Spring 2013

The Insignificance of Feature Frequency in Classifying Gender of Twitter Tweets

Amanda Marie Kroft

THE INSIGNIFICANCE OF FEATURE FREQUENCY IN CLASSIFYING GENDER OF TWITTER TWEETS

A Thesis

Submitted to the McAnulty College of Liberal Arts

Duquesne University

In partial fulfillment of the requirement of

the degree of Master of Science

By

Amanda Kroft

May 2013

THE INSIGNIFICANCE OF FEATURE FREQUENCY IN

CLASSIFYING GENDER OF TWITTER TWEETS

By

Amanda Kroft

Approved March 26, 2013

Patrick Juola Ph.D. Associate Professor Thesis Advisor, Chair of Committee	John Kern, Ph.D. Associate Professor Committe Member
Donald Simon, Ph.D. Director of Graduate Studies Professor of Computer Science	James Swindal, Ph.D. Dean, McAnulty College

ABSTRACT

THE INSIGNIFICANCE OF FEATURE FREQUENCY IN CLASSIFYING GENDER OF TWITTER TWEETS

By

Amanda Kroft

May 2013

Thesis supervised by Patrick Juola Ph.D.

In 2011, Internet users spent almost 23% of their time on social media sites such as Twitter and Facebook [5]. Twitter alone was estimated to have over 200 million active users [2]. With social media being such a popular online pastime, a tremendous amount of information becomes available from the posts that users put on social media sites. This information has the potential to reveal details about the social media users, such as the relationship between characteristics of the users and what they post. This relationship is a hot research topic and one of the most frequently studied characteristic is the gender of a user. Feature frequency is often included in such a task, but this thesis shows that for Twitter tweets it either does not contribute significantly to gender classification or hinders classification.

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1

Introduction

1.1 Authorship Attribution and Profiling

Authorship attribution is defined as the process of determining the author of an anonymous document. A similar process, called authorship profiling, is determining characteristics of the author of an anonymous document, such as age. Some researchers consider authorship profiling to be a specific type of authorship attribution [9]. For the purposes of this research authorship profiling will be a slightly different task than attribution.

Both authorship attribution and authorship profiling use the same methods of analysis. To start, features, such as words or characters, are extracted from all documents so each document becomes a set of features. The classifier trains a model, usually using machine learning, on the set of features from documents with known authors or characteristics and then classifies the anonymous documents using the constructed model. Usually the training of a model is called the training phase and the classifying of the unknown documents is called the testing phases. Optionally, the features extracted from the documents can be standardized, such as normalizing whitespace, using canonicizers or have certain features removed, such as non-ASCII characters, using cullers.

There are many sources of linguistic data that can be used for authorship attribution and profiling. These sources range from essays [11] to emails [6] to social media posts [12].

Theoretically, any source of linguistic data can be used for authorship attribution and profiling. With social media becoming a popular pastime in the past several years [5], the use of its data for authorship attribution and profiling has been on the rise.

1.2 JGAAP and WEKA

The Java Graphical Authorship Attribution Program (JGAAP) is a program created by the Evaluating Variation in Language Laboratory (EVL Lab) at Duquesne University which provides commonly used tools and algorithms for authorship attribution and profiling [8]. The methods described in section 1.1 are built into JGAAP and a graphical user interface is provided in order to provide a simple way for researchers to attribute authorship or to profile an author. For larger quantities of experiments, a command line experiment engine is also provided by JGAAP. The classification algorithms of JGAAP are termed "analysis drivers".

Two commonly used JGAAP classifiers are the Nearest Neighbor analysis driver and the Centroid analysis driver. Both of these classifiers use a distance function in order to determine a classification of an unknown document. Nearest Neighbor driver merely uses the distance function to determine which known document is closest to the current unknown document. Centroid driver classifies in a similar manner, but instead first determines the centroid, or the mean, of each group of documents with the same author. It then uses the distance function to determine which centroid is closest to the unknown document.

WEKA is a data mining tool written in Java which provides a collection of machine learning algorithms. WEKA is an open source project and so its algorithms can be incorporated in to other Java projects [7]. Some of these algorithms can be used in authorship attribution and so JGAAP has incorporated several commonly used WEKA algorithms into its classifier set.

1.3 Social Media and Twitter

Social media sites allow users to post information which can be viewed, shared, and discussed by other users. Social media sites, out of necessity, will restrict these posts by limiting the character limit [14] or number of posts per day [13] but rarely ever restrict the content of posts. The content is purely of the whim of the user and therefore has been researched as to giving insight into the characteristics of the user. The posts from users are often available for download and can be used for research.

