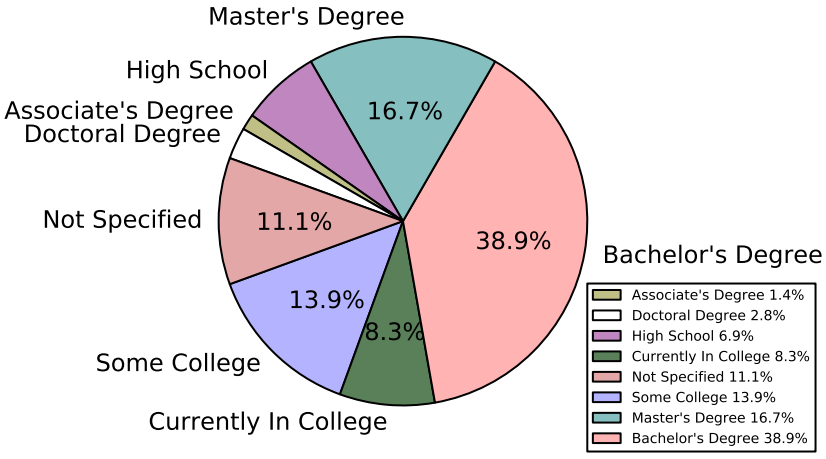
Two of the collected demographic items allowed multiple values to be submitted: languages



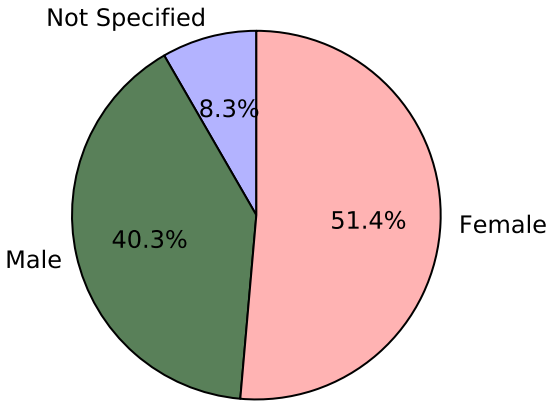
(a) Birth Months



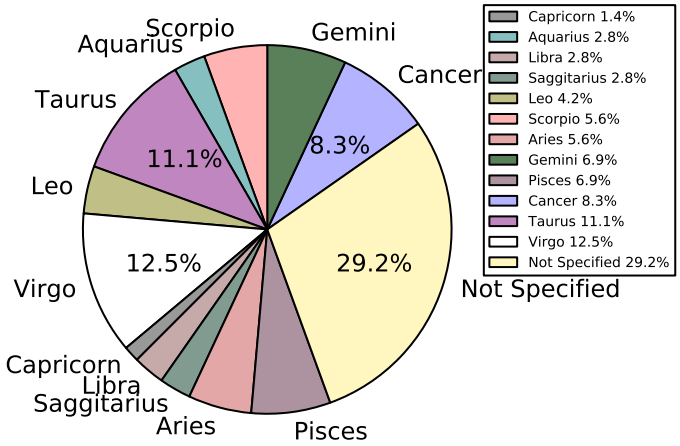
(b) Languages Used Most on Twitter



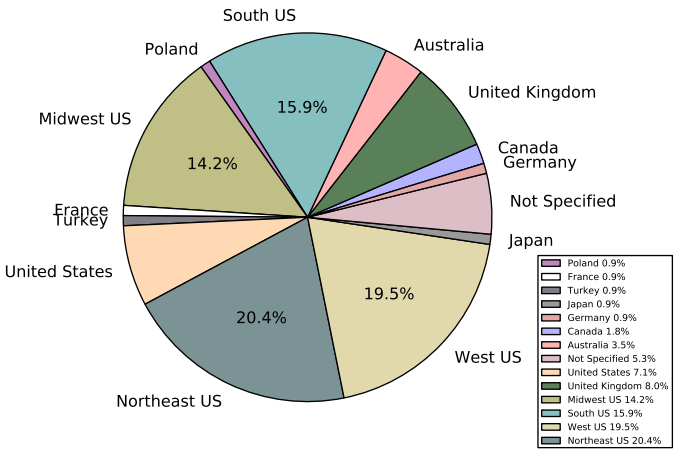
(c) Highest Education



(d) Genders

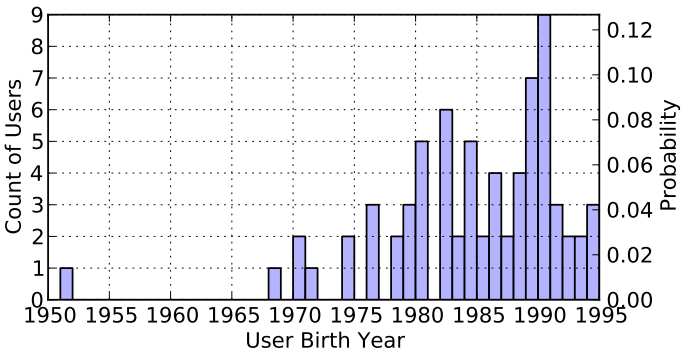


(e) Astrological Signs



(f) Geographical Regions of Residence

Figure 8: User demographic data specified for the data set. Data can be found in Appendix A in tabular form.



(a) Distribution of participant birth years.

Ages	Participants
≤ 24	43.1%
25-34	43.1%
35-44	12.5%
45-54	0.0%
55-64	1.3%
65+	0.0%

(b) Participant ages within discrete ranges consistent with those used by other Twitter demographic research.

Figure 9: Reported birth years of participants and age ranges at time of publication

most used on Twitter, and regions of residence. Users were also allowed to submit any value to these entries, if none of the suggested values were adequate. In spite of this, user-specified languages did not need any additional cleaning. The regions of residence submitted did require additional cleaning, however. Several users were more specific with their region submissions, listing their city of residence. Some other users submitted a phrase describing their residence situation or history. In Figure 8f, these values are adjusted to country or US region. The majority of participants were from the US (77% of total participants; 5% did not specify any region), so the states specified within the US were separated into five geographical regions, plus *United States* for those who only submitted the country name. Languages much more frequently received multiple values, and in a greater number than regions of residence. 71.7% of languages reported were English. 91.7% of users reported using at least English, and 6.9% did not specify any language. Frequently, those who did not specify a language did not specify a region of residence or other information. Only 1% did not specify English in their languages used. The user that did not specify English was withheld from analysis, while those users that did not specify any language were verified manually as using English.

The focal point of the data collection was participant birth years. The collected data are shown in two forms in Figure 9. Other research that targets Twitter user age often treats age as a binary classification, while much published demographic research uses age range bins of ten years. The collected data show a similar distribution to these demographics with a higher usage rate in younger people, with the usage rate tapering off for more aged persons. Almost 90% of participants were under 35, which is a higher rate than many demographic publications on Twitter users. Pingdom reported about 45% of Twitter users were under 35, with another 45% aged 35 to 54. The last 10% were twice more likely to be under 65 as over [1]. By comparison, 13% and 1% of this study's

N-gram	N	Frequency
!	1	0.998
! !	2	0.562
thank	1	0.335
thank you	2	0.258

N-gram	N	Frequency
, too	2	0.187
, too .	3	0.115
ur	1	0.103

Table 6: N-grams generated and their frequencies. The frequencies reflect those found in data sets with 100 tweets per instance. N-gram elements are space separated. Very few trigrams were selected, as their frequency was very low. Of those that were selected, most include some form of punctuation, such as *, too .* shown above.

participants were in the 35 to 54 and the 55 and over categories, respectively. It is difficult to do comparisons with much published work, however, due to the differences in sampling and reporting methods, but it becomes relatively clear that the collected data set is thus far heavily influenced by younger Twitter users. This is similar to the data reflected by Beevolve, which collected user ages when they were defined in a user's profile on Twitter textually. Beevolve found, of those users that self-report their age on their Twitter profile (about 0.45%), almost 90% are under 35. Of those, 73.3% are aged 15-25 [48]. This suggests that younger users are much more willing to divulge their age, which could account for the prevalence of young participants in this data set, as revealing age was mandatory for participation. These biases are part of what makes age prediction such a difficult problem.

7 Experiments

The experiments were run on several hundred data sets with a number of classifiers and different combinations of abbreviation features and n-gram features. The base feature types were (1) boolean and (2) percentage vectors, as explained in subsection 5.4. Additionally, (3) n-gram feature vectors were created from the tokenized versions of each tweet, using Weka's n-gram tokenizer. Since the tokenizing was already done as part of the pre-processing, the tokenizer simply separated tokens by spaces, and additional parts of traditional n-gram processing, such as stop lists, were not used. All n-gram tokens were processed in lower case, to avoid capitalization overlaps. A Lovins stemmer [34] was applied as part of the process, since many words can appear with many derivational and inflectional suffixes. The most frequent n-grams (up to an n of 3) were selected automatically by Weka, resulting in about 1200 text features. In experiments using n-grams, feature selection or extraction was run to reduce the number of features, often to around 400 or less. Each n-gram feature was considered as a numerically represented boolean in each instance, one

Bins	Age Range
2	$<= 25, <= 61$
4	$<= 22, <= 25, <= 30, <= 61$
6	$<= 20, <= 22, <= 25, <= 28, <= 32, <= 61$
8	$<= 20, <= 22, <= 23, <= 25, <= 28, <= 30, <= 33, <= 61$
10	$<= 19, <= 21, <= 22, <= 23, <= 25, <= 27, <= 28, <= 30, <= 33, <= 61$

Table 7: Age values covered by equal size classification bins. Bin time ranges were generated so that the number of instances were as equal as possible between bins. Instances are assigned a class based on the age of the user (in years) at the time of writing the tweet or tweets represented by the instance's feature data.

if present, zero if absent. Some selected n-grams are shown in Table 6. These three feature types were used in experiments individually, as well as in three combinations of two types and one of all three types.

