Author Gender Identification from Text

Atoosa Mohammad Rezaei

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
July 2014
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Cem Ergün
Supervisor

Examining Committee

1. Asst. Prof. Dr. Yıltan Bitirim
2. Asst. Prof. Dr. Cem Ergün
3. Asst. Prof. Dr. Gürcü Öz

ABSTRACT

The identification of an author's gender from a text has become a popular research area within the scope of text categorization. The number of users of social network applications based on text, such as Twitter, Facebook and text messaging services, has grown rapidly over the past few decades. As a result, text has become one of the most important and prevalent media types on the Internet. This thesis aims to determine the gender of an author from an arbitrary piece of text such as, for example a journal article or email. This field of research has garnered the interest of the researchers for the reason that some people fake their gender in text-based Internet forensics.

The psychology of linguistic indicates how closely the words and writing styles people use correlate with their gender. Various feature sets have been used by researchers in recent decades to identify the gender of an author; however, identifying feature sets remains a research obstacle. In this dissertation, five feature sets were selected to prepare a feature space for the gender identification problem. The features in these sets included character-based features, word-based features, syntactic-based features, structure-based features and the function words that an author used in a text.

Two state-of-the-art machine learning algorithms were considered for the author gender identification problem, based on the proposed feature space in this thesis. Weka (data mining software) was used to design a support vector machine classifier and a Bayesian logistic regression classifier. The reason for choosing these two classifiers was that support vector machine and Bayesian logistic regression are the most powerful classifiers for text mining.

An Enron email dataset, which is available to researchers on the Internet, was used in the training and testing phases during experiments to provide sufficient data for the classification process.

Keywords: Machine Learning, classifier, psychology linguistic, Support Vector Machine, Bayesian logistic regression, gender identification

ÖZ

Metinden yazar cinsiyetinin belirlenmesi, metin sınıflama kapsamında yaygın bir araştırma konusu olmuştur.Metin tabanlı sosyal medya uygulamalarındaki kullanı sayısı son yıllarda hızla artmıştır.Sonuç olarak metin, internet üzerindeki en önemli ve yaygın medya haline gelmiştir.Bu çalışmada, rastgele seçilmiş metin parçalarından, örneğin makale veya e-posta yazarının cinsiyeti belirlenmiştir.Bu çalışma alanı, araştırmacıların ilgisini çekmiştir çünkü bazı kişiler metin tabanlı internet ortamında cinsiyetlerini saklamaktadırlar.

Dil psikolojisi, yazarın cinsiyeti ile kullandığı kelimelerin ve yazım şeklinin çok yakından ilişkili olduğunu göstermektedir.Geçtiğimiz on yılda, araştırmacılar yazar cinsiyetini belirlemek için çeşitli özellik kümeleri kullanmışlardır.Bununla beraber özellik kümelerinin belirlenmesi zorluğunu korumaktadır. Bu çalışmada, cinsiyet belirleme problemi için hazırlanan özellik uzayı; beş özellik kümesi seçilerek oluşturulmuştur. Kümelerdeki özellikler karakter tabanlı özellikler, kelime tabanlı özellikler, sözdizimsel özellikler, yapısal özellikler ve bir yazarın metinde kullandığı işlev kelimelerden oluşmaktadır.

Bu çalışmada, yazar cinisiyeti belirleme problem için, sunulan özellik uzayında, iki en yeni makine öğrenmesi algoritması kullanılmıştır. Bir Destek Vektör Makinası sınıflayıcı ve bir Bayes lojistik regresyon sınıflayıcısı tasarlamak için Weka (veri madenleme yazılımı) kullanılmıştır. Bu iki sınıflayıcının seçilmesinin nedeni, metin madenciliği için destek vektör makinası ve Bayes lojistik regresyonun en güçlü sınıflayıcılardan olmasıdır.

Sınıflama sürecinde kullanılan veriler internetten sağlanmıştır.Araştırmacılar için bağışlanan Enron e-posta veri kümesi, denemeler sırasında eğitim ve test fazlarında kullanılmıştır.

Anahtar Kelimeler: Makine öğrenme, sınıflandırıcı, dilsel psikoloji, Destek Vektör Makinesi, Bayes lojistik regresyon, cinsiyet belirleme

ACKNOWLEDGMENT

I would like to express my deepest appreciate to my supervisor Asst. Prof. Dr. CemErgün for encouraging my research and for letting me grow. Without your supervision and helpful guidance this dissertation would not have been completed.

I would also like to thank my committee members for serving me as committee members and letting my defense become memorable for me and also all the staff and members of computer engineering department that without their cooperation I could not achieve the results in this dissertation.

Above all I would like to thank my family who sponsored me and devoted their love and unconditional support throughout my degree. I love them all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The rapid growth of the Internet has created innumerable ways for sharing information in cyberspace. The number of social network users, ecommerce and other web applications are increasing daily. This growth has given rise to a variety of misuses, as anonymity is a significant characteristic of Internet-based communities[1]. Users might not reveal their true identities in terms of name, age, gender and address in cyberspace. Therefore, it has become important to design an efficient method for identity-tracing in the field of cyberspace forensics.

Gender identification is always of importance, as gender is a category of identification can be misused in various instances including email forgery, online communities', forensic matters, marketing, etc. From a marketing perspective, for example, companies need to know what product is more successful among what gender and can do this by analysing reviews on blogs and social networks [2].

Psychology research has revealed that the words an individual uses can specify their mental/physical health and emotion[3][4]. Moreover, each person has their own stylistic tendencies; this is referred to as their author profile. With the development of computers, stylometry has been widely used for identifying authorship. Over 1000 stylometric features have been proposed to date, including word or character-based stylometric features, function words and punctuation [3]. For this thesis, my research has been narrowed down to investigate short text documents and extract from them features that potentially divide authors into male or female classes.

The author gender identification problem is a classification problem with two classes. When submitting a text, it should be assigned to class one if the author is female or class two if the author is male. To design such a classifier we need to extract feature sets from the text that remain the same for most authors of the same gender[4]. In general, the gender identification process is divided into four steps:

1) Collecting a suitable corpus of textual messages to make up the dataset

2) Identifying features that are significant indicators of gender

3) Automatically extracting feature values from each message

4) Building a classification model to identify the author of a candidate text message's gender

Figure 1 shows the process of gender identification from texts that are available over Internet.

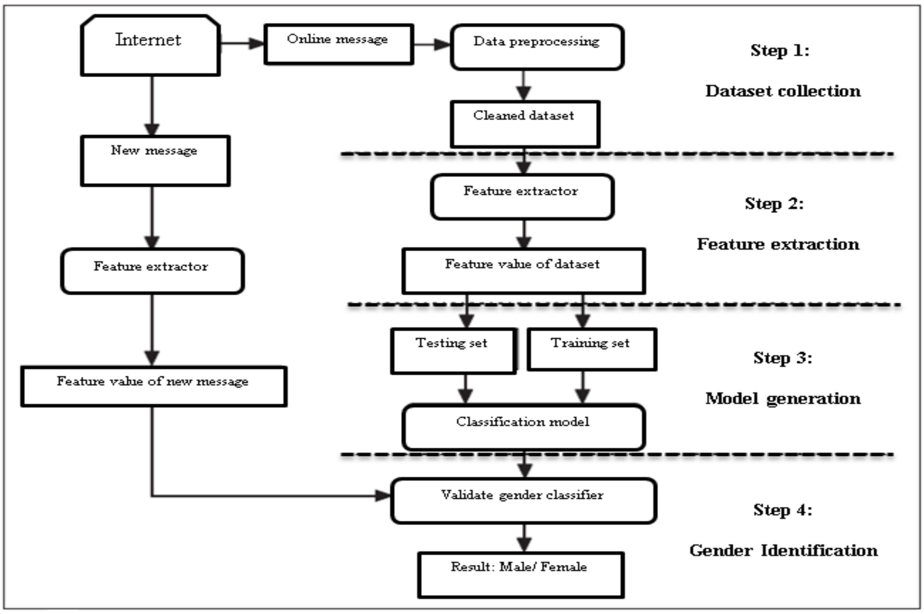


Figure 1: Gender identification process

Following is the thesis organized in different chapters. In Chapter 2, the dataset preprocessing issues are explained. In Chapter 3, selecting the feature set will be discussed following by Chapter 4, automatic feature extraction will be reviewed. In Chapter 5 classification techniques are discussed. In Chapter 6 the results of the experimentations will be demonstrated with analysis. Finally last chapter is the conclusion of the thesis.

Chapter 2

DATA PREPROCESSING

2.1 Corpus

The aim of this thesis was to state whether the author of a submitted text was male or female. In this regard, there was the need for a dataset containing various writing samples divided according to gender type, which made it easier for us to extract each gender's writing styles. Different datasets were available to the researchers, each of which had different categories considering age, gender, the type of text, etc.

The Reuters news group dataset[5] is one of the most widely used datasets for author verification from text. Reuters is the largest global multimedia news agency. Reuters Corpora was made available in 2000 to researchers in fields such as text mining, natural language processing and machine learning systems. This corpus contains all reports that had been written by Reuters journalists between August 1996 and August 1997, and was released in May 2005. Since the Reuters dataset contains news articles, their authors would have paid more attention to grammar and punctuation, unlike texts that are exchanged when, for example, writing a text message to a friend. In short, when writing textual messages, people tend to not obey grammatical or punctuation rules the way they do when writing more formal texts.