Twitter is a social media website which allows its users to post documents on a subject of their choice, similar to blogging websites. These documents are limited to at most 140 characters and because of their small size are referred to as microtext. Twitter refers to the microtext on its site as tweets. Users are also able to post information about themselves, such as name, and change visual effects of their main page, such as the background color.

Twitter offers an Application Programming Interface (API) in order for various public information on its users and their tweets to be easily accessed. For Twitter users various pieces of information about them are publicly accessible, such as self-reported name, number of followers, and time zone. A subset of their recent tweets, called their timeline, is also publicly accessible. In addition to being able to search for information on specific users, Twitter also offers the ability to do standard searches for tweets using their API. The searches can be refined in any way number of ways including by location, language, and date posted. All the data collected through Twitter's API can then be used to conduct research. The data cannot be redistributed to other researchers, but the user IDs and tweet IDs can be released so other researchers can build a similar corpus [16].

Profiling the characteristics of the author of Twitter tweets has been a subject that has drawn the attention of several researchers. Rao, et al. works through determining the gender, age, and regional origin based on Twitter tweets and also makes some discoveries on exactly what kinds of features are more commonly used by each gender [10]. For example, Rao, et al. shows that the use of "LMFAO" is a strong indicator of a male

writer. Burger, et al. researched exclusively on gender and was able to accomplish as high as 92% accuracy for gender classification, but they included other fields such as screen name and full name. For simply using Twitter tweets they achieved a 74% accurate classification [4].

1.4 Feature Frequency

The frequency of the features extracted from documents is often used in the construction of the classifier model and in the classification of anonymous documents. For example, if author A uses the word "tree" much more often than author B, an anonymous document with the word "tree" used at a frequency closer to author A than author B will most likely be classified as author A. Now, this is much more significant when working with documents of sufficient length so that words repeat. Micro-documents such as text messages or Twitter tweets have words repeating at a far less frequency than documents such as journal articles. Specifically for Twitter tweets, it was noted by Burger, et al. that in their research the feature frequency did not contribute significantly to the classification of the gender of the tweets [4].

1.5 Structure of Thesis

This thesis goes through testing whether feature frequency is significant in the classification of gender of Twitter tweets, along with some best practice suggestions based on the experiences from these experiments. In Chapter 2, the methods for creating the corpus, the design of the experiments, and the methods of analysis are listed. In Chapter 3, the results of the experiments are listed in table format. These tables show the accuracy and that statistical results of the experiments. Chapter 4 is the discussion of the results of the experiments. Finally, Chapter 5 is the conclusion of this thesis.

Chapter 2

Materials and Methods

2.1 Corpus

The corpus of Twitter tweets was created using recently tweeting users. To get a set of recently tweeting users, tweets from Twitter searches were gathered from the end of December, 2011 to June, 2012. From the tweets, a list of users was gathered. For each user, his or her user profile information was pulled from June to August, 2012. Then, each user's timeline was pulled from August to September, 2012. Users who had protected profiles or whose accounts were deleted were not incorporated in the final corpus.

The self-reported display name from a users profile was used to get the gender of each user. The first set of characters listed was extracted and taken to most likely be the first name. U.S. Census data was then used to determine whether the extracted first name was more likely to be male or female [3]. This was done by determining if a name had a ratio higher than 19 for a certain gender, and if so it was classified to that gender. Any user whose name could not be attributed to a particular gender did not have their tweets incorporated into the gender corpus. The final corpus of tweets contained 466,281 tweets from females and 258,404 tweets from males for a total of 724,685 tweets.

The process of communicating with Twitter to create the corpus was done through Twitter's API. Automation of the communication was done by creating a Java program

which ran at specified intervals using the Unix time-based job scheduler, cron. Calculating the ratio value for the first names was also done using Java along with determining whether the names were male or female. A simple data extraction and reporting language called AWK was used to create several scripts to help manage and manipulate the corpus data.

2.2 JGAAP Modifications

JGAAP did not have any way to run an experiment without feature frequency, so a feature culler was needed in order to perform the experiments for this thesis. The Set Culler was created which takes the existing feature set and creates a new feature set which only contains the first occurrence of each feature.

Additionally, the Nearest Neighbor classifier normally outputs the distance of all the training documents from the specific test document it is currently classifying. This is a problem when you are training and testing on thousands of documents. Since the distance from each training document is not needed for the experiments of this thesis, the listed results was reduced to merely the top 10 closest tweets.

2.3 Pilot Experiments

A set of pilot experiments was performed to test how well commonly used classifiers of the JGAAP system work and to get an initial gauge of how significant feature frequency is in gender classification. These experiments were run using JGAAP with 1,000 tweets which were randomly selected from the whole gender corpus. A 100 tweet subset was used for testing with 900 tweets used for training. The training tweets were balanced between males and females while the test tweets were left unbalanced to reflect the imbalance in the corpus. The test tweets contained 57 female tweets and 43 male tweets.