The data were comprised of a bit over a hundred thousand instances (one per tweet), which were combined in several ways for analysis. A group of data sets was created with one tweet per instance, as well as with groups of 25, 50, 75, and 100 tweets. For example, in the 100 tweet grouping, 100 tweets from the same user in chronological order were grouped and treated as one instance and analyzed for the three types of feature vectors. Extra tweets that did not fill a full group were ignored for those data sets, so that each instance was comprised of a full 100 tweets. This reduced the total number of instances in the data set (around 1000 instances for 100 tweets per group), while increasing the feature information contained in each instance. Initial experiments were run on several tweet groupings in order to determine an optimal amount of data per instance. Later experiments restricted the data sets to one tweet per instance, a group with 75 tweets per instance, and three groups with 100 tweets per instance. Experiments performed to determine optimal grouping sizes are discussed in subsubsection 7.1.2.

About 30 thousand of the tweets (30%) did not have any recorded word and phrase abbreviation features, as they were written in standard English. For this reason, each group of data sets were duplicated and filtered to only include tweets which exhibited at least one feature type. This division was done on a per-tweet level, so when grouped, all tweets in a group would have had at least one feature type alone. This generally improved classifier accuracy, except in the case of the n-gram feature type. Since the division was based on the boolean and percentage vectors, there was generally little or no improvement in n-gram based analyses between the filtered and unfiltered groups.

Using the age data provided by the study participants, and the timestamps of their collected tweets, each tweet was assigned an age. For grouped instances, the assigned age was the average

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$FMeasure = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{4}$$

Figure 10: Metrics used in evaluation of classifiers. TP, TN, FP, and FN represent the number of true positive, true negative, false positive, and false negative results respectively.

age of all contained tweets. Using the assigned age, each instance was assigned a classification bin. Instances were grouped into 2, 4, 6, 8, and 10 bins to create separate data sets for initial experiments. The high baseline accuracy for lower numbers of bins resulted in lower accuracy improvements compared to that of data sets with higher numbers of bins, so data sets binning by 2 and 4 were later ignored. In later experiments, 10 bins were used for single-tweet-per-instance and 75-tweet-per-instance data sets. In the data sets with groups of 100 tweets per instance, data sets were created using 6, 8, and 10 bins. Single-tweet data sets were selected for comparison, and the larger groupings and associated binnings were selected based on the best accuracy gain results in initial experimentation.

Initially, experiments were run on two bin types: equal-width bins, in which each bin covered an equal time span; and equal-size bins, in which each bin covered roughly the same number of instances. Equal-width bins were associated with low accuracy gains compared to equal-size bins and were excluded from later experiments. Initial experiments that led to these bin types are discussed in subsubsection 7.1.3. The age ranges covered by equal-size bins is shown in Table 7.

In later experiments, each data set was run through Weka's best first feature selection algorithm, as well as Weka's Principal Component Analysis (PCA) algorithm for feature extraction. When dealing with n-gram features alone or in any combination with abbreviation features, feature selection and extraction were necessary to keep the feature set a manageable size and ensure classifier run time did not become unfeasible. Additionally, such filtering was found in many cases to improve results over unfiltered, raw features. The data sets with filtered features were run through experiments equivalent to those run on the raw feature data set. One of the best cases, running an SVM on numeric features, saw an improvement of 34% when PCA was used compared to without (bolded in Table 9).

Each group of experiments was run on a similar set of classifiers, using the Weka Experimenter framework, and analyzed using t-tests with a confidence of 0.05 to ensure the results selected for note were statistically better than baseline by more than just random chance. Each of the classifiers was evaluated using accuracy, precision, recall, and F-measure as defined in Equation 1, Equation 2, Equation 3, and Equation 4, respectively, in Figure 10. The F-measure is defined as the harmonic mean of precision and recall, so it provides a good comparison point that combines precision and recall.

Each data set was run through a set of basic classifiers and a set of more advanced learning algorithms, through Weka. Additionally, some experiments were run through an apriori association mining algorithm in Weka. All classifier experiments began with a ZeroR classifier as an absolute baseline. The ZeroR classifier simply assigns the most common class to all instances. Association mining results are reported with their confidence.

The basic classifier set included several rule-based classifiers: a OneR classifier, which uses a single feature which has the most statistical likelihood to be a good predictor to decide which class to assign each instance, described by Holte [20]. The OneR classifier selected various features, depending on the data set, but chose to use the contraction feature most often, perhaps due to its frequency and variance of occurrence. This was followed with a Naive Bayesian classifier, which uses probabilistic mappings of features to classes, assuming statistically independent features, described by John [28]. Next, three variations of a J48 (C4.5 decision tree) classifier were run: a standard run, which uses subtree raising in its pruning; a run with reduced error pruning, which uses a held-out set for validation and prunes nodes based on results with those instances; and a completely unpruned run [38]. Initial experiments also included a Decision Table classifier, which utilizes a best-first attribute mapping, described by Kohavi [30]. After initial experiments to determine optimal parameters, the three J48 classifiers were replaced with two, using subtree raising for pruning and confidence factors of 0.185 and 0.2, and the Decision Table classifier was dropped from further experiments.

The more advanced classifiers began with Support Vector Machine (SVM) classifiers, provided by the LibSVM framework [8]. The framework provides five types of SVM classifiers, but only two support non-binary classifications: C-SVC and nu-SVC, cost-based support vector classification implementations [43]. In C-SVC, the cost parameter is unconstrained and positive, while in nu-SVC, the cost parameter is in the range [0,1]. Next, experiments were run through a multilayer perceptron neural network (MPN) classifier implemented in Weka. The advanced classifiers were reduced to one C-SVC and one neural network classifier after initial experiments to determine optimal parameters. Experiments to determine optimal classifier parameters are discussed in sub-

subsection 7.1.1, and experiments to determine optimal data set parameters are discussed in other subsections of subsection 7.1.

Initial experiments were run using an 80/20 train/test split. For each run, Weka randomly selected 80% of the data set instances to be used train a classifier, and the remaining 20% was used to validate the classifier and obtain accuracy metrics. For the basic classifier set, each classifier and data set pair were run with random splits 10 times. The longer running advanced classifiers were run three times each. The results from each run were averaged for analysis. These initial experiments are discussed in subsection 7.1.

Each of the later experiments, described in subsection 7.2 and its subsections, was run using Weka's experimenter and 10-fold cross validation. In Weka's 10-fold cross validation, Weka randomly divides the input data set into 10 non-overlapping groups (folds). The classifier algorithm is trained on 9 of the 10 folds, and tested using the one remaining fold. The training and testing is repeated such that each of the 10 folds is used as a test set once. Each classifier-data set pair was run through cross validation three times and the results were averaged for analysis.

One set of experiments in subsection 7.2 does not follow this paradigm: the withheld users experiments, described in subsubsection 7.2.8. Due to the nature of the experiment and limitations of Weka's experimenter, a manual single-fold experiment was run. Classifiers were trained on a data set with all instances from 5 or 6 users removed (about 10%), then tested using the users' instances that were withheld from training.

Association mining experiments, described in subsection 7.3, were run next. The previously used data sets were augmented to include user demographic data, as well as all abbreviation and n-gram features. These data sets were run through Weka's apriori association mining algorithm with various parameters to isolate interesting rule sets.

Lastly, longitudinal analysis was run on a single-tweet-per-instance data set with numeric abbreviation features, as described in subsection 7.4. The features were analyzed for usage versus time spent using Twitter to see if changes abbreviation usage could be observed.

The following sections discuss the variety of experiments run and notable results. The experiments run to determine optimal binning, grouping, and classifier parameters are discussed in subsection 7.1. Based on the results of those experiments, further experiments were run in an attempt to improve results, discussed in subsection 7.2. The results of association mining algorithms on selected data sets are described in subsection 7.3. Lastly, some observations regarding longitudinal abbreviation pattern usage changes are discussed in subsection 7.4.

7.1 Initial Pilot Experiments

A series of pilot experiments were run to determine optimal configurations for further experimentation. In this section are descriptions of those experiments, their results, and conclusions carried through to later experiments. First, experiments were run to determine optimal parameters for the variety of classifiers. These experiments are discussed in subsubsection 7.1.1. In subsubsection 7.1.2, experiments to determine the optimal grouping of tweets per instance are discussed. Lastly, experiments to determine the best number of bins for each grouping are discussed in subsubsection 7.1.3.

In general, all types of classifiers performed better using a filtered data set than an equivalent full data set. This can be intuitively explained by the properties of a full data set. There are about 30 thousand more instances to train and test with in a full data set (one tweet per instance–numbers differ for the grouped data sets), but those 30 thousand do not exhibit any abbreviation features, which makes correctly classifying those instances much more difficult than when only training and testing with instances that show abbreviation feature use.