The PAN data set[6] is another corpus that is available to researchers. This corpus consists of XML documents containing writing in HTML format. Many different topics are grouped by author and labelled with his/her language, gender and age group. The documents are divided into two languages (English and Spanish), two genders (male and female) and three groups of age (10s: 13-17 years, 20s: 23-27 years and 30s: 33-47 years).

Since this thesis focused on short textual messages to identify the gender of the author, we required using a dataset that contained writing samples divided by gender. Additionally, texts that are written in a more informal style are more suitable for this type of research. As a result, we used the Enron[7] dataset for gender identification.

2.2 Enron Dataset

The Enron dataset is a collection of emails from roughly 150 Enron employees, mostly senior managers and were categorized into folders that had been named by the authors of the emails. Thanks to Leslie Kaelbling at MIT and Melinda Gervasio, who corrected this dataset, the Enron dataset is available via the Internet to researchers. Attachments, as well as some of emails have been removed at the request of the employees who had written them. The Enron dataset currently contains 619,466 emails[7]. After removing many duplicates, which had been stored in different folders but primarily in the “all document” folder and that were also available in other folders such as “sent”, “sent emails” and “received emails”, 200399 emails from 158 users remained in the trimmed corpus. As this corpus is almost the only dataset made up of real emails written in different categories (contracts, announcements, invitations or even chatting with a friend) and has thousands of samples, we chose this particular dataset as our training and testing dataset.

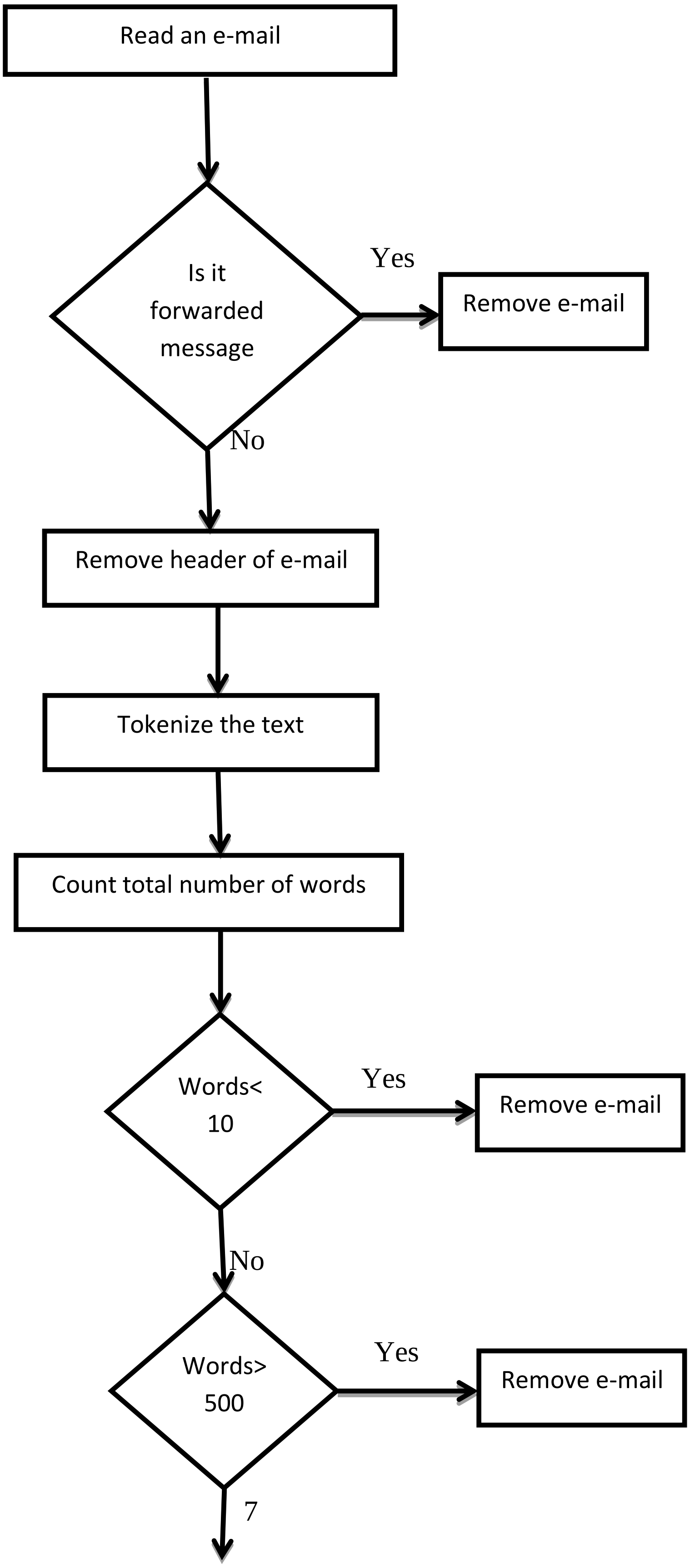
2.3 Data Preparation

To automatically extract those emails with the desired length from the dataset and that had been written by a particular gender, emails were first divided into two separate folders, i.e., female and male. Indicating the gender of the author was done based on the name of the folders, manually referenced by name dictionaries [8][9]. The emails of authors who had unisex names were removed from the list. Finally, 150 authors remained on the list.

The gender of the original message's author’s was in some cases different from the individual who had forwarded the message; for example, a woman sent an email to a man and asked him to forward the message to other employees; as such, this email was available in his sent items, but the gender of the original author was different from the individual who forwarded that email. To prevent this contention, only emails were chosen where we were sure about the writer’s gender.

The next step was to remove all email headers, which contains information about the sender, receiver, date, time, subject, Cc information and Bcc information and to access only the body of the emails. In this regard, Java codes were used and the trimmed dataset was stored in relevant folders.

In the next step, the number of words in emails was counted. The length of the emails available in the Enron dataset varies from one word to thousands of words; therefore, we chose emails that were at least ten words in length to at most five hundred words in length, since the aim of the thesis was to assess short text examples such as casual conversations and in general, these texts were not very long. Once again using Java codes, we counted the amount of words in each email; if the length of emails were more than 10 and less than 500 words, this filter stored them in one of two folders, depending on the gender of their authors. We chose about 14 000 emails for the dataset. In this stage, features could be extracted from the prepared dataset. Figure 2 contains a flowchart for the data preparing process.



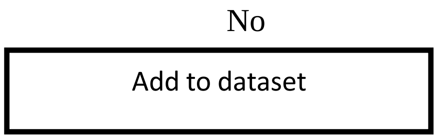


Figure 2: Preparing Dataset

2.4 [2]Training and Testing Dataset

Depending the way of text-learning, a part of the dataset should be considered for the training phase, while the remaining part should be considered for the testing phase within the context of a machine learning system. Many different corpuses are available to researchers and have been categorized based on a collection of characteristics.

These days, texts that are exchanged across the Internet play an important role in our society and have become one of the most common media communication tools for individuals from many different walks of life. As there is no real way of establishing an email's authenticity, research has sought to find a means for detecting forged emails[2]. In an attempt to hide the identity of the sender, emails can be re-routed between many different anonymous servers, so that in some cases, the only way of establishing the original sender is to do so through the writing style and structure of the email. In this research, some structural and linguistic features have been applied in a bid to train a learning engine.

In the case of email forgery, to train a machine learning system, there is a need for a corpus containing a variety of writing samples from different authors that can be extracted to prepare a training dataset. Machine learning systems can infer an email forgery after applying this labelled dataset by comparing the forged email's writing style to the writing styles that are already available[10]. This can result in the system making a judgment about whether a text was written by a particular person.

Writing style changes from one individual to another and there are a wide variety of different writing styles; therefore, specific categorization cannot be defined for predicting the author of an email. Figure 3 presents a schematic for the author verification process[2].