The first round of pilot experiments were designed to test how well certain classifiers function for classifying gender in general. The first round used Words as features

and ran each classifier with and without the Set Culler. The classifiers tested were: Centroid Driver with Cosine Distance, Nearest Neighbor with Cosine Distance, Markov Chain Analysis, WEKA Decision Stump, WEKA J48 Decision Tree, WEKA Least Median Squared, WEKA Linear Regression, WEKA Multilayer Perceptron, WEKA Naive Bayes, WEKA RBF Network, WEKA SMO, WEKA Voted Perceptron.

From the first set of pilot experiments, WEKA Linear Regression and WEKA Multilayer Perceptron were determined to be too time intensive and Markov Chain Analysis resulted in giving a tie between male and female for almost all of its classifications. These classifiers were left out of the remainder of pilot experiments. The rest of the rounds of pilot experiments use the smaller set of classifiers and experimented on a range of commonly used features: word bigrams, characters, character bigrams, character 5-grams.

2.4 Final Experiments

For the final experiments, another random subset of the gender corpus was selected. This subset totaled 40,000 tweets. The tweets were balanced between male and female so 20,000 tweets were from females and 20,000 were from males. These tweets were then divided into balanced sets of 10,000 tweets. Each set was used as known gender data for a document set with the other three tweet sets becoming the unknown data, or testing data. This produced four document sets, each training on a different 10,000 tweets.

For each document set, a series of experiments was run. These experiments used the subset of classifiers proven to run well from the pilot experiments: Centroid Driver with Cosine Distance, Nearest Neighbor with Cosine Distance, WEKA Decision Stump, WEKA J48 Decision Tree, WEKA Least Median Squared, WEKA Naive Bayes, WEKA RBF Network, WEKA SMO, WEKA Voted Perceptron. Each classifier was run using characters as features and character bigrams as features. Due to memory and time constraints, character 5-grams, words, and word bigrams as features were left out of these experiments except for the experiments using Nearest Neighbor classifier. Nearest Neigh-

bor ran extremely fast and was light enough on memory to be able to be run on the other features.

2.5 Analysis of Results

To analyze the results of the pilot experiments, the accuracy of each experiment was calculated using an AWK program. Each pair of accuracies for a particular feature/classifier combination - with feature frequency and without feature frequency - was compared using a two-tailed, paired t-test to determine if there was any significant difference in accuracy. The times used to determine the speed of the classifiers were taken from the output of the JGAAP experiment engine. The percentage of classifications that resulted in ties was calculated using an AWK program which simply checked for whether the classification value given by both genders was equal.

To analyze the results of the final experiments, a contingency table was generated using an AWK program for each pair of experiments - with feature frequency and without feature frequency. From the discordant pairs of the contingency table, McNemar's test statistic [15] was calculated along with the subsequent p-value. The test statistic is calculated as follows:

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{2.1}$$

With sufficiently large numbers of discordant pairs ($b + c \geq 25$) the statistic has a chi-squared distribution with 1 degree of freedom. This statistic indicates, given a disagreement in classification, whether one experiment in the pair is more likely to give accurate classification than the other or whether it is merely random chance that one experiment gives a more accurate classification than the other. The null hypothesis for the test is that it is merely random chance that one experiment gets the discordant classification correct over the other experiment. A significant p-value results in rejecting this hypothesis.

Some of the experiments in this research contained too small a number of discordant

pairs to use McNemar's test, so no test was performed on those experiments and the results were taken to be indicative of the insignificance of feature frequency in those experiments. With 30,000 tweets being classified for each experiment, 25 or fewer discordant pairs is an insignificant quantity.

Once the p-values from McNemar's test were calculated, the Benjami-Hochberg procedure was used to control the false discovery rate [1] with a significance level of $\alpha = .05$ in order to determine which p-values were significant. This procedure lowers the value of α based on the ranking of the p-value amongst all resulting p-values and the number of experiments performed. With m experiments, if the p-values are ordered in increasing order, $P_{(1)}, P_{(2)}, \ldots, P_{(m)}$, then the Benjami-Hochberg procedure rejects the hypotheses for k where:

$$P_{(k)} \leq \frac{k}{m}\alpha \tag{2.2}$$

The experiments were broken down into three groups: use of characters as features, use of character bigrams as features, use of Nearest Neighbor classifier. Excluding the experiments with a low number of discordant pairs, m equaled 31, 27, and 20 respectively. The Benjami-Hochberg procedure was then used to calculate the significance of the p-values of the experiments. Though 8 of the experiments were cross-listed in two groups, changing m to reflect this did not change the significance of any p-values.