The last parameter investigated in the initial experiments was the type of feature vector used: boolean or percentage. The effect of this parameter varied depending on the classifier being used. Basic classifiers had better accuracy and precision when using percentage vector features. Recall varied from slightly better to much worse, which reflects in the F-Measure, which was generally slightly better for the boolean features. SVM classifiers performed much better in all respects when using boolean features. The multilayer perceptron network (MPN) classifiers performed better in all metrics when using numeric features.

7.1.1 Parameter Selection Experiments

Initial experiments were only run on data sets with boolean and percentage feature vectors, as described in subsection 5.4. In order to determine optimal parameters, each data set was run through several instantiations of each classifier with adjustments to various classifier parameters. The results were examined to determine which parameters resulted in the best accuracy and F-measure, compared to each other and to the ZeroR baseline.

The OneR classifier's minimum bucket size parameter, which defaults to 6 in Weka, had little or no effect on its results, so it was left at 6. The naive Bayesian classifier had one parameter: whether to use a kernel estimator or a normal distribution in its calculations. The default normal distribution method was found to have better results. Between the three types of J48 trees, a variety of parameters associated with error pruning were experimented with. The optimal results were split

between two confidence factors (a parameter that determines how much pruning is incurred: lower means more pruning) in one pruning type. The two selected J48 configurations used default subtree raising for pruning and confidence factors of 0.2 and 0.185. The Decision Table classifier was also experimented on, but never outperformed the other classifiers and was not considered further due to the time it took to run.

Several series of SVM classifier runs were executed, and one configuration was selected. The C-SVC classifier was selected with *coef0* equal to 0, cost equal to 1, a degree 3 kernel with a radial bias function, 0 gamma, 0.1 loss, and 0.5 nu. Similarly, a series of multilayer perceptron network classifications were run with a variety of parameters. The best performing parameters selected were no learning rate decay, one hidden layer with a number of nodes equal to the number of attributes plus the number of bins, a learning rate of 0.3, a momentum of 0.25, and a training time of 500 iterations.

The selected parameters were used throughout all following experimentation.

7.1.2 Grouping Experiments

The next step in initial experimentation was to determine optimal grouping for the number of tweets per data instance. With all other parameters kept constant (generally those selected above), the best performing classifications came from instances with 100 tweets grouped per instance. The difference between grouping 25, 50, 75, and 100 tweets were slim, but they all performed somewhat better than those data sets with one tweet per instance. When grouping tweets, all but four users generated at least one instance. A majority of users produced at least four instances, even when grouping 100 tweets per instance.

Because of these observations, additional experiments were run testing 125, 150, and 175 tweets grouped per instance. With boolean features, the larger groups offered no improvement in classification results. Numeric features saw slightly improved results with groups of 125, but not with 150 and 175. Additionally, experiments were run in which all of a user's tweets were combined into a single instance. Those experiments did not perform well. Because of this, the larger groupings were not used.

A parameter of interest is the minimum amount of data required for an accurate result. Because of this, and due to the results of experiments with 25, 50, 75, and 100 tweets per instance, groups of 1, 75, and 100 tweets per instance were selected for later experimentation. These were combined with optimal binning as determined by the experiments discussed below.

7.1.3 Binning Experiments

After running the above experiments, it became obvious that certain data sets had much better accuracy, precision, and recall than others. The equal-sized bins (number of instances per bins) almost always outperformed relative baselines more than the equal-width bins (time span covered by the bins). Additionally, data sets with a larger number of bins outperformed relative baselines with a larger margin than those with fewer bins. However, increasing the number of bins lowered the absolute values for accuracy, precision, recall, and F-Measure.

Based on these results, data set groups with 6, 8, and 10 bins were selected for further experimentation, combined with the other data set and classifier parameters discussed above. The specifics of these data sets and results of experiments using them are discussed in subsection 7.2.

7.2 Selected Data Set Experiments

As described in section 7, data sets were created in groups with several parameters. In each feature-based group of data sets are five combinations of tweet-per-instance grouping and binning: 1 tweet per group, 10 bins; 75 tweets per group, 10 bins; 100 tweets per group, 10 bins; 100 tweets per group, 8 bins; and 100 tweets per group, 6 bins. Each of these combinations occurs in two data sets: one with all tweets represented, and one with only tweets that exhibit at least one abbreviation feature represented. All binning was equal-size (roughly equivalent number of instances per bin).

Several sets of experiments were run on each of seven data set groups. The seven groups were created with different combinations of the boolean, numeric, and n-gram features (each of the three alone, three combinations of two, and one of all three). Additionally, all groups contained raw features, features selected with best-first feature selection, or features derived from PCA feature extraction. Experiments on feature combinations that did not have n-gram type features included sets with just raw features. However, the number of features created by n-gram analysis was very large, so experiments on feature mixtures that did include n-gram type features were limited to those reduced by best-first selection or PCA.

It is worth noting the bin selections for the experiments. It is somewhat counter-intuitive that a larger number of bins would lead to increased accuracy in classifiers. Two bins should have a baseline accuracy of 50%, four bins should have a baseline of 25%, and so on to a baseline accuracy of 10% for 10 bins. This assumes a perfectly equal number of instances per bin. While the best attempt was made at bin equality, the relatively small size of the data set made them slightly unbalanced. For example, with 10 bins a maximal baseline accuracy was 17%.

When only examining classifier accuracy, without comparing it to the relative baseline, classi-

fication into two bins will likely have a higher accuracy than classification into 10 bins. However, these experiments were analyzed for improvements over baseline. Two factors contribute to the higher performance of a larger number of bins in this analysis. First, there is less possible improvement over a 50% baseline than over a 17% baseline. This matters when the largest improvement made was 67%. Second is the issue of data outliers. In a two-bin classification, outlyers (some extreme, such as the 1951 user) will be grouped in with the rest of the instances in a bin. A four bin classification will overcome this problem somewhat. In this case, the best division of instances to bins was in 10, 8, and 6 bins.

Each set of experiments is briefly described in the following sections: boolean features in subsubsection 7.2.1; numeric features in subsubsection 7.2.2; n-gram features in subsubsection 7.2.3; boolean with numeric features in subsubsection 7.2.4; boolean with n-gram features in subsubsection 7.2.5; numeric with n-gram features in subsubsection 7.2.6; and all three feature types in subsubsection 7.2.7. Based on the results of the boolean feature experiments and the numeric feature experiments, an additional set of experiments were run to determine efficacy of trained classifiers when run against novel data from withheld users. These experiments are discussed in subsubsection 7.2.8.

7.2.1 Boolean Feature Experiments

Classifiers trained on boolean features were found to have less improvement over baseline than numeric features, or any combination of features involving n-gram features. The results of experiments on these features are shown in Table 8. The best performing classifier, both in accuracy, and in run time on boolean features was the chosen SVM run on PCA extracted features (75 tweets per group and 10 bins). It achieved an accuracy of 9% over the ZeroR baseline. Its accuracy was closely followed by that of the J48 classifier on unmodified boolean features of the same data set, as well as the Naive Bayes classifier on a best-first feature set with 100 tweets per group and 10 bins. While the boolean features were shown to be somewhat useful, the results could definitely be improved on by combining them with other features. Such experiments are described in subsubsection 7.2.5, subsubsection 7.2.4, and subsubsection 7.2.7.

7.2.2 Numeric Feature Experiments

Numeric feature experiments showed notable accuracy improvements over baseline, as shown in Table 9. The best improvement was shown in data sets with 8 bins and 100 tweets per instance. The J48 tree classifier performed well with best-first selected features at 25% above relative baseline, well outperforming all other classifiers using best-first features, except the MPN. The MPN did

Bins	6	8	10	10	10
Tweets per Instance	100	100	100	75	1
ZeroR (Raw features)	20.61 %	17.10 %	16.63 %	15.80 %	13.78 %
OneR	22.33 %	15.85 %	16.63 %	17.29 %	14.65 %
Naive Bayes	22.34 %	24.03 %	19.99 %	22.34 %	15.05 %
J48 Tree	19.60 %	22.47 %	21.98 %	**23.89 %**	15.08 %
SVM Classifiers	20.85 %	21.69 %	20.37 %	21.06 %	
Multilayer Perceptron Network	20.76 %	22.15 %	20.29 %	21.81 %	
ZeroR (Best-first)	20.61 %	17.10 %	16.63 %	15.80 %	13.78 %
OneR	22.72 %	16.16 %	17.33 %	18.51 %	14.65 %
Naive Bayes	23.89 %	20.90 %	**23.58 %**	19.44 %	15.05 %
J48 Tree	22.40 %	20.21 %	22.18 %	22.91 %	15.08 %
SVM Classifiers	23.89 %	18.58 %	20.07 %	18.28 %	
Multilayer Perceptron Network	20.84 %	19.59 %	18.28 %	21.81 %	
ZeroR (PCA)	20.61 %	17.10 %	16.63 %	15.80 %	13.78 %
OneR	21.72 %	21.23 %	19.90 %	21.35 %	15.14 %
Naive Bayes	22.65 %	23.80 %	21.31 %	22.11 %	10.36 %
J48 Tree	20.29 %	19.74 %	19.89 %	22.05 %	15.16 %
SVM Classifiers	23.19 %	23.58 %	22.55 %	**25.00 %**	
Multilayer Perceptron Network	21.71 %	22.08 %	20.90 %	21.70 %	

Table 8: Accuracy values for boolean feature experiments. All values are from experiments run on partial data sets (those that only include instances which show at least one abbreviation feature), for comparison. Some of the values that show the greatest improvement over relative baselines are bolded. Results are discussed in subsubsection 7.2.1.

better still using PCA features, however, at 29% above relative baseline. While the SVM did not perform well with raw numeric features or best-first selected features, it outperformed all other classifiers with PCA features at 34% above relative baseline. The high accuracy of these classifiers using numeric features suggested that age classification could be greatly aided by these features. In combination with other features, such as n-gram analysis, very high accuracy could be achieved. Experiments on such combinations are discussed in subsubsection 7.2.4, subsubsection 7.2.6, and subsubsection 7.2.7.