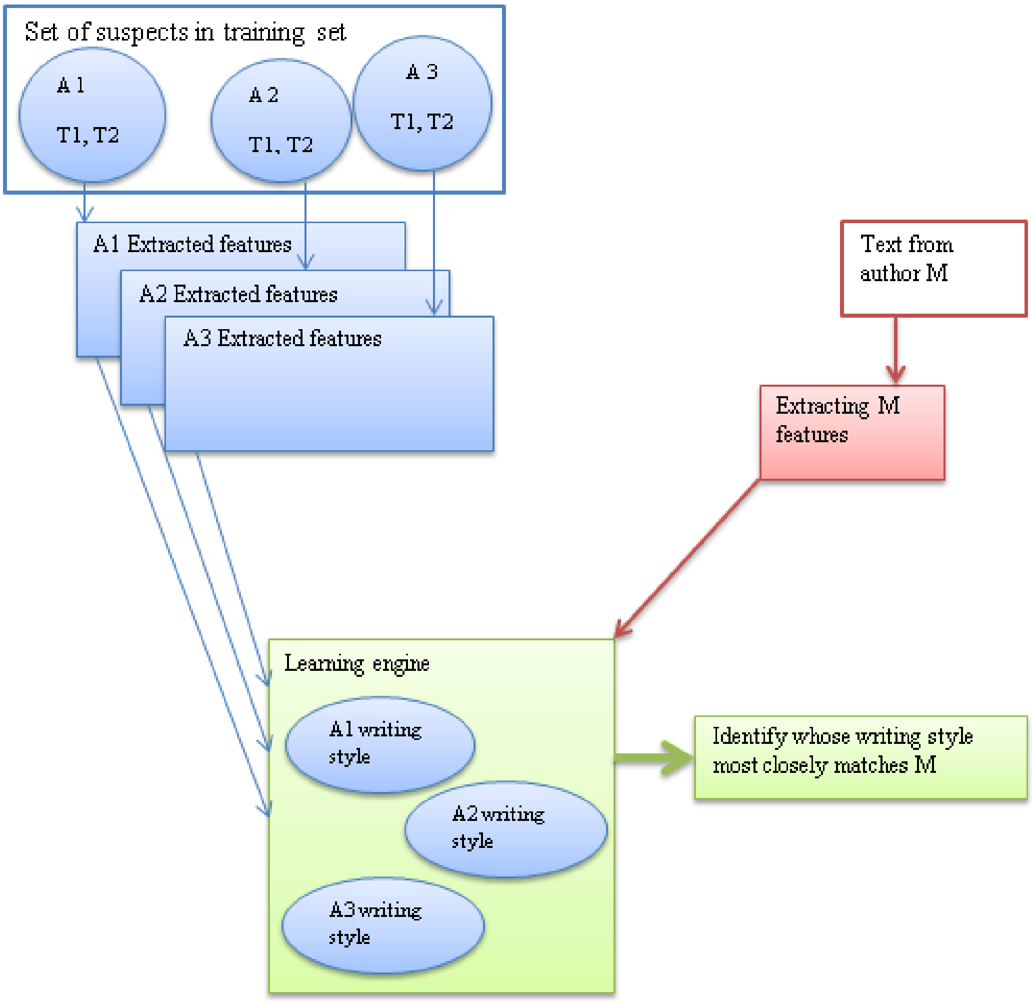


Figure 3: Process of Identifying the Author of Suspected E-Mail

This schematic shows that in order to presume the author of a submitted email, different sets of emails are provided and each set belongs to a suspect. These sets are then used as a training dataset. The features of each set are extracted for the machine learning engine, which extracts each author’s writing style. The process of identifying the author of a suspected email is to first extract the features of that email in order to determine the writing style and then comparing this writing style to the available writing styles in order to predict the author of the submitted email.

Unlike the mentioned case that is shown in Figure 3, there are cases where researchers have needed to access a dataset containing many emails written by various authors categorized by gender and age. For example, in a study aiming to predict the gender and age of blog writers [11], researchers used the PAN dataset, a corpus consisting of writing samples by non-anonymous writers. This dataset is a collection of essays, reviews and newspaper articles consisting of hundreds to thousands of words. Table 1 indicates the distribution of blog authors from the dataset on the basis of gender and age (retrieved from bloggers.com, 2004 statistics). These results show that the majority of bloggers under the age of 18 were female, while male bloggers were the majority among older bloggers.

Table 1: Blogs Distribution Over Age and Gender

Age range	Gender	
	Female	Male
Unknown	12287	12259
13-17	6949	4120
18-22	7393	7690
23-27	4043	6062
28-32	1686	3057

33-37	860	1827
38-42	374	819
43-48	263	584
>48	314	906
Total	43169	71493

Chapter 3

FEATURE SET DESCRIPTION

What are good linguistic features that differentiate female writers from their male counterparts? Differences in the techniques that women and men use to express the same subject have been of interest to many researchers in the field. The past few decades have demonstrated significant changes in terms of how women and men use language. For example, in a text containing sport- related words such as 'cricket', 'beat', 'champion'', coach' and 'league', it has been found that the author of a text containing these words will most likely be male rather than a female. On the other hand, for a text containing words such as 'pink' and 'boyfriend', the probability of the writer being female appears to be increasing[10].

When analysing the age criterion, researchers have found that teenagers tend to write about friends and moods, where individuals in their 20s mostly write about their college lives; those in their 30s are more likely to write about marriage, work and politics.

Different genders use many of the same words in their writing, but with different intentions. For example, when males talk about 'daily life', they tend to mean their work; when females use the same phrase, they are more likely to be discussing love and the more spiritual aspects of life. Another example concerns the use of the word “dress”, which males tend to use for tuxedos, while females use the word when talking about bridal gowns and evening dresses.

As with gender, different age groups tend to write about different topics. Based on existing research, this dissimilarity can be reviewed from the perspective of human psychology. Generally, scientists have studied how the different genders talk, the dissimilarities within their speech, grammatical features, intonation, etc. Robin Lakoff, a contributor to feminist linguistics, believes that females use weaker and more sweet-sounding words such as 'dear' and 'oh my goodness', while males tend to use stronger words such as 'damn' (Braun, 2004: 13). On the other hand, there are words that both genders use, but with different frequency.

For example women tend to use intensifying adverbs like 'very' or 'really' and multiple question marks in their writing. Generally, in their conversations, women make indirect orders while men tend to use more directives; women tend to converse more closely to standard grammatical language than men, who talk more dialectical. For example, when a woman wants to ask others out to dinner, she might write, “Does anybody wants to go out for dinner???” On the other hand, a man might write, “Let’s go out for dinner”. The length of the sentence is another feature that can be used as a measure to differentiate between genders; sentences written by females are generally longer than those written by men. In terms of subject, women talk more about personal and emotional aspects than men, who tend to talk more about fact-based and less dramatic subjects[12].

Stylometry[13] is the study of how people judge others according to their writing style. Stylometry can not only be used to identify a writing style, but can also assist in identifying the gender of the author. The following section discusses the features that can assist in separating writing on the basis of gender by using different stylometrics. All the extracted features are available in appendix 1.

3.1 Character-Based Features

This section discusses the text analysis by considering each of the characters included therein. The text included 27 stylometric features that have been widely used in author attribution studies[13]. First, we counted the total number of characters, including all the letters, digits, punctuations, spaces, etc. The other stylometric analysed in this part is the total number of letters, including all uppercase and lowercase letters (a-z, A-Z). Along with the total number of uppercase characters (A-Z), the total number of digits was counted (0-9), all white spaces were counted, as well as the total number of special characters. Table 2 illustrates the text's character-based features[12][14][15].

Table 2: Character-Based Features

Feature	**Description**
Total number of characters	Alphabet, digits, special characters
Total number of letters	a-z and A-Z
Total number of upper characters	A-Z
Total number of digital characters	0-9
Total number of white space characters	White space
Total number of special characters (22 feature)	“, #, $, %,&, (,), *, +, _, /, <, =, >, @, \, ^, _, {,}, |, ~

3.2 Word-Based Features

This part discusses the analysis of words by applying 11 statistical measures [16], including total number of words, the average number of characters per word, the total number of different words that are available in a text and the total number of words with at most three characters.

Hapaxlegomena[17] are another measure for indicate the total number of words that do not iterate throughout the entire text. Hapaxlegomena imply that a single word occurs once in a specific text by a particular author; it does not infer that the word has been used only once in all of the author's writings. The author may make use of this special word in other writings. Hapaxdislegomena[17] is another evaluation measure, which refers to double occurrences. As with hapaxlegomena, these words can be used in other writings of an author, but they may only occur twice in a specific text.

Another measure that assists in the evaluation of vocabulary richness is Yule’s K measure[18], which represents the diversity of words that a writer has used in a text. Yule’s K measure is calculated using Equation 3.1.

$$\text{Yule's K}=10^4\left(-\frac{1}{N}+\sum_{i=1}^{V} V\text{i}\;\left(\frac{i}{N}\right)^2\right) \qquad \textbf{(3.1)}$$

Simpson’s D [19] is a measure indicating that if we randomly select two words from a text, how large the probability is of selecting the same words. If the result is zero, it indicates infinite diversity when no diversity is meant at all; thus, the smaller the value of Simpson’s D measure, the higher the diversity. Using Equation 3.2, Simpson’s D measure can be calculated.

$$Simpson'\text{D}=\sum_{i=1}^{V} Vi\,\frac{i}{N}\frac{i-1}{N-1} \qquad \textbf{(3.2)}$$

Honore's R measure [20] has been used to evaluate the richness of text. This measure indicates that if the hapaxlegomena value is bigger, the text will be richer. Honore's R measure generates the richness of a text by considering the number of words that occur once in the text as a proportion of the total number of words, as shown by Equation 3.3.

$$Honore's\ R = \frac{100 \log_{10} N}{1 - \frac{Hapax \quad Legomena}{V}}) \qquad \textbf{(3.3)}$$

This section also measures entropy using Equation 3.4 to evaluate the randomness of the data.

$$\text{Entropy} = \sum_{i=1}^{N} Vi \left(-\log_{10} \frac{i}{N} \right) \frac{i}{N} \qquad \textbf{(3.4)}$$

Word based features are shown in table3.