The accuracy in classification was also calculated for each experiment in order to further understand the significance of feature frequency. These accuracies were also used to further understand how well certain classifiers performed on classifying gender of Twitter tweets.

Chapter 3

Results

3.1 Code and JGAAP Output

The two Java programs, all AWK programs, the output from JGAAP, and all files inputted into JGAAP Experiment Engine are publicly accessible on Github at `https://github.com/amkroft/DuqThesis.git`.

3.2 Tables

Table 3.1. Pilot Experiments - Accuracies - Characters, Character Bigrams, and Character 5-grams as Features

	Words		Word Bigrams		Characters		Char Bigrams		Char 5-grams	
Classifier	Regular	Sets	Regular	Sets	Regular	Sets	Regular	Sets	Regular	Sets
WEKA Decision Stump	0.57	0.57	0.57	0.57	0.46	0.50	0.57	0.44	0.58	0.58
WEKA J48 Decision Tree	0.44	0.48	0.44	0.48	0.42	0.48	0.45	0.59	0.61	0.58
WEKA Least Median Squared	0.43	0.43	0.57	0.57	0.61	0.48	0.60	0.59		
WEKA Linear Regression	0.58	0.57								
WEKA Multilayer Perceptron	0.57	0.57								
WEKA Naive Bayes	0.54	0.56	0.59	0.58	0.58	0.50	0.62	0.58	0.60	0.59
WEKA RBF Network	0.43	0.43	0.43	0.43	0.45	0.42	0.43	0.43	0.43	0.43
WEKA SMO	0.52	0.49	0.58	0.61	0.46	0.55	0.57	0.55	0.53	0.54
WEKA Voted Perceptron	0.56	0.56	0.55	0.57	0.43	0.50	0.53	0.52	0.57	0.55
Centroid - Cosine	0.48	0.49	0.55	0.55	0.57	0.57	0.51	0.52	0.59	0.56
Nearest Neighbor - Cosine	0.46	0.52	0.47	0.47	0.46	0.51	0.52	0.53	0.56	0.61

Table 3.2. Pilot Experiments - T-tests

Feature Set	p
Words	0.28907
Word bigrams	0.30688
Characters	0.75809
Character bigrams	0.81335
Character grams N=5	0.69113
Classifier	**p**
WEKA Decision Stump	0.56908
WEKA J48 Decision Tree	0.22179
WEKA Least Median Squared	0.35086
WEKA Naive Bayes	0.22886
WEKA RBF Network	0.37390
WEKA SMO	0.49535
WEKA Voted Perceptron	0.49337
Centroid Driver - Cosine	0.79897
Nearest Neighbor - Cosine	0.04813

Table 3.3. Final Experiments - Accuracies - Characters as Features

	Doc Set 1		Doc Set 2	
Classifier	Regular	Sets	Regular	Sets
WEKA Decision Stump	0.538367	0.538100	0.539000	0.538067
WEKA J48 Decision Tree	0.630600	0.637900	0.604567	0.634567
WEKA Naive Bayes	0.532767	0.530633	0.526767	0.537967
WEKA RBF Network	0.507267	0.503767	0.508067	0.506500
WEKA SMO	0.572900	0.581200	0.575333	0.581533
WEKA Voted Perceptron	0.563467	0.574367	0.568000	0.570133
Centroid - Cosine	0.550500	0.569067	0.556500	0.572100
Nearest Neighbor - Cosine	0.680167	0.689100	0.682200	0.688067
	Doc Set 3		Doc Set 4	
Classifier	Regular	Sets	Regular	Sets
WEKA Decision Stump	0.538467	0.538533	0.536533	0.536800
WEKA J48 Decision Tree	0.606833	0.641067	0.622367	0.641133
WEKA Naive Bayes	0.525333	0.531667	0.521233	0.534967
WEKA RBF Network	0.504133	0.506600	0.509033	0.510633
WEKA SMO	0.574900	0.582667	0.573067	0.582767
WEKA Voted Perceptron	0.566800	0.573700	0.571500	0.575600
Centroid - Cosine	0.554100	0.570133	0.558467	0.575533
Nearest Neighbor - Cosine	0.677933	0.680933	0.683867	0.680933