Bins	6	8	10	10	10
Tweets per Instance	100	100	100	75	1
ZeroR (Raw features)	20.61 %	17.10 %	16.63 %	15.80 %	13.78 %
OneR	27.53 %	23.02 %	24.10 %	24.37 %	15.34 %
Naive Bayes	32.31 %	33.80 %	31.55 %	31.31 %	12.46 %
J48 Tree	39.34 %	40.28 %	39.33 %	39.48 %	15.56 %
SVM Classifiers	20.61 %	17.10 %	16.78 %	20.47 %	
Multilayer Perceptron Network	43.66 %	45.68 %	43.08 %	43.56 %	
ZeroR (Best-first)	20.61 %	17.10 %	16.63 %	15.80 %	13.78 %
OneR	28.09 %	23.02 %	24.10 %	24.37 %	15.34 %
Naive Bayes	32.17 %	36.84 %	37.47 %	37.03 %	13.71 %
J48 Tree	39.12 %	**41.85 %**	37.54 %	35.65 %	15.23 %
SVM Classifiers	20.61 %	17.10 %	16.78 %	20.42 %	
Multilayer Perceptron Network	44.87 %	43.47 %	43.56 %	43.35 %	
ZeroR (PCA)	20.61 %	17.10 %	16.63 %	15.80 %	13.78 %
OneR	28.52 %	25.37 %	23.89 %	22.35 %	
Naive Bayes	41.53 %	36.92 %	35.74 %	35.01 %	
J48 Tree	36.47 %	34.50 %	33.57 %	33.68 %	
SVM Classifiers	49.73 %	**50.83 %**	48.17 %	47.10 %	
Multilayer Perceptron Network	45.29 %	**46.44 %**	43.69 %	42.65 %	

Table 9: Accuracy values for numeric feature experiments. All values are from experiments run on partial data sets, for comparison. The values that show the greatest improvement over relative baselines are bolded. Results are discussed in subsubsection 7.2.2.

7.2.3 N-gram Feature Experiments

For comparative purposes, experiments were run using n-gram features to classify age. The data shown in Table 10 reflects experiments on data sets equivalent to those shown in other similar tables. While n-gram features sometimes performed better using the full tweet set (not just those tweets that exhibit abbreviation features), some classifications performed worse with the full data set. Additionally, unlike all boolean and numeric feature based classifications, n-gram classifications frequently performed better using equal-width bins, where each bin covered an equal time span. Results shown are for equal-size bins for comparison purposes.

Classification using n-gram features far outperformed boolean and numeric classifiers, as was

Bins	6	8	10	10
Tweets per Instance	100	100	100	75
ZeroR (Best-first)	20.61 %	17.10 %	16.63 %	15.80 %
OneR	28.25 %	18.81 %	20.53 %	24.31 %
Naive Bayes	83.70 %	81.10 %	79.42 %	**83.21 %**
J48 Tree	80.17 %	79.63 %	77.14 %	81.94 %
SVM Classifiers	81.27 %	76.58 %	70.51 %	76.74 %
Multilayer Perceptron Network	82.99 %	83.77 %	83.30 %	**85.07 %**
ZeroR (PCA)	20.61 %	17.10 %	16.63 %	15.80 %
OneR	31.68 %	41.82 %	38.48 %	36.64 %
Naive Bayes	73.40 %	73.69 %	74.63 %	78.88 %
J48 Tree	70.64 %	73.31 %	71.75 %	74.77 %
SVM Classifiers	63.32 %	60.34 %	56.45 %	62.79 %
Multilayer Perceptron Network	63.15 %	56.49 %	47.85 %	61.17 %

Table 10: Accuracy values for n-gram feature experiments. All values are from experiments run on partial data sets, for comparison. The values that show the greatest improvement over relative baselines are bolded. Results are discussed in subsubsection 7.2.3.

expected. N-grams encompass a greater breadth of information than the selected abbreviation features, so they should be more indicative of user age. Most of the lowest performing classifications in this experiment still outperformed the best in the boolean and numeric experiments.

Interestingly, however, PCA feature extraction did not aid results nearly as much as best-first feature selection. This is likely due to the filtering methods and the way n-gram features are represented in Weka. Weka treats n-gram features as a zero or one boolean value: zero if an n-gram is not present in an instance, one if it is. The best-first selection algorithm chooses subsets of features based on their predictive ability with respect to the data set classifications (bins), without modifying the features. It might choose a collection of 200 n-grams out of 700 total. The PCA algorithm derives different numeric features with high predictive ability by combining the existing features with various fractional constants. It might derive 200 new features like *.1 n-gram-1 + .25 n-gram-2 + .05 n-gram-3*.

The best performing classifiers were the Naive Bayes and MPN classifiers at 67% and 69% above baseline, respectively. Results from combining n-gram features with boolean and numeric abbreviation features are discussed in subsubsection 7.2.5, subsubsection 7.2.6, and subsubsection 7.2.7.

Bins	6	8	10	10	10
Tweets per Instance	100	100	100	75	1
ZeroR (Raw features)	20.61 %	17.10 %	16.63 %	15.80 %	
OneR	28.09 %	23.02 %	24.10 %	24.37 %	
Naive Bayes	34.80 %	34.90 %	32.71 %	33.39 %	
J48 Tree	41.14 %	41.30 %	40.12 %	**40.68 %**	
SVM Classifiers	25.13 %	20.21 %	20.53 %	21.87 %	
Multilayer Perceptron Network	39.73 %	38.73 %	36.84 %	35.81 %	
ZeroR (Best-first)	20.61 %	17.10 %	16.63 %	15.80 %	13.78 %
OneR	28.09 %	23.02 %	24.10 %	24.37 %	15.34 %
Naive Bayes	35.51 %	38.10 %	36.99 %	36.28 %	13.72 %
J48 Tree	43.10 %	**44.04 %**	38.55 %	41.74 %	15.25 %
SVM Classifiers	20.61 %	19.45 %	19.67 %	18.58 %	
Multilayer Perceptron Network	47.14 %	**45.67 %**	46.99 %	**44.38 %**	
ZeroR (PCA)	20.61 %	17.10 %	16.63 %	15.80 %	
OneR	25.01 %	20.06 %	20.77 %	23.25 %	
Naive Bayes	37.55 %	33.98 %	35.03 %	36.91 %	
J48 Tree	32.47 %	31.23 %	27.64 %	28.77 %	
SVM Classifiers	49.50 %	**49.03 %**	47.94 %	46.69 %	
Multilayer Perceptron Network	44.80 %	42.49 %	44.41 %	41.53 %	

Table 11: Accuracy values for boolean and numeric feature experiments. All values are from experiments run on partial data sets, for comparison. The values that show the greatest improvement over relative baselines are bolded. Results are discussed in subsubsection 7.2.4.

7.2.4 Boolean and Numeric Feature Experiments

By simply using a combination of boolean and numeric features, classifiers were found to outperform equivalent experiments using just one of the feature types. Results of these experiments are shown in Table 11. In many cases, the improvements are slight, and in the case of SVM classifiers run on PCA extracted features, there was a slight decrease in accuracy. A J48 tree classifier run on unmodified features (75 tweets per instance, 10 bins) far outperformed its equivalent in either boolean or numeric feature experiments at 25% above baseline. Better performing still was the J48 classifier run on best-first selected features with 100 tweets per instance and 8 bins at 27% above baseline. Within the same feature type, the MPN classifier performed consistently well,

Bins	6	8	10	10
Tweets per Instance	100	100	100	75
ZeroR (Best-first)	20.61 %	17.10 %	16.63 %	15.80 %
OneR	28.25 %	19.13 %	20.53 %	24.31 %
Naive Bayes	76.73 %	75.95 %	74.20 %	76.55 %
J48 Tree	69.70 %	77.23 %	74.31 %	75.69 %
SVM Classifiers	74.70 %	73.16 %	65.21 %	71.94 %
Multilayer Perceptron Network	76.75 %	78.23 %	75.73 %	**78.48 %**
ZeroR (PCA)	20.61 %	17.10 %	16.63 %	15.80 %
OneR	38.25 %	39.73 %	29.43 %	34.95 %
Naive Bayes	63.69 %	71.90 %	73.54 %	77.15 %
J48 Tree	59.63 %	62.83 %	64.59 %	68.88 %
SVM Classifiers	75.79 %	76.27 %	73.38 %	**82.59 %**
Multilayer Perceptron Network	76.75 %	78.23 %	75.73 %	78.48 %

Table 12: Accuracy values for boolean and n-gram feature experiments. All values are from experiments run on partial data sets, for comparison. The values that show the greatest improvement over relative baselines are bolded. Results are discussed in subsubsection 7.2.5.

reaching 29% and 29% above baseline. As in other experiments, the SVM performed much better with PCA extracted features. In the 100 tweet per instance, 8 bin data set, it achieved 32% above relative baseline.