Table 3: Word-Based Feature

Feature	**Description**
Total number of words	Total number of all words in the text
Average length per word	In characters
Vocabulary richness	Total number of different words
Total number of long words	Words longer than 6 characters
Total number of short words	1-3 character words
Hapaxlegomena	Words that occurs only one
Hapaxdislegomena	Words that occurs only twice
Yule's K measure	Measure of vocabulary richness
Simpson's D measure	Measure of diversity
Honor's R measure	Measure of vocabulary richness
Entropy measure	Measure of disorder of set of data

In recent decades, researchers have found that the words that people use can be correlated to their physical and mental health situations[21][22]. Evidence has shown that professional authors use more positively-inclined words like 'beautiful', 'love', 'pretty' and only a modest number of negative emotions like 'hate' and 'nasty', as well as cognitive words such as 'know' and 'because'. Moreover, they change the pronunciation of these words from one part to another part of the document [23]. LIWC (Linguistic Inquiry and Word Count) is software was created by James W. Pennebaker, Roger J. Booth and Martha E. Francis, in which the authors have categorized thousands of words into 68 categories. When submitting a text to LIWC, the software output provides the amount of words that a writer has used in each of the 68 categories[23].

We have considered these 68 categories as part of the word-based feature extraction in this thesis.Table 4 demonstrates some of LIWC features set.

Table 4: LIWC Feature Sample

Features	**Some of words in the feature**
Assent	Agree, Ok, Never
Certainty	Never , Always
Tentative	Guess, Perhaps, Maybe
Insight	Consider, Think, Know
Negation	Not, Never, No
Sadness	Sad, Cry, Grief
Positive emotions	Sweet, Love, Nice
Anger	Annoyed, Hate, Kill
Negative emotions	Nasty, Ugly, Hurt
Anxiety	Nervous, Fearful, Worried

3.3 Syntactic-based features

Syntactic features extract a writer’s writing style by considering the sentences therein. In this regard, we counted the total number of single quotes, commas, periods, semicolons, question marks, multiple question marks, exclamation marks, multiple exclamation marks and ellipses to establish how often females and males used punctuation in their writing.

In informal writing, it is common to use multiple question marks or exclamation marks to better express a feeling or mood. Women tend to use more multiple question marks than men[24]. Table 5 shows the syntactic features.

Table 5: Syntactic-Based Features

Feature	**Description**
Total number of single quotes	‘
Total number of commas	,
Total number of period counters	.
Total number of colons	:
Total number of semi-colons	;
Total number of question marks	?
Total number of multiple question marks	More than one question mark
Total number of exclamation marks	!
Total number of multiple exclamation marks	More than one exclamation mark
Total number of ellipsis	...

3.4 Structurally-Based Features

People have different habits in terms of organizing the layout of their writing; this might relate to how they switch to another paragraph or the length of their paragraphs. This dissertation investigated short texts exchanged across the Internet. The most outstanding feature of these types of texts was that they were flexible in terms of structure, meaning that authors rarely obeyed rules concerning paragraphing or spacing. Another particular feature of short textual messages was that they had less useful information in terms of content.

These features are extracted by counting the total number of sentences, total number of paragraphs, the average number of sentences per paragraph, the number of words per paragraph, the average number of characters per paragraph, the average number of words per paragraph and the total number of blank lines in the entire text. Table 6 shows the features that were categorized in the structurally-based features category.

Table 6: Structural-Based Features

Feature	Description
Total number of sentences	
Total number of paragraphs	In the case of pressing enter
Average number of sentences per paragraph	
Average number of words per paragraph	
Average number of characters per paragraph	
Average number of words per sentence	
Total number of blank lines	In the case of pressing enter

3.5 Function Word-Based Features

Function words are words that do not have important lexical meaning but which the writer uses to generate grammatical relationships with other words in the sentence. Another type of function word is words authors use to express their feelings or their mood. For this part of the analysis, we used mostly function words, as there are thousands of words that are not generally used.

Function-based features are divided into six different categories. Article words precede nouns to indicate whether we are referring to a specific or general thing. Pro-sentence words are single words that can take the place of a full sentence. Auxiliary verbs add functional or grammatical meaning to the related clause and are another function-based feature category. Conjunctions are words that connect phrases, clauses and sentences. Finally, interjections express emotions. To prevent the presence of too many zeros in the results, we did not investigate all of these word types. Table 7 shows the words that were extracted in this part of the analysis.

Table 7: Function-Based Features

Feature	Description
Total number of article words	The, A, An
Total number of pro-sentence words	Yes, No, Okay, Amen
Total number of pronoun words	a , an, all, another, any, anybody, anyone, anything, both, each, either, everybody, everyone, everything, few, he, her, hers, herself, him, himself, his, I, it, its, itself, many, me, mine, more, most, much, my,

	myself, neither, no one, nobody, nothing, one, other, others, our, ours, ourselves, several, she, some, somebody, someone, something, that, their, theirs, them, themselves, these, they, this, those, us, we, what, whatever, which, whichever, who, whoever, whose, you, your, yours, yourself, yourselves, yes
Total number of auxiliary verbs	Be, am, is , are, was, were, being, can, could, dare, do, does, did, have, has, had, having, may, might, must, need, ought, shall, should, will, would, can’t, don’t, won’t, aren’t, isn’t, wasn’t, weren’t, couldn’t, doesn’t, didn’t, haven’t, hasn’t, hadn’t, shouldn’t, wouldn’t
Total number of conjunction words	Him, himself, his, I, it, its, itself, many, me, mine, more, most, my, myself, neither, no one, nobody, nothing, one, other, others, our, ours, ourselves, several, she, some, somebody, someone, something, that, their, theirs, them, themselves
Total number of interjection words	Aah, aha, ahem, ahh, argh, aww, aw,

	bah, boo, booh, brr, duh, eek, eep, eh, eww, gah, gee, grr, hmm, humph, harumph, huh, hurrah, ich, yuck, yak, meh, eh, mhm uh-hu, mm, mmh, muahaha, mwahaha, nah, nuh-uh, oh, ooh-la-la, oh-lala, ooh, oomph, umph, oops, ow, oy, pew, pff, phew, psst, sheesh, jeez, shh, shoo, tsk-tsk, uh-uh, oh-oh, uh-uh, uhh, err, wee, whee, whoa, wow, yahoo, yay, yeah, yee-haw, yoo-yoo, yah-uh, yuck, mwah, neener-neener, zowie, zoinks, yow, yikes, va-va-voom, ugh, tchah, rah, sis-boom-bah, shh, ole, lah-de-dah, hup, ich, hubba-hubba, ho-hum,

Chapter4

AUTOMATIC FEATURE EXTRACTION

The implementation of extracting the features, which was described in the previous section, was done using Java codes. First, emails were trimmed in such a way that the remaining parts only consisted of the body of the message without the additional information that is automatically added to the top of every email message, such as date, time, sender, receiver, subject, Cc, Bcc, etc. Using the Java class "java.io.BufferedReader.readline ()" in a loop, headers of e-mails were trimmed. The remaining part was the body of the message, which was saved in the dataset folder using the " Java.io.PrintWriter " class.

As the aim of this thesis was to analyse texts exchanged over the Internet, the dataset was filtered to obtain messages between 10 and 500 words in length. To apply this filter, all the words in the emails include in the dataset were counted. Emails shorter than 10 words in length or longer than 500 words were automatically removed from the dataset. This modified dataset was then used to extract features in forthcoming experiments.

To implement character-based features extraction, at the time text was searched in a character-by-character manner, the counter of relevant features increased by applying appropriate criteria, such as whether the character was a digit, a specific punctuation or a white space.

To be able to count a specific word in the text, the “java.util.StringTokenizer” class was used. This part of the code tokenized the message based on the number of white spaces; thus, if there was any punctuation after a word, it was counted as one token. These tokenized words can be useful for extracting punctuations in the syntactic-based features category, such as periods, question marks or even multiple exclamation marks by using the “endsWith” property of “StringTokenizer”.

In informal writing, people might apply punctuation wrongly, for example, the author might put a space between words where it is not needed or use multiple question marks. For the present study, even in such a situation punctuation was extracted, because multiple question mark use will be tokenized in a single token and the counter of this punctuation will be increased. To avoid an inaccurate result, text was analysed word by word in order to count these wrongly-applied punctuations. In this phase of the analysis, all syntactic features were retrieved.

When using punctuation after a word, there is no space between the word and the punctuation in question. This causes some obstacles when retrieving the words we were searching for, for example, retrieving the word “However” from the following sentence: “However, the situation is good”. Here, “However” was tokenized with a semicolon directly following the word. In order to retrieve the words we were searching for, after reading the message, punctuations are removed using regular expression. For example semicolon is removed in “However,” token in this example.

```
justWords = st.replaceAll("[^\\p{L}]+", " ")                (4.1)
```

After removing punctuations, words were tokenized again to establish word-based features and to save them in a list, enabling each word to be analysed separately. At this level, we counted the exact total number of words that were used in a text. Figure 4 shows the flowchart of automatic feature extraction.