Table 3.4. Final Experiments - Accuracies - Character Bigrams as Features

	Doc Set 1		Doc Set 2	
Classifier	Regular	Sets	Regular	Sets
WEKA Decision Stump	0.514833	0.514933	0.515367	0.515500
WEKA J48 Decision Tree	0.680267	0.674367	0.679833	0.679633
WEKA Naive Bayes	0.541867	0.544233	0.534100	0.539000
WEKA RBF Network	0.500900	0.500100	0.525067	0.504267
WEKA SMO	0.651433	0.651300	0.653033	0.651967
WEKA Voted Perceptron	0.590333	0.592767	0.592567	0.592800
Centroid - Cosine	0.571767	0.578733	0.576067	0.585533
Nearest Neighbor - Cosine	0.690367	0.699167	0.697567	0.702900
	Doc Set 3		Doc Set 4	
Classifier	Regular	Sets	Regular	Sets
WEKA Decision Stump	0.514700	0.514667	0.514733	0.514700
WEKA J48 Decision Tree	0.674567	0.674433	0.682067	0.680600
WEKA Naive Bayes	0.539267	0.545467	0.543967	0.545667
WEKA RBF Network	0.525567	0.532333	0.500033	0.500067
WEKA SMO	0.650367	0.648200	0.648767	0.650367
WEKA Voted Perceptron	0.590767	0.590400	0.592733	0.594000
Centroid - Cosine	0.576967	0.582400	0.570767	0.578900
Nearest Neighbor - Cosine	0.695767	0.699267	0.694900	0.701733

Table 3.5. Final Experiments - Accuracies - Nearest Neighbor as Classifier

	Doc Set 1		Doc Set 2	
Classifier	Regular	Sets	Regular	Sets
Characters	0.680167	0.689100	0.682200	0.688067
Character bigrams	0.690367	0.699167	0.697567	0.702900
Character 5-grams	0.697433	0.703667	0.703533	0.703400
Words	0.689300	0.687600	0.691133	0.689033
Word bigrams	0.679300	0.679200	0.680967	0.681300
	Doc Set 3		Doc Set 4	
Classifier	Regular	Sets	Regular	Sets
Characters	0.677933	0.680933	0.683867	0.687133
Character bigrams	0.695767	0.699267	0.694900	0.701733
Character 5-grams	0.697567	0.699833	0.701033	0.701500
Words	0.685967	0.686733	0.690167	0.693333
Word bigrams	0.673600	0.674867	0.679233	0.679400

Table 3.6. Final Experiments - Contingency Tables - Characters as Features

Classifier			Doc Set 1 Set Correct	Doc Set 1 Set Incorrect	Doc Set 2 Set Correct	Doc Set 2 Set Incorrect	Doc Set 3 Set Correct	Doc Set 3 Set Incorrect	Doc Set 4 Set Correct	Doc Set 4 Set Incorrect
Decision Stump	Regular	Correct	16115	36	16045	125	16154	0	16080	16
		Incorrect	28	13821	97	13733	2	13844	24	13880
J48 Decision Tree	Regular	Correct	13670	5248	13517	4620	12916	5289	13299	5372
		Incorrect	5467	5615	5520	6343	6316	5479	5935	5394
Naive Bayes	Regular	Correct	8867	7116	14021	1782	13057	2703	14000	1637
		Incorrect	7052	6965	2118	12079	2893	11347	2049	12314
RBF Network	Regular	Correct	14298	920	14005	1237	3466	11658	12773	2498
		Incorrect	815	13967	1190	13568	11732	3144	2546	12183
SMO	Regular	Correct	13585	3602	14147	3113	13781	3466	13819	3373
		Incorrect	3851	8962	3299	9441	3699	9054	3664	9144
Voted Perceptron	Regular	Correct	13204	3700	13426	3614	13066	3938	13389	3756
		Incorrect	4027	9069	3678	9282	4145	8851	3879	8976
Centroid - Cosine	Regular	Correct	12027	4488	12348	4347	11995	4628	12399	4355
		Incorrect	5045	8440	4815	8490	5109	8268	4867	8379
N. N. - Cosine	Regular	Correct	15767	4638	15836	4630	15660	4678	15866	4650
		Incorrect	4906	4689	4806	4728	4768	4894	4748	4736

Table 3.7. Final Experiments - Contingency Tables - Character Bigrams as Features