As had been suspected, combining features tended to improve results. Further results of combining features are discussed in subsubsections below.

7.2.5 Boolean and N-gram Feature Experiments

An interesting effect was observed when combining boolean and n-gram features. It was hoped that by combining abbreviation features with n-gram features, a higher overall accuracy could be achieved. However, in combining these features, accuracy for best-first selected features was lowered somewhat. The highest accuracy for best-first selected features was from an MPN classifier at 63% above baseline. In contrast, accuracy for PCA extracted features was greatly improved for SVM and MPN classifiers. The SVM classifier achieved 67% over baseline. These results are shown in Table 12.

These results suggested that combining numeric features with n-gram features or all three types

Bins	6	8	10	10
Tweets per Instance	100	100	100	75
ZeroR (Best-first)	20.61 %	17.10 %	16.63 %	15.80 %
OneR	28.09 %	23.33 %	24.10 %	24.37 %
Naive Bayes	75.40 %	76.03 %	76.20 %	75.81 %
J48 Tree	72.60 %	75.50 %	75.12 %	76.37 %
SVM Classifiers	74.56 %	72.37 %	64.19 %	69.57 %
Multilayer Perceptron Network	82.98 %	79.94 %	78.70 %	**79.17 %**
ZeroR (PCA)	20.61 %	17.10 %	16.63 %	15.80 %
OneR	37.06 %	40.82 %	28.42 %	34.15 %
Naive Bayes	60.51 %	71.20 %	75.11 %	*77.26 %*
J48 Tree	61.67 %	64.00 %	68.40 %	71.69 %
SVM Classifiers	76.33 %	76.43 %	73.54 %	**82.82 %**
Multilayer Perceptron Network	63.77 %	57.93 %	60.98 %	66.04 %

Table 13: Accuracy values for numeric and n-gram feature experiments. All values are from experiments run on partial data sets, for comparison. The values that show the greatest improvement over relative baselines are bolded. Results are discussed in subsubsection 7.2.6.

of features would lead to even better results. Experiments on those combinations are discussed below in subsubsection 7.2.6 and subsubsection 7.2.7.

7.2.6 Numeric and N-gram Feature Experiments

Experiments on numeric and n-gram features yielded results very similar to those on boolean and n-gram features, as shown in Table 13. The accuracy of best-first selected features was lowered in comparison to just n-gram features, while the accuracy of PCA extracted features was improved. Again, the SVM classifier performed best with PCA features, and the MPN classifier performed best in the best-first feature experiments.

While very similar, the results of the well performing classifiers in these experiments were slightly better than those from the boolean and n-gram experiments. However, those classifiers that were not shown to do as well (SVM in best-first experiments, MPN in PCA experiments) performed much less well with numeric and n-gram features. A similar effect was also observed with the experiments described in the following section.

Bins	6	8	10	10
Tweets per Instance	100	100	100	75
ZeroR (Best-first)	20.61 %	17.10 %	16.63 %	15.80 %
OneR	28.09 %	23.33 %	24.10 %	24.37 %
Naive Bayes	75.40 %	76.19 %	**76.51 %**	75.81 %
J48 Tree	72.60 %	75.26 %	75.12 %	76.37 %
SVM Classifiers	74.56 %	72.22 %	63.80 %	69.57 %
Multilayer Perceptron Network	82.98 %	80.40 %	**80.26 %**	79.17 %
ZeroR (PCA)	20.61 %	17.10 %	16.63 %	15.80 %
OneR	38.25 %	42.13 %	29.67 %	37.04 %
Naive Bayes	61.74 %	70.96 %	71.43 %	76.11 %
J48 Tree	58.41 %	64.54 %	69.10 %	72.15 %
SVM Classifiers	75.79 %	75.34 %	73.07 %	**82.53 %**
Multilayer Perceptron Network	63.31 %	57.51 %	63.08 %	67.02 %

Table 14: Accuracy values for boolean, numeric, and n-gram feature experiments. All values are from experiments run on partial data sets, for comparison. The values that show the greatest improvement over relative baselines are bolded. Results are discussed in subsubsection 7.2.7.

7.2.7 Boolean, Numeric, and N-gram Feature Experiments

Experiments combining all three feature types showed slight improvements over those experiments that combined n-gram features and boolean or numeric features. Results are displayed in Table 14. The Naive Bayes classifier performed quite well with best-first selected features, ten bins, and 100 tweets per instance at 60% above relative baseline. It was surpassed in the same category by the MPN classifier, which achieved 64% above baseline. These classifiers did not perform as well with PCA extracted features. The SVM classifier performed better with PCA extracted features, however, and achieved the best improvement over baseline at 67% with ten bins and 75 tweets per instance.

The results of this experiment are promising, however they indicate a ceiling on accuracy rates using the given features. Furthermore, combining all three feature types is not necessary, as nearly identical results can be achieved using n-gram features and one type of boolean or numeric features.

	Training				Withheld Testing			
Classifier	Accuracy	Prec.	Recall	F-Meas.	Accuracy	Prec.	Recall	F-Meas.
ZeroR	17.76 %	0.03	0.18	0.05	0.00 %	0.00	0.00	0.00
0R 1951 WH	17.94 %	0.03	0.18	0.06	0.00 %	0.00	0.00	0.00
J48	79.08 %	0.80	0.79	0.78	12.50 %	0.08	0.13	0.10
J48 1951 WH	79.85 %	0.81	0.80	0.79	10.00 %	0.07	0.10	0.08
SVM	59.61 %	0.64	0.60	0.58	31.25 %	0.31	0.31	0.31
SVM 1951 WH	59.95 %	0.65	0.60	0.58	25.00 %	0.25	0.25	0.25
MPN	61.56 %	0.64	0.62	0.61	18.75 %	0.28	0.19	0.20
MPN 1951 WH	60.93 %	0.65	0.61	0.61	15.00 %	0.27	0.15	0.16

Table 15: Results of withheld user testing, 100 tweets per instance. "1951 WH" denotes classifiers run with the 1951 user excluded from the training set, and included in the withheld testing set. As could be expected, withheld testing performed better when the 1951 outlyer was included in training.

7.2.8 Withheld Users Experiments

Based on the results of experiments on boolean and numeric features, described in subsection 7.2, three pairings of classifiers and data sets were selected for experiments involving novel data testing. The selected data sets all featured 100 tweets grouped per instance and numeric-based features and were filtered to only include tweets exhibiting one or more features. A J48 tree classifier using a 0.2 confidence factor was selected to be run on data sets with 8 class bins and features filtered using a best-first feature selection algorithm. An SVM classifier using parameters as described in subsubsection 7.1.1 was run on data sets with 10 bins, with features derived from PCA. The third classifier was an MPN, again as described in subsubsection 7.1.1, which was run on data sets with 8 bins and PCA-derived features.

For each classifier-data set pairing, 4 data sets were developed for training and testing. The testing sets had all data from 6 and 7 participants withheld (about 10% of the participants), while the training sets had the remaining data. This was similar to the 10-fold cross validation experiments described above, except that only one fold was used for testing. Additionally, instead of being randomly selected, the fold containing data from withheld participants was selected manually based on the users' reported age in order to get an even age distribution. The participants selected to be withheld reported birth years were 1995, 1990, 1985, 1980, 1976, 1970, and 1951. In order to investigate how the outlier born in 1951 affected results, half the data sets had the 1951

	Training				Withheld Testing			
Classifier	Accuracy	Prec.	Recall	F-Meas.	Accuracy	Prec.	Recall	F-Meas.
ZeroR	17.36 %	0.03	0.17	0.05	0.00 %	0.00	0.00	0.00
0R 1951 WH	17.52 %	0.03	0.18	0.05	0.00 %	0.00	0.00	0.00
J48	82.64 %	0.80	0.79	0.78	13.04 %	0.08	0.13	0.10
J48 1951 WH	83.21 %	0.84	0.83	0.83	10.71 %	0.13	0.11	0.10
SVM	62.57 %	0.66	0.63	0.61	13.04 %	0.10	0.13	0.11
SVM 1951 WH	62.96 %	0.67	0.63	0.61	14.29 %	0.10	0.14	0.11
MPN	64.20 %	0.68	0.64	0.64	17.39 %	0.10	0.17	0.13
MPN 1951 WH	65.51 %	0.68	0.66	0.66	14.29 %	0.13	0.14	0.12

Table 16: Results of withheld user testing, 75 tweets per instance. "1951 WH" denotes classifiers run with the 1951 user excluded from the training set, and included in the withheld testing set. Compared to Table 15, classifiers perform better at classifying training data when trained with with an increased number of data instances, but fall short in withheld testing.

participant's data withheld to the testing set, and half had it included in the training set. The four data sets each had 427 instances, 16 of which were withheld when 6 users are withheld for testing after training. 20 were withheld when the 1951 user was withheld.

The results of the four classifier runs are shown in Table 15. The J48 classifier performed very well in training, but did not handle novel data well. In contrast, the SVM classifier performed a bit less well, but performed significantly better with novel data. The MPN classifier performed only slightly better than the SVM classifier in training, but much worse in testing. The effect of training with the 1951 user's data versus without was noticeable only in the testing phase. It had little effect on the results of training, but consistently decreased testing accuracy when it was included in the testing set.