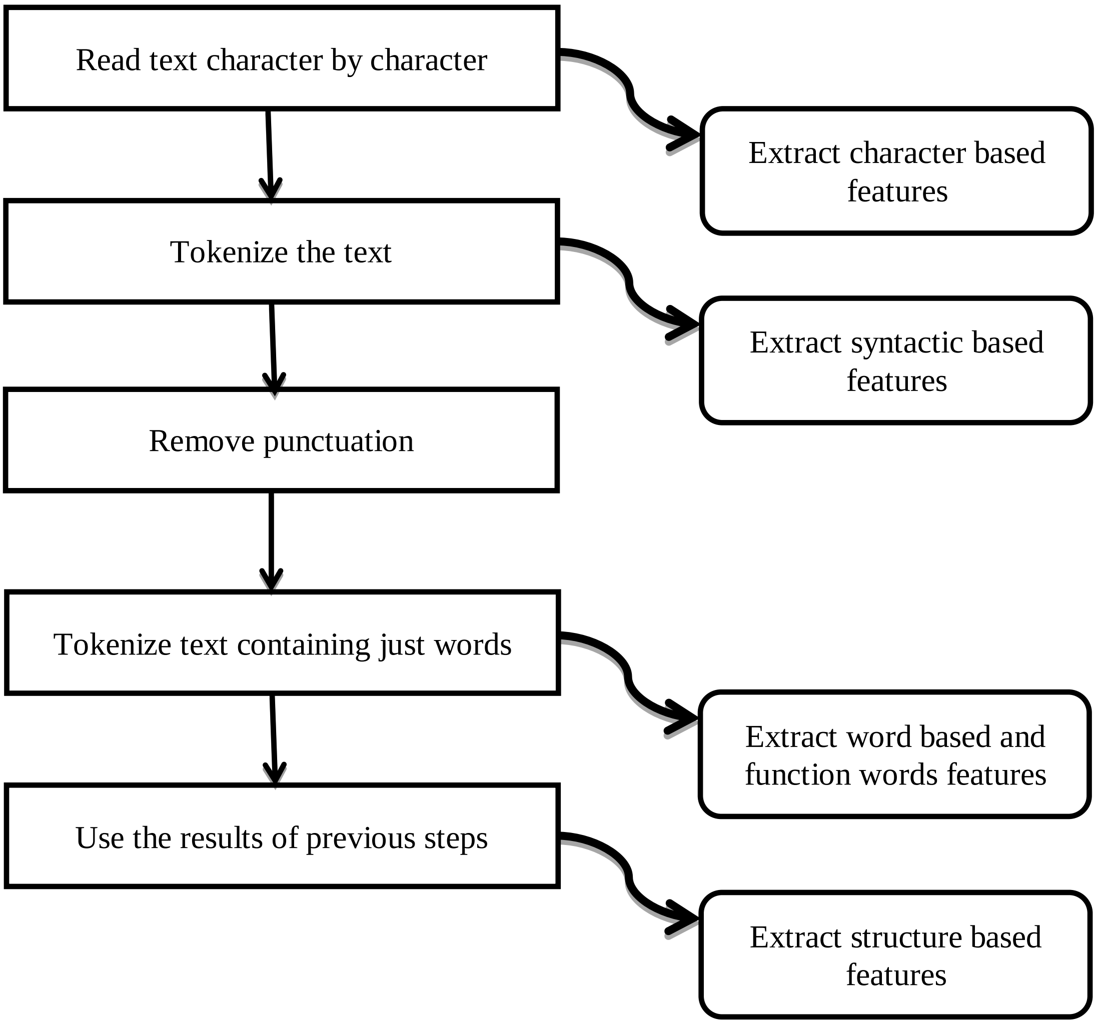

Read text character by character
Extract character based features
Tokenize the text
Extract syntactic based features
Remove punctuation
Tokenize text containing just words
Extract word based and function words features
Use the results of previous steps
Extract structure based features

Figure 4: Automatic feature extraction flowchart

The average word length was calculated using the "length" property of tokenized words. The vocabulary richness of a text consisted of the total number of different words that a writer used. The "java.util.TreeMap" Java class was used to produce a tree in which each node represented a unique word in the text, as well as the number of iterations of said word. By applying this piece of code, the total count for different words that had been used by an author in the message was retrieved. Figure 5 illustrates the implementation of vocabulary richness.

```
public static intrichnesOfText(TreeMap<String, Integer>frequencyData)
{
intrichnes = 0;
richnes = frequencyData.size();
    return richnes;
}
```

Figure 5: Implementation of vocabulary richness

The hapaxlegomena words were counted using the generated TreeMap. For implementation, we used the TreeMap's() properties to retrieve the number of iterations for each word in a text. Figure 6 illustrates the function used to calculate hapaxlegomena.

```
public static inthapaxLegomena(TreeMap<String, Integer>frequencyData, inthapax) {
int occurrences = 0;
    for (String word : frequencyData.keySet()) {
      if (frequencyData.get(word) == hapax) {
        occurrences++;
      }
    }
    return occurrences;
  }
```

Figure 6: Implementation of HapaxLegomena

Hapaxdislegomena was implemented in the same manner as the hapaxlegomena. Figure 7 illustrates the implementation of Yule's K measure.

```
public static double youleKmeasure(TreeMap<String, Integer>frequencyData, intwordCounter) {
intoccurences = 0;
        double yule=0;
        double sigmaV=0;
        double[] v=new double[richnes(frequencyData)];
        for (int j=0;j<v.length;j++){
        for (String word : frequencyData.keySet()) {
            if (frequencyData.get(word) == j+1) {
                occurrences++;            }
            v[j]=occurrences;        }
           occurrences=0;        }
        for(int g=0; g<v.length;g++)                    {
sigmaV+= ((double)v[g])*(Math.pow(((g+1)/(double) wordCounter),2));        }
yule=(Math.pow(10, 4))*((-(1/(double)wordCounter))+sigmaV);
               return yule;    }
```

Figure 7: Implementation of Yule's K measure

Simpson’s D was implemented by making use of the total number of words and their iterations. Figure 8 illustrates the function that retrieves this measure.

```
public static double simpsonDmeasure(TreeMap<String, Integer>frequencyData, intwordCounter) {
intoccurences = 0;
        double sympsonD=0;
        double sigmaV=0;
        double[] v=new double[richnesss(frequencyData)];
        for (int j=0;j<v.length;j++){
        for (String word : frequencyData.keySet()) {
            if (frequencyData.get(word) == j+1) {
occurences++;
            }
            v[j]=occurences;
        }
occurences=0;
        }
                for(int g=0; g<v.length;g++)
                {
sigmaV+= ((double)v[g])*(Math.pow(((g+1)/(double) wordCounter),2));
            sympsonD+=((double)v[g])*((g+1)/((double)wordCounter))*(g/((double) (wordCounter-1)));
        }
        return sympsonD;
    }
```

Figure 8: Implementation of Simpson's D measure

Figure 9 illustrates the function for calculating the entropy of the words.

```
public static double entropyMeasure(TreeMap<String, Integer>frequencyData, intwordCounter) {
intoccurences = 0;
        double sympsonD=0;
        double entropy=0;
        double[] v=new double[richnesss(frequencyData)];
        for (int j=0;j<v.length;j++){
        for (String word : frequencyData.keySet()) {
            if (frequencyData.get(word) == j+1) {
occurences++;
            }
            v[j]=occurences;
        }
occurences=0;
        }
          for(int g=0; g<v.length;g++)
                    {
          entropy+= ((double)v[g])*(Math.log10(((g+1)/wordCounter)))*((g+1)/wordCounter);
        }
           return entropy;
    }
```

Figure 9: Implementation of Entropy

Together, the above-mentioned features produced a 374-dimension vector for representing the values of the features of each message. Since the data set was a selection of messages up to 500 words, the result for each feature varied from zero to thousands. For example, the first three extracted features of an email was 2.42.250, respectively, and represented the number of sentences, total number of words and total number of characters that had been used in a specific message created by a female author. A sample of the extracted features is shown in Figure 10 and Figure 11 according to gender.

Sample No.	Sentence counter	White space counter	Character counter	Small letter counter	Big letter counter	Letter counter
1	1	4	21	14	1	15
2	1	6	36	28	1	29
3	4	107	650	360	45	405
4	6	282	2281	1339	258	1597
5	39	1068	6840	4593	438	5031
6	5	258	2178	1234	269	1503
7	9	146	781	600	12	612
8	2	24	143	109	4	113
9	5	84	461	348	15	363
10	1	32	186	145	5	150

Figure 10: Extracted male features

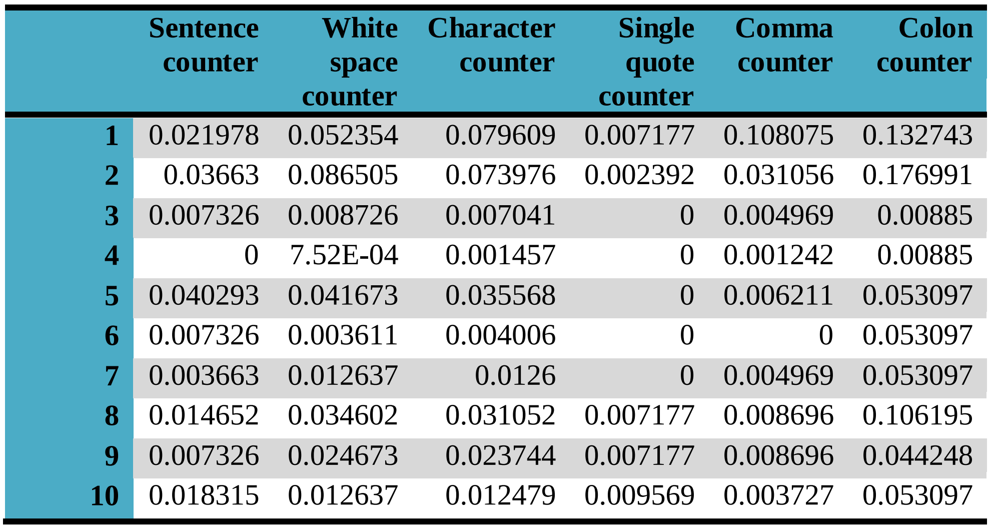
Sample No.	Sentence counter	White space counter	Character counter	Small letter counter	Big letter counter	Letter counter
1	15	223	1766	924	153	1077
2	4	54	437	262	43	305
3	11	362	6255	2363	2076	4439
4	3	50	442	261	48	309
5	4	61	501	335	34	369
6	15	330	2068	1387	122	1509
7	6	113	812	482	81	563
8	2	84	630	388	61	449
9	3	53	456	304	30	334
10	3	98	696	456	57	513

Figure 11: Extracted female features

Since the number of words in each text was different, message features were normalized using Equation 4.2, alongside a max-min normalization method to ensure that all features were treated equally. This resulted in the [0-1] range of feature values for gaining a fair result.

$$\text{Normalized } X_{ij} = \frac{Xij - \min(Xj)}{\max(Xj) - \min(Xj)} \qquad \textbf{(4.2)}$$

where $X_{ij}$is the ith feature of jth text, and min (X_j) and max (X_j) are the minimum and maximum of respected feature. Figure 12 shows a part of messages in the normalized feature value.