Classifier			Doc Set 1 Set Correct	Doc Set 1 Set Incorrect	Doc Set 2 Set Correct	Doc Set 2 Set Incorrect	Doc Set 3 Set Correct	Doc Set 3 Set Incorrect	Doc Set 4 Set Correct	Doc Set 4 Set Incorrect
Decision Stump	Regular	Correct	15443	2	15455	6	15436	5	15436	6
		Incorrect	5	14550	10	14529	4	14555	5	14553
J48 Decision Tree	Regular	Correct	15817	4591	15826	4569	15668	4569	16158	4304
		Incorrect	4414	5178	4563	5042	4565	5198	4260	5278
Naive Bayes	Regular	Correct	13334	2922	12969	3054	13233	2945	13473	2846
		Incorrect	2993	10751	3201	10776	3131	10691	2897	10784
RBF Network	Regular	Correct	14731	296	12178	3574	14510	1257	15001	0
		Incorrect	272	14701	2950	11298	1460	12773	1	14998
SMO	Regular	Correct	18106	1437	18134	1457	17905	1606	18016	1447
		Incorrect	1433	9024	1425	8984	1541	8948	1495	9042
Voted Perceptron	Regular	Correct	14665	3045	14855	2922	14647	3076	14667	3115
		Incorrect	3118	9172	2929	9294	3065	9212	3153	9065
Centroid - Cosine	Regular	Correct	14853	2300	14897	2385	15201	2108	14382	2741
		Incorrect	2509	10338	2669	10049	2271	10420	2985	9892
N. N. - Cosine	Regular	Correct	17110	3601	17299	3628	17178	3695	17182	3665
		Incorrect	3865	5424	3788	5285	3800	5327	3870	5283

Table 3.8. Final Experiments - Contingency Tables - Nearest Neighbor as Classifier

Features			Doc Set 1 Set Correct	Doc Set 1 Set Incorrect	Doc Set 2 Set Correct	Doc Set 2 Set Incorrect	Doc Set 3 Set Correct	Doc Set 3 Set Incorrect	Doc Set 4 Set Correct	Doc Set 4 Set Incorrect
Characters	Regular	Correct	15767	4638	15836	4630	15660	4678	15866	4650
		Incorrect	4906	4689	4806	4728	4768	4894	4748	4736
Character bigrams	Regular	Correct	17110	3601	17299	3628	17178	3695	17182	3665
		Incorrect	3865	5424	3788	5285	3800	5327	3870	5283
Character 5-grams	Regular	Correct	19640	1283	19624	1482	19603	1324	19503	1528
		Incorrect	1470	7607	1478	7416	1392	7681	1542	7427
Words	Regular	Correct	18523	2156	18605	2129	18490	2089	18702	2003
		Incorrect	2105	7216	2066	7200	2112	7309	2098	7197
Word bigrams	Regular	Correct	20043	336	20203	226	19991	217	20163	214
		Incorrect	333	9288	236	9335	255	9537	219	9404

Table 3.9. Final Experiments - McNemar χ^2 Statistics - Characters as Features

	Doc Set 1	Doc Set 2	Doc Set 3	Doc Set 4
WEKA Decision Stump	1.00000	3.53153		1.60000
WEKA J48 Decision Tree	4.47606	79.88166	90.88574	28.03299
WEKA Naive Bayes	0.28910	28.94769	6.45104	46.05100
WEKA RBF Network	6.35447	0.91018	0.23412	0.45678
WEKA SMO	8.31893	5.39551	7.57697	12.03368
WEKA Voted Perceptron	13.83836	0.56171	5.30113	1.98153
Centroid - Cosine	32.54474	23.90570	23.76101	28.42594
Nearest Neighbor - Cosine	7.52557	3.28275	0.85751	1.02192

Table 3.10. Final Experiments - McNemar χ^2 Statistics - Character Bigrams as Features

	Doc Set 1	Doc Set 2	Doc Set 3	Doc Set 4
WEKA Decision Stump				
WEKA J48 Decision Tree	3.47907	0.00394	0.00175	0.22606
WEKA Naive Bayes	0.85224	3.45468	5.69388	0.45290
WEKA RBF Network	1.01408	59.68363	15.16710	
WEKA SMO	0.00557	0.35531	1.34255	0.78314
WEKA Voted Perceptron	0.86468	0.00837	0.01970	0.23038
Centroid - Cosine	9.08318	15.95884	6.06737	10.39749
Nearest Neighbor - Cosine	9.33512	3.45200	1.47098	5.57731

Table 3.11. Final Experiments - McNemar χ^2 Statistics - Nearest Neighbor as Classifier

	Doc Set 1	Doc Set 2	Doc Set 3	Doc Set 4
Characters	7.52557	3.28275	0.85751	1.02192
Character Bigrams	9.33512	3.45200	1.47098	5.57731
Character 5-grams	12.70214	0.00541	1.70250	0.06384
Words	0.61042	0.94613	0.12592	2.20068
Word Bigrams	0.01345	0.21645	3.05932	0.05774