Based on the results above, equivalent data sets were created using 75 tweets per instance instead of 100. This increased the total number of instances in each data set to 576 (23 and 28 withheld, compared to 16 and 20 above). The results of these experiments are shown in Table 16. By increasing the number of training and testing instances, it was hoped that novel data testing would show improved results. All the training results were better, however, only the J48 tree classifier showed improvement in withheld testing. This would indicate that larger amounts of training data will improve classification results, but hinder the classifier when working with novel data.

Rule	Confidence
* Turkey=false → Turkish=false	100%
* Poland=false → Polish=false	100%
* Germany=false → German=false	100%
* Japanese=false → Japan=false	100%
Mandarin=false → Japan=false	100%
Japan=false → Mandarin=false	98%
* *You to U*=False → *Word End*=False	99%
* *You to U*=False → *Single Character*=False	99%
Drop Last Character=False → *Word End*=False	99%
Word End=False → *Drop Last Character*=False	98%
Drop Last Character=False → *Single Character*=False	99%
Single Character=False → *Drop Last Character*=False	98%
Drop Last Character=False → *You to U*=False	99%
You to U=False → *Drop Last Character*=False	98%
* *Drop Last Character*=False → *Repeat Letter*=False	98%

Table 17: Some association rules found in analysis. Rules that were found with the same confidence in both directions are denoted with an asterisk (*). In most cases, rules with 100% confidence dealt with language and region outlyers. Languages other than English were uncommon, as were non-English speaking countries. The high confidence rules dealing with abbreviation patterns were based on low-occurrence patterns. The *Contraction* abbreviation pattern, for example, does not appear until much lower confidence levels. This also leads to many similar associations, such as not having lived in a country implies not using an uncommon abbreviation type.

7.3 Association Mining

Following the above described experiments, a series of experiments were conducted using an apriori association mining algorithm in Weka [31, 3]. The primary goal of these experiments was to investigate if any useful rules could be generated that associate user age classifications, abbreviation features, and collected demographic data.

Generated rules were evaluated based on their confidence values. A rule's confidence is defined as a percentage. Out of a set of instances from the overall data set where the conditional portion (the left side) of the rule is observed, the confidence is the percentage of those instances where the consequent portion (the right side) of the rule is true [31].

The first step for this experiment series was data preparation. In all previous experiments, demographic data (except for age classifications) were withheld from classifiers. Since part of

Rule	Conf.
Occupation=Student → Age Class<=19.0	36%
Occupation=Student *Drop Vowel*=False → Age Class<=19.0	37%
Occupation=Student *Drop Vowel*=False *Contraction*=False → Age Class<=19.0	40%
Occupation=Student Gender=M → Age Class<=19.0	46%
Gender=F Education=Bachelor's Degree → Age Class<=25.0	28%
Gender=F Education=Bachelor's Degree *Drop Vowel*=False → Age Class<=25.0	28%
Education=Some College → Age Class<=21	25%
Education=Some College *Drop Last Character*=False → Age Class<=21	25%
Gender=M → Age Class<=21	25%
Gender=M *You to U*=False *Drop Vowel*=False → Age Class<=21	24%
Gender=F → Age Class<=33.0	18%
Gender=F *Word End*=False *Th to D*=False → Age Class<=33.0	18%
Gender=Male United States=true → Age Class<=21.36	69%

Table 18: Class association rules (CARs) found in analysis. Above, female users were age 30–33 18% of the time. Adding abbreviation features to this rule did not change the confidence level, as for many other associations. However, when examining the occupation feature, adding additional feature information to the rule raised its confidence up to 10%. In some cases, the additional information even lowered the rule's confidence, such as with males in the age class <= 21. Features with higher variance (which would be evident with numeric features), such as the *Contraction* abbreviation feature did not appear above 42% confidence when making CARs. One of the highest confidence CARs is shown with a confidence of 69% when excluding language data and including region data.

the focus of these experiments was to investigate the relationship between collected demographic data and abbreviation features, the demographic data had to be preprocessed and associated with proper instances with abbreviation features and age classifications. A small part of this preparation involved normalizing values for features such as education, as described in subsection 5.2. The rest of this preparation involved splitting multi-valued features, such as region and language, into several boolean features, eg. *Australia*: true if a user reported having lived in Australia; or *Turkish*: true if a user reported writing in Turkish at any time.

Using the same data set paradigm as described in section 7 and subsection 7.2, data sets with boolean features and 10 bins were examined. Rules were generated from sets with groupings of 1, 75, and 100 tweets per instance. These data set selections gave a group of results to compare to

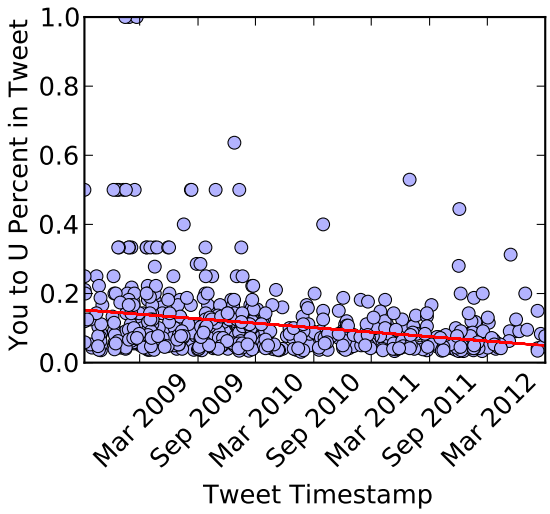
above experimental results. For the most part, the best rules were found from the 100 tweets-per-instance data set. The algorithm used was only able to operate on nominal features, so numeric abbreviation features and n-gram features could not be used. Grouped instances likely showed better results due to the boolean feature type containing less information than an equivalent numeric type, as well has having a smaller total number of instances. For comparison, results highlighted in Table 8 may be useful.

Initial runs of the association algorithm were not restricted to creating class association rules (CARs) [31]. Such general association rules map combinations of features to other combinations of features, as opposed to CARs, which map combinations of features to classes. These general association runs generated many associations that might be considered obvious, and were found with a confidence of 100% or very close to it. Several such rules are shown in Table 17. For example, those instances where a user did not report speaking Turkish, the user also did not report living in Turkey. Such rules were based on the collection of instances belonging to one or a few users, as such internal consistency was high. They may not be considered useful rules, however, as the language and region values reported were not associated directly with tweets, only with the user. Most tweets were not examined for spoken language, as it was assumed to be English, except in one case where a user tweeted solely in Lojban, a logically engineered artificial language.

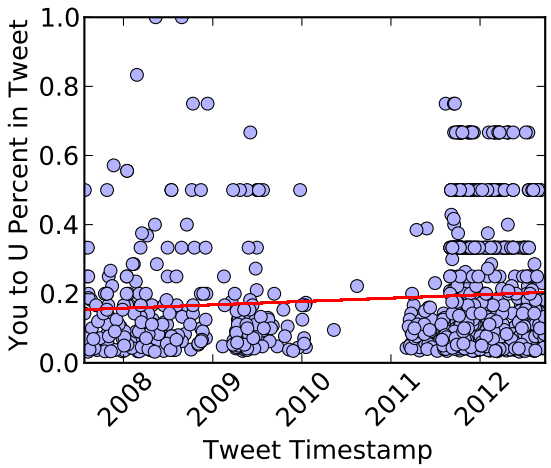
More interesting rules were generated when the algorithm was set to only create CARs and when the available features were reduced to exclude language and region features. In many cases, a mixture of features did not alter rule confidence. Sometimes, such as shown in Table 18, additional features in the conditional part of the rule lowered confidence values. In a few cases, however, combining additional features with a basic rule raised confidence a great deal. Based on features showing educational level (Occupation=Student, Education), females reported higher education levels. Within the data set, a larger number of females reported in the age ranges of 23–28, while males reported in larger numbers as under 21. This could account for the higher educational levels reported by females. Similarly, those who reported their occupation as *Student* were most often classified as under age 19.

CARs ignoring demographic data showed confidence peaking at 14%. Most of these rules did not outperform statistical baselines. For example, a number of rules combined various abbreviation features and the age classification <=21 with confidence of 14%. However, that age classification also appears in the data set at a rate of 14%. In these rules, additional features in the conditional portion of the rule often lowered its confidence below baseline slightly. No notable rules of this type were generated from the 1 tweet per instance data set.

However, when examining the 100 tweets per instance data set for rules ignoring demographic



(a) The most negative slope for *You to U* feature use percentage over time at −104.35.



(b) The most positive slope for *You to U* feature use percentage over time at 50.14.

Figure 11: Plots of *You to U* feature use percentages over time. When plotting feature usage percents versus their respective tweet timestamps, some change can be observed with respect to time, especially in the *You to U* feature. The most negative and most positively sloped best fit lines are shown here. Slopes are based on UTC timestamp versus percentage occurrence of a feature in a tweet. The slopes are normalized with time span shown being normalized to span 0 to 1.

data, the results changed a bit. This data set showed confidence peaking at 22%. A collection of rules, such as "Users using the *prefix* abbreviation and not using the *Th to D* or *Single Character* abbreviations are classified as age 23–25" are found at this confidence level. This classification is present in 16% of instances, showing there is an observable association between age and abbreviation patterns, as found in previous experiments. For comparison, the boolean feature experiments on the same data set achieved almost 7% accuracy improvement over baseline, while this set of rules achieved 6%. Several other collections of abbreviation feature-based rules appear with lower confidence levels. These rules still outperform random chance selection, but not as well as the 22% confidence group or many of the higher performing demographic data-based rules.