	Sentence counter	White space counter	Character counter	Single quote counter	Comma counter	Colon counter
1	0.021978	0.052354	0.079609	0.007177	0.108075	0.132743
2	0.03663	0.086505	0.073976	0.002392	0.031056	0.176991
3	0.007326	0.008726	0.007041	0	0.004969	0.00885
4	0	7.52E-04	0.001457	0	0.001242	0.00885
5	0.040293	0.041673	0.035568	0	0.006211	0.053097
6	0.007326	0.003611	0.004006	0	0	0.053097
7	0.003663	0.012637	0.0126	0	0.004969	0.053097
8	0.014652	0.034602	0.031052	0.007177	0.008696	0.106195
9	0.007326	0.024673	0.023744	0.007177	0.008696	0.044248
10	0.018315	0.012637	0.012479	0.009569	0.003727	0.053097

Figure 12: Normalized extracted features

All the features that were supposed to be evaluated for author gender identification were collected in three different CSV (comma-separated values) files. The first of these files contained features extracted from female authors, the second features extracted from male authors, while the third file was the normalized set of both female and male extracted features collections.

Chapter 5

MACHINE LEARNING

Machine learning involves writing a computer program that sees a computer attempting to mimic the intelligent abilities of a human. The computer attempts this by having the program use training data that has been collected specifically for this aim or by referencing the program's previous software executions. Many successful applications exist that can predict the behaviour of customers or optimizing the performance of a robot by analysing previously collected datasets[25].

In some situations, programmers may be unable to write a program directly for a particular system; in these cases, the system needs to learn from a range of different situations to be able to recognize a particular problem. An important task of machine learning is to implement an algorithm that can differentiate between specific input data and how this data relates to classes based on the sample’s different features[26].

In terms of speech recognition, a programmer needs to convert signals to ASCII codes; the problem in this context is that we are unable to explain how a human recognizes different accents, or cases where people use different words to describe the same thing as a result of their specific culture, age, gender, etc. The approach taken by machine learning is to collect a vast amount of training data concerning various accents from people of different ages and other criteria, and to try and map these data to a specific word[27].

Another problem arises when data is recorded at different times and in different places under noisy conditions; here, we can still expect the need for solving the same problem, rather than writing a single program for each problem. However, in some cases like packet routing within a network, it is impossible to write an explicit program for each problem. By making use of machine learning techniques, we can train the system by assigning to it a training corpus that can assist the system in making decisions regarding destination changes or network traffic[28].

In this thesis, machine learning algorithms were used to design a system that would be able toidentify whether the author of a text was female or male. This was done by assigning a training corpus to the system, enabling the system to learn the identification criteria for each category according to the defined feature sets. Figure 13 shows two data sets containing texts that were written by female and male authors.

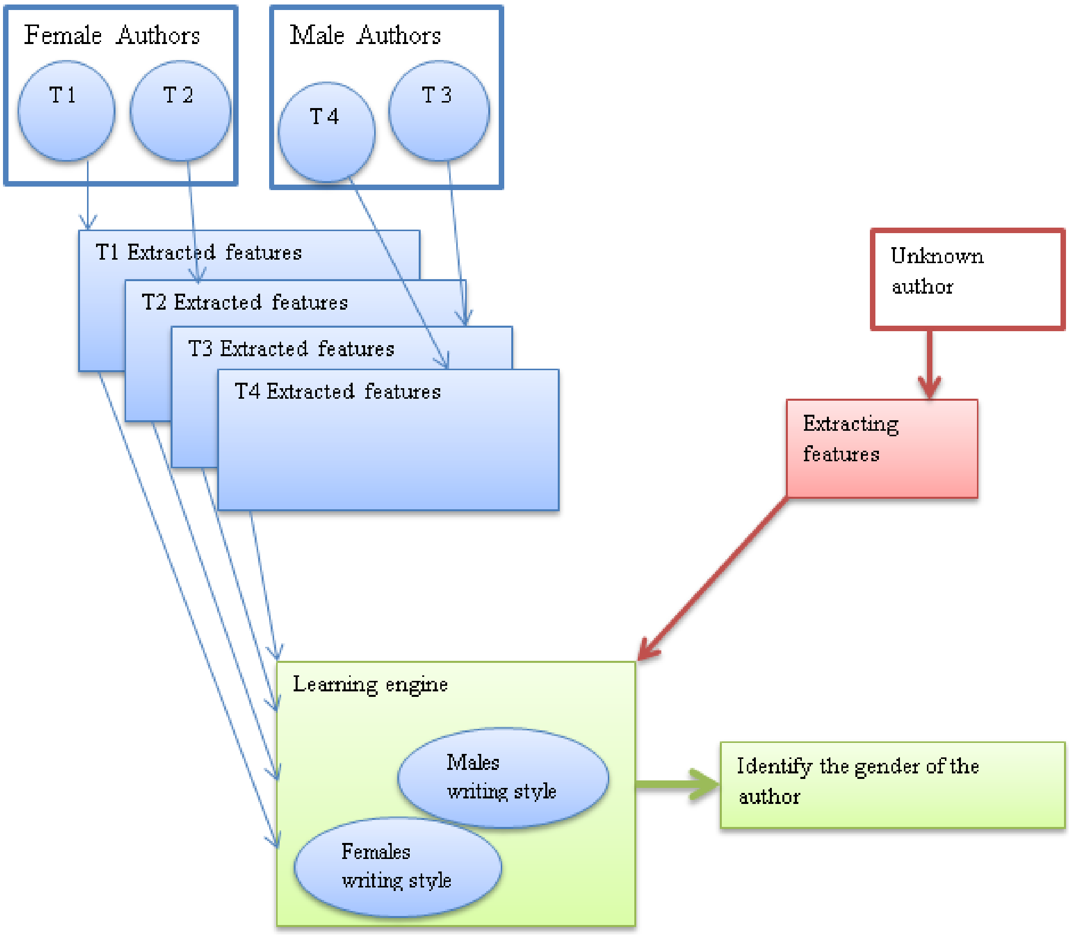


Figure 13: Process of identifying the gender of an author

The female dataset contained various text samples T1, T2 ..., of female authors same as male dataset that contains texts written by male authors. The results of extracting all the texts' features were submitted to the learning engine in order to train the system. After training the system, the learning engine extracted female writing and male writing styles. In this stage of the analysis, when submitting a text written by a gender-unknown author, the system extracted the particular features of the text. Then, comparing the extracted features with the existing writing styles, the system was able to predict the gender of the author.

Each learning machine application had two main parts. The first part was the learning association, which described association rules by defining the conditional probabilities of the defined problem. The second part was classification; in this part, each problem was categorized into a related class that had already been defined. For example, if the problem concerned recognizing the writer of a submitted text as a teenager or a middle-aged or old individual, three different classes would be defined: the class teenager, class middle-aged and class old. The system was then responsible for assigning the input text to one of these three classes according to the learning-association rules[29].

There are several algorithms that can be used for classification techniques. When choosing a classifier for training the system, the first aspect that should be considered is how big the training dataset is. Is there no training data? Is the dataset very small? Is there a large training dataset? An enormous one? The first challenge in the field of machine learning is therefore preparing a suitable training dataset. In most applications in the real world, a large training corpus is needed to produce a high performance system [29].

If there is no labelled training set, the solution is to use expert staff in the specific field to write the rules. This means that some queries need to be written such as:

if (A and B) or (C and D) then result = Y

If there is a small training dataset, it is better to apply classifiers with a high bias, for example, the naive Bayes classifier outperforms other classifiers in these situations[30].

If there is a big enough training corpus, almost all of the algorithms can be used. The other criteria for deciding what classifier to choose is considering the advantages of each algorithm. In the following section, the advantages and disadvantages of the naive Bayes classifier, logistic regression, the decision tree, support vector machine and Bayesian-based logistic regression algorithms are briefly described.