Table 3.12. Final Experiments - P-values - Characters as Features

Classifier	Doc Set 1	Doc Set 2	Doc Set 3	Doc Set 4
WEKA Decision Stump	0.31731	0.06021		0.20590
WEKA J48 Decision Tree	0.03437	3.9752E-19	1.52212E-21	1.1926E-07
WEKA Naive Bayes	0.59080	7.4359E-08	0.01109	1.1521E-11
WEKA RBF Network	0.01171	0.34007	0.62849	0.49913
WEKA SMO	0.00392	0.02019	0.00591	0.00052
WEKA Voted Perceptron	0.00020	0.45357	0.02131	0.15923
Centroid - Cosine	1.1648E-08	1.0117E-06	1.0907E-06	9.7352E-08
Nearest Neighbor - Cosine	0.00608	0.07001	0.35444	0.31206

Table 3.13. Final Experiments - P-values - Character Bigrams as Features

Classifier	Doc Set 1	Doc Set 2	Doc Set 3	Doc Set 4
WEKA Decision Stump				
WEKA J48 Decision Tree	0.06215	0.94994	0.96662	0.63446
WEKA Naive Bayes	0.35592	0.06307	0.01702	0.50096
WEKA RBF Network	0.31393	1.1140E-14	9.8403E-05	
WEKA SMO	0.94048	0.55112	0.24659	0.37618
WEKA Voted Perceptron	0.35243	0.92708	0.88837	0.63124
Centroid - Cosine	0.00258	6.4735E-05	0.01377	0.00126
Nearest Neighbor - Cosine	0.00225	0.06318	0.22519	0.01819

Table 3.14. Final Experiments - P-values - Nearest Neighbor as Classifier

Features	Doc Set 1	Doc Set 2	Doc Set 3	Doc Set 4
Characters	0.00608	0.07001	0.35444	0.31206
Character Bigrams	0.00225	0.06318	0.22519	0.01819
Character 5-grams	0.00037	0.94139	0.19196	0.80052
Words	0.43463	0.33071	0.72270	0.13795
Word Bigrams	0.90766	0.64176	0.08028	0.81011

Chapter 4

Discussion

4.1 Feature Frequency

The results of the pilot experiments proved that initially the significance of feature frequency depends on the type of feature being used and the type of classifier. But, when analyzing the results of the final experiments, it can be seen that feature frequency was consistently insignificant in the classification of Twitter tweets. For the majority of the results, there was no significant difference between using feature frequency and not using feature frequency. For 23 of the 25 experiments where there was a significant difference (mostly when using characters as features), it can be seen that dropping feature frequency actually significantly improved the classification accuracy. Therefore it is beneficial to use the Set Culler of JGAAP to remove frequency of features when classifying gender of Twitter tweets.

The reason for this insignificance is an interesting topic. Because Twitter tweets are so short, it is possible that features rarely repeat often enough to significantly affect classification. This is more so when using character N-grams or words as features since Twitter only allows its tweets to be 140 characters. The experiments which were run on character bigrams, character 5-grams, words, and word bigrams showed few experiments with a significant difference in classification. If there are few repeated features, then

feature frequency would not be useful in classification.

For the experiments which were run on characters as features, there were significantly more experiments which produced significant p-values. This shows that at the character level, there was enough frequency of features to affect classification, though in a negative way. This could be that the frequency of features is not indicative of gender and is merely random noise. Trying to train a classifier with feature frequency might be producing overfitting which would hinder classification.

4.2 Classifiers

The results of the pilot experiments showed that certain classifiers either do not perform well or perform very slowly on the classification of gender of Twitter tweet. WEKA Linear Regression and Multilayer Perceptron proved to be incredibly slow to run and Markov Chain Analysis showed to give very poor results. Therefore these classifiers should not be used in future experiments on Twitter tweets with the JGAAP system.

Nearest Neighbor classifier is the best classifier for the experiments from this thesis. It produced the highest accuracy of all the classifiers tested, across all features. This shows that Nearest Neighbor is the best classifier to use in the JGAAP system for classifying gender of Twitter tweets. Nearest Neighbor consistently produced about a 68% accuracy of classification. Combine this with the fact that Nearest Neighbor was the only classifier to run fast enough to classify based on the full list of features. This can be attributed to the low computation needed by Nearest Neighbor to classify a document. The training of the WEKA classifiers is a huge computational task when working across thousands of documents and therefore not efficient enough for classifying Twitter tweets.

Two WEKA classifiers produced accuracies in classification close to those of Nearest Neighbor. WEKA J48 Decision Tree's accuracy was only 2-6% lower than Nearest Neighbor and WEKA SMO only 5-10% lower in classification. These differences were smaller when using Character bigrams as features. Therefore, these two classifiers would make a

sufficient second to using Nearest Neighbor driver for classification of gender of Twitter tweets.