Astrology was included as a feature in all association experiments. Despite this, it was never used in any rule. This suggests its use as a control variable is warranted, since it has no observed relationship to other features.

7.4 Longitudinal Analysis

Few users in the collected data set provided tweets that covered a large time span. The top 10 longest tweeting users were selected from the data set for longitudinal analysis, to see if any change in their usage of abbreviation types could be observed. The tweets analyzed covered from 4 to just over 5 years per user. Tweets were plotted with UTC Unix epoch timestamp on the x axis and abbreviation feature percentage on the y axis. The collection of points was analyzed with NumPy to obtain a best fit line, which minimizes squared error [47]. The best fit line was plotted on top of the data points for visualization.

Due to the large numbers that are used to represent UTC timestamps, as well as the inconsistent time frame covered by each user's tweets, the timestamps were normalized during analysis. Each user's tweet timestamps were normalized to span 0 to 1 for analysis purposes. In most cases, this resulted in a slope calculated for each abbreviation feature type in the range -1 to 1. In some cases, such as those displayed in Figure 11, the slope far exceeded those bounds.

The *You to U* abbreviation type feature had the most noticeable change over time compared to other abbreviation types, as well as the most significant amount of change, as shown in Figure 11a and Figure 11b with slopes of -104 and 50, respectively. 9 out of 10 users showed some change in their use of the *You to U* abbreviation type. For the most part, the calculated slopes were negative, suggesting that users tended to use less of the abbreviation as they got more familiar with Twitter. The most negative slope of the ten selected users is shown in Figure 11a. 3 of the 10 users showed a positive (even if slight) slope for the *You to U* feature. The most positive is shown in Figure 11b. Other features showed slight change over time within the users, but the *You to U* feature showed the most change.

In some cases, analysis indicated that a user began to use significantly less of one abbreviation type and somewhat more of another. This would suggest that as users get more familiar with the Twitter service, they adapt their writing patterns to fit the medium, tending to write more standard English as time passes, but sometimes adopting a more comfortable abbreviation type, such as contractions, in favor of another.

These changes might also parallel language change with respect to abbreviation usage in other mediums, such as SMS. Many abbreviations were used in SMS texts, due to the restricted character length, similar to Twitter. As smart-phones with full keyboards have become more popular and text messaging has become cheaper, it is easier for SMS users to type full words and utilize multiple sequential SMS messages for longer messages, which lessens the need for abbreviations. Should such usage changes be a reality for SMS, it is likely that they will be reflected in other restricted-length texts, such as on Twitter. Such parallels are additionally likely due to the prevalence of

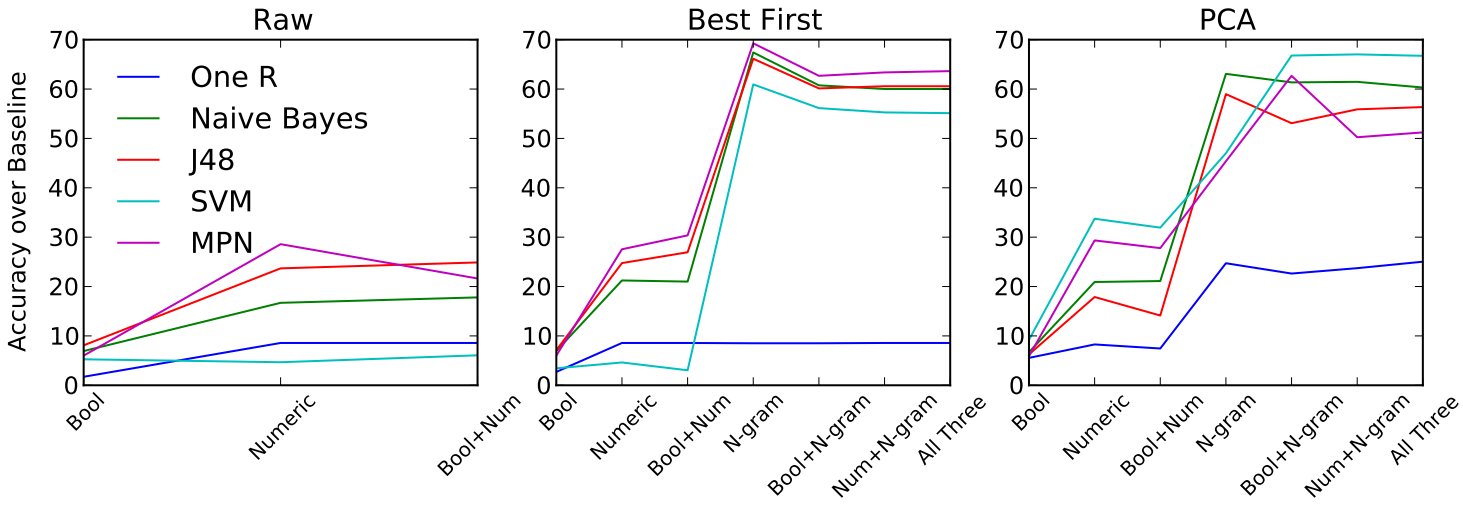


Figure 12: Best achieved accuracy for each feature type and classifier. Best results were chosen independent of data set, though most were from the 10 bin data sets with either 100 or 75 tweets per group. Displayed values are the percentage accuracy above each relative ZeroR baseline. The *Raw* chart displays results from experiments on features without best first feature selection or PCA feature extraction run on them, while the *Best First* and *PCA* charts show results of experiments run on features after applying a best-first feature selection algorithm or principal component analysis feature extraction, respectively.

creating Twitter messages from smart-phones using the same input systems as SMS.

8 Conclusions

The results of the classifier algorithm experiments (subsection 7.2) were a promising verification of the hypothesis that word and phrase abbreviation types used on Twitter can aid in classifying a Twitter user's age. Classifications using word and phrase abbreviation patterns alone did not perform as well as those using n-gram features, as could be expected. The information sparsity of abbreviation features compared to the wealth of information contained in unigrams contributes to the lower accuracy. Abbreviation classification did significantly outperform relative baselines, however, and may be a useful in the future as an additional feature for those seeking to perform age and other demographic classification on noisy texts, such as those found on Twitter. Classifications using combinations of n-gram and abbreviation features performed quite well, and using feature selection or PCA improved classification results further still. Some of the best performing classifier and feature combinations are shown in Figure 12.

In addition, somewhat equivalent relationships between abbreviation features and age classi-

fications were found using association mining, as described in subsection 7.3. When including demographic data, many other relationships were observed that reflected the state of the collected demographic data. However, much of the time, including abbreviation features in the demographic-based rules had little or no effect on the confidence value of the rule. This would suggest that certain relationships between demographic data are much stronger than those between abbreviation features and demographic data, which makes intuitive sense.

Lastly, as described in subsection 7.4, there is some observable evidence for longitudinal change in abbreviation pattern use within the collected data set. A little change was observed in use of most abbreviation types, while the most change was observed in the *You to U* abbreviation type. Users seem to write with fewer abbreviations as they become more comfortable with the Twitter medium, or adopt their writing in other ways to fit its restrictions. These changes may parallel language changes in other mediums as well, such as in SMS message writing.

9 Future Work

There are a few directions one could take to further develop on the work presented here. First, further experimentation using word and phrase abbreviation features can be done. The abbreviation features presented here may be indicative of different demographic data than has been shown (age, in this work, and user time zone and Twitter client by Gouws *et al.* [17]). Additionally, the work may be generalizable to more types of noisy texts, such as SMS messages, and perhaps non-length restricted texts, such as in online forums.

Second, but perhaps most importantly, further development of the collected data set should be pursued. The data set is somewhat small, but has already shown to be useful for age classification problems. Getting a large number of users to participate in such a data collection may be a difficult task, but doing so could pave the way for a larger variety of future research on Twitter texts.

Third, as it is a frequent topic of interest in data analysis, work could be done on clustering such data sets. Relationships between the data and classifications do seem to exist, so it would make sense if a clustering algorithm may be able to harness those relationships for interesting, effective clustering.

Last, further longitudinal study should be pursued. Language use on Twitter is undoubtedly evolving, reflecting the evolution of spoken and other textual language. The data collected and analyzed here did not cover a large time span, but a few noticeable within-user changes in abbreviation feature use could be detected. With a larger data set that spans a longer amount of time, perhaps more language use change could be observed. It would also be of interest to compare

changes observed in writing on Twitter to changes in other restricted-length texts, such as SMS or other chat type corpora.

References

[1] Report: Social Network Demographics in 2012. [Online] Available: `http://royal.pingdom.com/2012/08/21/report-social-network-demographics-in-2012/`, August 2012.

[2] What is the Frequency of the Letters of the Alphabet in English? [Online] Available: `http://oxforddictionaries.com/words/what-is-the-frequency-of-the-letters-of-the-alphabet-in-english`, 2012.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[4] Chidanand Apte, Bing Liu, Edwin P. D. Pednault, and Padhraic Smyth. Business Applications of Data Mining. *Commun. ACM*, 45(8):49–53, August 2002.

[5] Steven Bedrick, Russell Beckley, Brian Roark, and Richard Sproat. Robust Kaomoji Detection in Twitter. In *Proceedings of the Second Workshop on Language in Social Media*, pages 56–64, Montréal, Canada, June 2012. Association for Computational Linguistics.