5.1 Naive Bayes

This algorithm has been widely used because of its simplicity. For naive Bayes conditional independent assumptions, the algorithm gathers the needed information quicker than other discriminative algorithms such as logistic regression, which leads to the use of less training data. Naïve Bayes outperforms in real applications and as such, this algorithm is the best choice in cases where fast, easy and reliable classifier is needed. The primary disadvantage of this classifier is that it is not able to understand interactions between criteria. An example of this in practice is if, for example, a customer likes bread she might also like meat but she might hate eating meat and bread together. Naïve Bayes is unable to understand the concept that one might like eating meat and bread separately, but hate eating them together [31].

5.2 Logistic regression

The most helpful feature of this algorithm is that there are many model regularization methods available and therefore, unlike the naive Bayes,there is no limit to correlations among features. It is also possible to add new training data at a later stage, which is impossible when using the decision tree and support vector machines. Logistic regression has been advised to be used when needing a probabilistic framework in order to be able to adjust thresholds in the case of uncertainty, or when researchers expect more datasets to be added at a later stage, thus enabling them to incorporate additional training data into their models[32].

The disadvantage of logistic regression is that it can only be used to predict discrete functions. Therefore, the dependent variable of logistic regression is restricted to the discrete number set. This restriction is problematic, as it is prohibitive to the prediction of continuous data.

5.3 Decision Trees

The reason that decision trees have become popular is that they are fast in giving the results, have the ability to be expanded and there is no need to set a large number of parameters. This classifier is also easy to understand and describe to others. Since it is not parametric, this feature makes decision trees' features easy to handle, meaning there is no need to worry about whether classes are linearly devisable. For example if the class 'female' is in the bottom and top range of the results chart and there is also a male class in the midrange, these classifiers will be able to successfully work with these classes. The primary disadvantage of these classifiers is that they do not support online learning; this means that in the case of a new instant, the tree has to be rebuilt from scratch [33].

5.4 Support Vector Machine

In the over fitting cases, the support vector machine (SVM) classifier performs with high accuracy and very strong theoretical guarantees, even when the classes involved are not linearly distinguishable. This algorithm is highly recommended for use in text classification, as its input vectors are highly dimensional. The disadvantage of this classifier is that it is memory intensive and too complicated to explain to others with limited knowledge thereof.

The idea of support vector machines was introduced by ValdemirVapnik in 1979, but the first officially submitted paper in this field appears to date back to 1995, written by Vapnik[34]. The main reason for using this relatively new machine learning algorithm in this thesis is to find a hyper-plane in high dimensional data that would be able to devise input data into two classes, that is, male and female. The input data, however, was not always linearly devisable, which is why kernel has been defined through the support vector machine classifier and places the data in a higher dimensional space where this classifier can easily categorize the data into the two stated divisible classes.

Generally, casting data in a higher dimensional space causes some computational difficulties; additionally, some over-fitting will also occur. The support vector machine classifier deals with this problem by not directly engaging with the higher dimensional data. Moreover, there is a measure for evaluating the likeliness of unseen data in the system (VC-dimension) and which can be easily calculated, unlike some other machine learning algorithms that do not have such a measure.

Overall, it has been stated by a number of researchers, as well as in practice, that the support vector machine classifier is successful in classifying the input data into related classes and can even be used for solving regression problems. Modern support vector machines differ from earlier algorithms in three ways, that is, in terms of optimal hyper-plane, kernel and soft margins[35], which will be discussed in upcoming paragraphs.

The training set is considered as linearly separable when there is a linear discriminant function that can easily match categories of the entire training corpus. In linearly separable problems, there are usually infinite numbers of support vectors that divide classes. Vapnik and Lerner (1963) chose the hyper-plane that left the largest space between the hyper-plane and the nearest instant. This is illustrated in Figure 14.

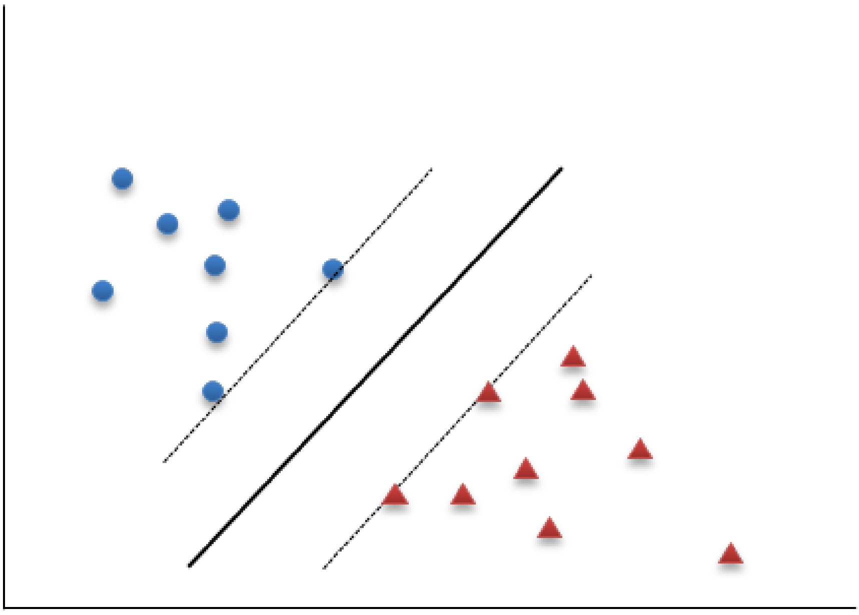


Figure 14: The best hyper-plane is one that separates circles from triangles by taking to account the nearest instances.

In the earliest linear classifiers, a pattern, *x*, is given to a class, *y*=±1, which transforms the pattern into the feature vector $\varphi(x)$, where $\hat{y}$ (x) = $w^T\varphi(x) + \mathrm{b}$, the parameters *w* (that is, the normal vector to the hyper-plane known as the weight vector) and b (that is, bias) are determined by running on a training dataset (x_1, y_1), ..., (x_n,y_n) and $\varphi(x)$ is always chosen by the person who solves the problem. Choosing the optimum hyper-plane is expressed by the optimization shown in Equation 5.1.

$$\min\rho(w,b) = \frac{1}{2}w^2 \qquad \textbf{(5.1)}$$

where

$$\forall i \quad yi(w^T\varphi(x) + \mathrm{b}) \geq 1$$

Equation 5.1 is hard to solve as the constraints are too complex. Using lagrangian duality this problem simplifies and leads to solve following dual problem in equation 5.2:

$$\max \mathrm{D}(\alpha) = \sum_{i=1}^{n} \alpha_{\mathrm{i}} - \frac{1}{2}\sum_{i,j=1}^{n} y_{\mathrm{i}}\, \alpha_{\mathrm{i}} \mathrm{y}_{\mathrm{j}} \alpha_{\mathrm{j}} \varphi(\mathrm{x}_{\mathrm{i}})^{\mathrm{T}} \varphi(\mathrm{x}_{\mathrm{j}}) \qquad \textbf{(5.2)}$$

where

$$\begin{cases} \forall i\ \alpha i \geq 0, \quad (lagrange\, multiplier) \\ \qquad \sum_i yi\alpha i = 0. \end{cases}$$

The direction of the hiperplane (w^*) can be specified from the solution α^* of the above formula.

$$w^* = \sum_i \alpha_i^* y_i\, \varphi(x_i) \qquad \textbf{(5.3)}$$

This, results in a simple equation to find b^* and the linear discriminant function can be reconstruct as equation 5.4:

$$\hat{y} = w^{*T} + b^* = \sum_{i=1}^{n} y_i \alpha_i^*\, \varphi(x_i)^T \varphi(x_j) + b \qquad \textbf{(5.4)}$$

The perceptron algorithm is described in Figure 15[36].

```
Require: A linearly separable set S, learning rate $\eta \in R^+$
$w_0 = 0; b_0 = 0; k = 0;$
$R = \max \| x_i \|$
      $1 \leq i \leq L$
while at least one mistake is made in the for loop do
    for $i = 1; : : : ; L$ do
        if $y_i(<w_k; x_i> +b_k) \leq 0$ then
            $w_{k+1} = w_k + \eta y_i x_i$
            $b_{k+1} = b_k + \eta y_i R^2$(updating bias)
            $k = k + 1$
        end if
    end for
end while
Return $w_k;b_k$, where $k$ is the number of mistakes
```

Figure 15: Perceptron Algorithm

This algorithm takes an instance and predicts its class. If the prediction is correct, there is no need to make any adjustments. If the prediction is wrong, the parameters that describe the hyper-plane are moved in the direction of the point in which the mistake occurred.A scalar value, η, referred to as the learning rate, determines how far the parameters will be moved. The choice of learning rate can significantly affect the number of iterations until convergence occurs on a linearly-separable set.