From the final experiments, it can be seen that WEKA RBF Network should not be used for the types of experiments of the type performed in this thesis. The WEKA RBF Network consistently performed the worst among all classifiers tested and rarely gave a classification accuracy above baseline. Also, the significance of feature frequency in using this classifier is yet to be determined. In the experiments from this thesis, the significance of feature frequency was inconsistent across the different document sets used. Overall, WEKA RBF Network performed the worst.

4.3 Future Work

Since feature frequency proved insignificant for classifying gender, it stands to reason that there might be other author characteristics that are also insignificant in classifying. Since other characteristics of the author are available through Twitter, such as time zone, these experiments could be expanded to be run on other characteristics.

Twitter tweets are similar in structure and size to other types of written data, such as text messages. Future work could include researching whether these methods work well for classifying the gender of the author of text messages or other micro-text documents.

Since gender can be classified at an accuracy above baseline, it would be interesting to research into which features are most significant in the classification. These features could aid in the classification of larger types of documents, such as emails or blog posts.

Chapter 5

Conclusion

With social media being a significant pastime of most internet users, researching into authorship attribution and profiling using social media data has become a large research topic. Twitter, one of the more popular social media data, has been at the front of research on social media. Particularly, profiling the gender of the author has received a lot of attention from author profiling researchers. Previous work on the topic shows that using several fields provided by the author, up to a 91% accuracy can be obtained.

Using Nearest Neighbor classifier with character bigrams from JGAAP can profile the gender of the author of a single Twitter tweet at a 70% accuracy rate. Classifiers WEKA RBF Network, WEKA Linear Regression and WEKA Multilayer Perceptron proved to be classifiers that should not be used in profiling gender of Twitter Tweets. Furthermore, this thesis showed that the frequency of features from Twitter tweets proved either insignificant in the classification of gender or hindered classification of gender and therefore it can be removed from future research to increase speed and accuracy.

BIBLIOGRAPHY

[1] Yoav Benjamini and Yosef Hochberg, *Controlling the false discovery rate: a practical and powerful approach to multiple testing*, Journal of the royal statistical society. Series B (Methodological) **57** (1995), 289–300.

[2] Shea Bennett, *There are now more than 200 million active twitter users*, WebMediaBrands, Inc., 12 2012. http://www.mediabistro.com/alltwitter/200-million-twitter-users_b32986

[3] U.S. Census Bureau, *Genealogy data: Frequently occurring surnames from census 1990 - names files*, Author, Suitland, MD, 1990. http://www.census.gov/genealogy/www/data/1990surnames/names_files.html

[4] John D. Burger, John Henderson, George Kim, and Guido Zarrella, *Discriminating gender on Twitter*, Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processings (2011), 1301–1309.

[5] The Nielson Company, *What Americans do online social media and games dominate activity*, Author, New York, NY, 8 2010. http://www.nielsen.com/us/en/newswire/2010/what-americans-do-online-social-media-and-games-dominate-activity.html

[6] O. de Vel, A. Anderson, M. Corney, and G. Mohay, *Mining e-mail content for author identification forensics*, ACM SIGMOD **30** (2001), 55–64.

[7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, *The Weka data mining software: An update*, SIGKDD Explorations **11** (2009).

[8] Evaluating Variations in Language Lab, *The java graphical authorship attribution program*, Author, Pittsburgh, PA. http://evllabs.com/

[9] Patrick Juola, *Authorship attribution*, Foundations and Trends in Information Retrieval **1** (2008), 233–334.

[10] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta, *Classifying latent user attributes in Twitter*, SMUC '10 Proceedings of the 2nd international workshop on Search and mining user-generated contents, 2010, pp. 37–44.

[11] Milton Rokeach, Robert Homant, and Louis Penner, *A value analysis of the disputed Federalist papers*, Journal of personality and social psychology **16** (1970), 245–250.

[12] Rui Sousa Silva, Gustavo Laboreiro, Luís Sarmento, Tim Grant, Eugénio Oliveira, and Belinda Maia, *'twazn me!!! ;(' automatic authorship analysis of micro-blogging messages*, Natural Language Processing and Information Systems (2011), 161–168.

[13] Tumblring.net, *Tumblr post limits*, 4 2010. http://tumblring.net/tumblr-post-limits/

[14] Twitter, *The fastest, simplest way to stay close to everything you care about*, Author, San Francisco, CA, 2013. https://twitter.com/about

[15] John Uebersax, *McNemar tests of marginal homogeneity*, John Uebersax Enterprises LLC, California, 8 2006. http://john-uebersax.com/stat/mcnemar.htm

[16] Joshua Weissbock and Taylor Singletary, *Can i release mined tweets for research?*, Twitter, San Francisco, CA, 2012. https://dev.twitter.com/discussions/8232