[6] John D. Burger, John Henderson, George Kim, and Guido Zarrella. Discriminating Gender on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1301–1309, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[7] William B. Cavnar and John M. Trenkle. N-Gram-Based Text Categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011.

[9] John Chen and Owen Rambow. Use of Deep Linguistic Features for the Recognition and Labeling of Semantic Arguments. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 41–48, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[10] Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. Unsupervised Cleansing of Noisy Text. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 189–196, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[11] Paul Cook and Suzanne Stevenson. An Unsupervised Model for Text Message Normalization. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC '09, pages 71–78, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[12] Evandro Cunha, Gabriel Magno, Giovanni Comarela, Virgilio Almeida, Marcos André Gonçalves, and Fabrício Benevenuto. Analyzing the Dynamic Evolution of Hashtags on Twitter: A Language-based Approach. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 58–65, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[13] D. Recordon E. Hammer-Lahav and D. Hardt. The OAuth 1.0 Protocol. [Online]. Available: `http://tools.ietf.org/html/rfc5849`, April 2010.

[14] Nikesh Garera and David Yarowsky. Modeling Latent Biographic Attributes in Conversational Genres. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 710–718, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[15] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In *ACL (Short Papers)*, pages 42–47. The Association for Computer Linguistics, 2011.

[16] Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL '06)*, pages 401–408, Trento, Italy, April 2006. European Chapter of the Association for Computational Linguistics.

[17] Stephan Gouws, Donald Metzler, Congxing Cai, and Eduard Hovy. Contextual Bearing on Linguistic Variation in Social Media. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 20–29, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[18] Bo Han and Timothy Baldwin. Lexical Normalisation of Short Text Messages: Makn Sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 368–378, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[19] Peter Hartmann, Martin Reuter, and Helmuth Nyborg. The Relationship Between Date of Birth and Individual Differences in Personality and General Intelligence: A Large-scale Study. *Personality and Individual Differences*, 40(7):1349–1362, May 2006.

[20] Robert C. Holte. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning*, 11:63–91, 1993.

[21] John Hutchins. Retrospect and Prospect in Computer-Based Translation. In *Machine Translation Summit VII, 13th-17th September 1999, Kent Ridge Labs, Singapore. Proceedings of MT Summit VII "MT in The Great Translation Era"*, pages 30–34, Tokyo, September 1999. Asia-Pacific Association for Machine Translation.

[22] Wonhong Lee Hyung Jin Kim, Minjong Chung. Literary Style Classification with Deep Linguistic Analysis Features. Technical report, Department of Computer Science, Stanford University, 2011.

[23] Google Inc. How Does Google Target Ads to My Website? - AdSense Help. [Online]. Available: `http://support.google.com/adsense/bin/answer.py?hl=en&answer=9713`, 2012.

[24] Twitter Inc. Twitter Blog: One Million Registered Twitter Apps. [Online]. Available: `http://blog.twitter.com/2011/07/one-million-registered-twitter-apps.html`, July 2011.

[25] Twitter Inc. Developer Rules of the Road. [Online]. Available: `https://dev.twitter.com/terms/api-terms`, May 2012.

[26] Twitter Inc. Twitter Blog: Twitter Turns Six. [Online]. Available: `http://blog.twitter.com/2012/03/twitter-turns-six.html`, March 2012.

[27] Twitter Inc. Twitter Help Center — Get to Know Twitter: New User FAQ. [Online]. Available: `https://support.twitter.com/groups/31-twitter-basics/topics/104-welcome-to-twitter-support/articles/13920-get-to-know-twitter-new-user-faq`, 2012.

[28] George H. John and Pat Langley. Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.

[29] Max Kaufmann and Jugal Kalita. Syntactic Normalization of Twitter Messages. In *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*, pages 149–158, Karagpur, India, 2010. Macmillan Publishers, India.

[30] Ron Kohavi. The Power of Decision Tables. In *Proceedings of the European Conference on Machine Learning*, pages 174–189. Springer Verlag, 1995.

[31] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating Classification And Association Rule Mining. In *Proceedings of the 4th international conference on Knowledge Discovery and Data mining (KDD'98)*, pages 80–86. AAAI Press, August 1998.

[32] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text Classification Using String Kernels. *Journal of Machine Learning Research*, 2:419–444, March 2002.

[33] Edward Loper and Steven Bird. NLTK: the Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, volume 1 of *ETMTNLP '02*, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[34] J. B. Lovins. Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[35] Beáta Megyesi. Shallow Parsing with POS Taggers and Linguistic Features. *Journal of Machine Learning Research*, 2:639–668, March 2002.

[36] Roy Murphy. An Analysis of the Distribution of Birthdays in a Calendar Year. [Online] Available: `http://www.panix.com/~murphy/bday.html`.

[37] Dong Nguyen, Noah A. Smith, and Carolyn P. Rosé. Author Age Prediction From Text Using Linear Regression. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, LaTeCH '11, pages 115–123, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[38] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[39] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. Classifying Latent User Attributes In Twitter. In *Proceedings of the 2nd international Workshop on Search and Mining User-Generated Contents*, SMUC '10, pages 37–44, New York, NY, USA, 2010. ACM.

[40] Ronald Rosenfeld. Two Decades of Statistical Language Modeling: Where Do We Go From Here? *Proceedings of the IEEE*, 88(8):1270–1278, August 2000.

[41] Sara Rosenthal and Kathleen McKeown. Age Prediction in Blogs: A Study of Style, Content, and Online Behavior in Pre- and Post-Social Media Generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 763–772, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[42] Ruchita Sarawgi, Kailash Gajulapalli, and Yejin Choi. Gender Attribution: Tracing Stylometric Evidence Beyond Topic and Genre. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 78–86, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[43] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New Support Vector Algorithms. *Neural Computation*, 12(5):1207–1245, May 2000.

[44] Aaron Smith and Joanna Brenner. Twitter Use 2012. Technical report, Pew Research Centers Internet & American Life Project, 1615 L St., NW – Suite 700 Washington, D.C. 20036, May 2012.

[45] Richard Sproat, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. Normalization of Non-standard Words. *Computer Speech & Language*, 15(3):287–333, 2001.

[46] Andreas Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904, 2002.

[47] Matthew Turk. Analysis and Visualization of Multi-Scale Astrophysical Simulations Using Python and NumPy. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 46–50, Pasadena, CA USA, 2008.

[48] Goldee Udani. An Exhaustive Study of Twitter Users Across the World. [Online] Available `http://www.beevolve.com/twitter-statistics/`, October 2012.

[49] Suzanne Evans Wagner. Age Grading in Sociolinguistic Theory. *Language and Linguistics Compass*, 6:371–382, June 2012.

[50] Yusuke Yamamoto. Twitter4J - A Java Library for the Twitter API. [Online]. Available: `http://twitter4j.org`.

Appendices

Appendix A Demographic Tables

Month	Count	Percentage
January	2	3 %
February	5	7 %
March	5	7 %
April	6	8 %
May	7	10 %
June	7	10 %
July	5	7 %
August	4	6 %
September	11	15 %
October	6	8 %
November	2	3 %
December	5	7 %
Not Specified	7	10 %
N = 72		

(a) Birth Months

Language	Count	Percentage
Arabic	1	1 %
English	66	72 %
Esperanto	1	1 %
French	5	5 %
German	2	2 %
Italian	1	1 %
Japanese	2	2 %
Latin	2	2 %
Lojban	2	2 %
Mandarin	1	1 %
Polish	1	1 %
Russian	1	1 %
Spanish	1	1 %
Turkish	1	1 %
Not Specified	5	5 %
N = 92		

(b) Languages Used Most on Twitter

Education	Count	Percentage
High School	5	7 %
Currently In College	6	8 %
Some College	10	14 %
Bachelor's Degree	28	39 %
Associate's Degree	1	1 %
Master's Degree	12	17 %
Doctoral Degree	2	3 %
Not Specified	8	11 %
N = 72		

(c) Highest Education

Gender	Count	Percentage
Female	37	51 %
Male	29	40 %
Not Specified	6	8 %
N = 72		

(d) Genders

Astrology	Count	Percentage
Aquarius	2	3 %
Pisces	5	7 %
Aries	4	6 %
Taurus	8	11 %
Gemini	5	7 %
Cancer	6	8 %
Leo	3	4 %
Virgo	9	13 %
Libra	2	3 %
Scorpio	4	6 %
Sagittarius	2	3 %
Capricorn	1	1 %
Not Specified	21	29 %
N = 72		

(e) Astrological Signs

Region	Count	Percentage
Australia	4	4 %
Canada	2	2 %
France	1	1 %
Germany	1	1 %
Japan	1	1 %
Poland	1	1 %
Turkey	1	1 %
United Kingdom	9	8 %
United States	8	7 %
Midwest US	16	14 %
Northeast US	23	20 %
South US	18	16 %
West US	22	19 %
Not Specified	6	5 %
N = 113		

(f) Geographical Regions of Residence

Vita

Nathaniel Moseley was born in Missoula, Montana on April 6, 1989. He received the Bachelor of Science degree in Computer Science from Rochester Institute of Technology, Rochester, New York, United States of America in 2012. He is currently pursuing his Master of Science degree in Computer Science from Rochester Institute of Technology. His research interests are varied and include Computational Linguistics, Computer Vision, and Psychology.

This thesis was typeset with LaTeX by the author.