Equation 5.2 and equation 5.3 can only involve in dot products in the sample space, so one who deals with this sort of problems, there is no need to compute φ(x), instead of computing (x), it is possible to compute the dot product. For nonlinear separable spaces, Boser, Guyon, and Vapnik (1992) suggested to choose a kernel function K(x,x́) that can be able to represent φ(x) in a higher dimensional feature space. For too noisy problems it seems impossible to find a strict devisor for the classes, that's why Cortes and Vapnik (1995) proposed soft margins[37] to let some of instants overstep the devisor using positive slack variables (which measure the degree of misclassification of the data x_i) ε=(εI , ... , εn). On the other hand there is a need to take control of the greatness of the violation by using another parameter c. It transforms the equation 5.1 to equation 5.5.

$$\text{Min } \rho(\mathrm{w},\mathrm{b},\varepsilon) = \frac{1}{2}\mathrm{w}^2 + \mathrm{c}\sum_{i=1}^{n} \varepsilon \mathrm{I} \qquad \textbf{(5.5)}$$

where

$$\begin{cases} \forall i \quad yi(w^T\varphi(x) + \mathrm{b}) \geq 1 - \varepsilon \\ \quad \forall i \qquad \varepsilon \geq 0 \end{cases}$$

and equation 5.2 to equation 5.6 in the case of duality,

$$\text{Max D}(\alpha) = \sum_{i=1}^{n} \alpha_{\mathrm{i}} - \frac{1}{2}\sum_{i,j=1}^{n} y_{\mathrm{i}}\, \alpha_{\mathrm{i}}\mathrm{y}_{\mathrm{j}}\alpha_{\mathrm{j}}\mathrm{k}_{\mathrm{i,j}} \qquad \textbf{(5.6)}$$

where

$$\begin{cases} \forall i\ c \geq \alpha i \geq 0, \\ \sum_{i} yi\alpha i = 0, \\ K\ is\ the\ kernel\ values\ matrix. \end{cases}$$

and equation 5.4 transforms to equation 5.7 as follow:

$$\hat{y} = \mathrm{w}^{*\mathrm{T}} + \mathrm{b}^{*} = \sum_{i=1}^{n} y_{\mathrm{i}}\alpha_{\mathrm{i}}^{*}\, k(\mathrm{x}_{\mathrm{i}}, \mathrm{x}) + \mathrm{b}^{*}. \textbf{(5.7)}$$

The above discussed subject is just scratching of what are the basics of Support Vector Machines and further studying in this field is not the subject of this thesis.

5.5 Bayesian-Based Logistic Regression

Bayesian classifiers are widely used when examining data in almost every field, including machine learning problems. After more than a century, this type of classifier remains interesting to many researchers[36]. The logistic regression model shows that if vector instance x_i affiliates to category y_i:

$$P(y_i = +1 \mid \omega , x_i) = \psi (\omega^T x_i) \quad \textbf{(5.8)}$$

whereΨ ( ) is logistic link function

$$\Psi (r) = \frac{1}{1+\exp(-r)} . \quad \textbf{(5.9)}$$

To successfully assign an instant to corresponding class, choosing the appropriate threshold is playing an important role and it should be defined as below.

If

$P(y_i = +1 \mid \omega , x_i) >$ threshold

Then $y=+1$

Otherwise $y = -1$

To avoid over fitting in Bayesian problems a distribution of ω and an optimization algorithm can be applied[38]. Using Gaussian distribution to specify ω_j to solve the prior mentioned problems we have following equations:

$$P (\omega_j \mid \tau_j) = N (0, \tau_{j)} \quad \textbf{(5.10)}$$

where

$$j = 1, 2..., d.$$

where the density of $\tau_{j\,(}$variance) is calculated by exponential distribution:

$$P (\tau_j \mid \gamma) = \frac{\gamma}{2} \exp (-\frac{\gamma}{2} \tau_{j)} > 0 \quad \textbf{(5.11)}$$

This is same as none hierarchical double exponential distribution with density:

$$P(\omega_j \mid \lambda_j) = \frac{\lambda}{2} \exp(-\lambda_j \mid \omega_j \mid) \qquad \textbf{(5.12)}$$

where

$$\lambda_j = \sqrt{2} / \sqrt{\tau j}$$

Considering the components of ω are unconnected, we have equation 5.13 for prior density

$$P(\omega) = \Pi_{j=1}^{d} P(\omega_j \mid \lambda_{j)} = \Pi_{j=1}^{d}\frac{\lambda}{2} \exp(-\lambda_j \mid \omega_j \mid) \qquad \textbf{(5.13)}$$

Logistic link to corpus K will result in posterior density forω as below

$$L(\omega) = P(\omega \mid K_) \ \alpha \ P(K \mid \omega) \ P(\omega) \qquad \textbf{(5.14)}$$

and this is equal to the below equation by ignoring the normalization constant:

$$L(\omega) = -\sum_{i=1}^{n} ln\,(1 + \exp(-\omega^T x_i y_i)) - \sum_{i=1}^{n}(\ln 2 - \ln \lambda_j + \lambda_j \mid \omega_j \mid) \textbf{(5.15)}$$

Then ω can be estimated by finding the maximum posterior L (ω) or minimum −L (ω).

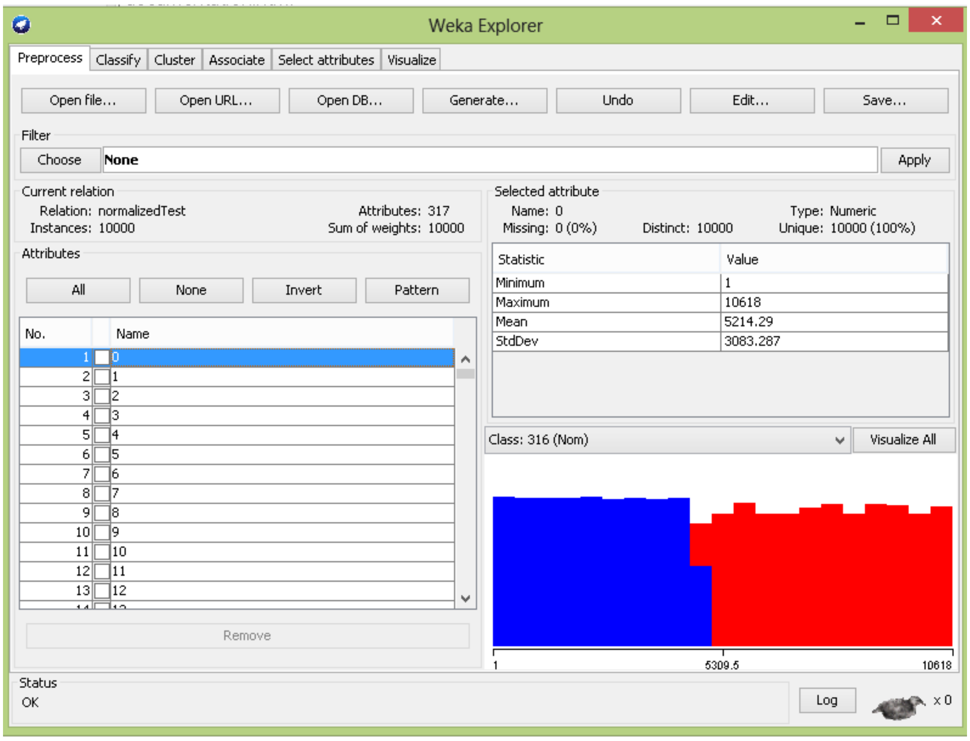
5.6 Weka

In this thesis, we used Weka version 3.7 to apply support vector machine and Bayesian-based logistic regression classifiers. The 'weka' is a flightless bird with an inquisitive nature found only on the islands of New Zealand. Weka[39] is a popular machine learning software suite written in Java, developed at the University of Waikato, New Zealand and is available under a GNU general public license.

The algorithms can either be applied directly to a dataset from the Weka GUI or Weka can be called from Java code. Weka contains tools for data pre-processing, classification, regression, clustering and association rules. It is also well-suited for developing new machine learning schemes and it is possible to apply Weka to big data.

The primary features of Weka include 49 data pre-processing tools, 76 classification/regression algorithms, eight clustering algorithms, three algorithms for finding association rules, 15 attribute/subset evaluators and 10 search algorithms for feature selection.

The main GUI includes three graphical user interfaces: “The Explorer”, which covers exploratory data analysis, “The Experimenter”, which encompasses the experimental environment and “The Knowledge Flow”, which includes a new process model-inspired interface[40]. Figure 16 shows the Weka Explorer environment that was used in the experiments for this thesis.



Filter: None

Current relation
Relation: normalizedTest
Instances: 10000
Attributes: 317
Sum of weights: 10000

Selected attribute
Name: 0
Missing: 0 (0%)
Distinct: 10000
Type: Numeric
Unique: 10000 (100%)

Statistic	Value
Minimum	1
Maximum	10618
Mean	5214.29
StdDev	3083.287

Attributes

No.	Name
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
13	12

Class: 316 (Nom)

Status
OK

Figure 16: Weka Explorer

Chapter 6

EXPERIMENTION RESULT ANALYSIS

We conducted various experiments to evaluate the performance of the system and to achieve optimum results. The support vector machine variant experiment was conducted to find the best kernel. We evaluated the performance of this classifier by applying three different kernels, linear, polynomial and RBF. For all the experiments, we set five-fold cross validation. Results showed that the best configuration for the support vector machine clarifier in this experiment was using the RBF kernel and setting Gamma=0.5 (parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’) and C=3 (parameter that trades off misclassification of training examples against the simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly). We applied 10 000 datasets to different support vector machine configurations in our experiments; as a result, we selected the RBF kernel and set C to 3 and Gamma to 0.5 in terms of supporting vector machine experiments.

Table8: Comparison the Performance of Support Vector Machine when Applying Different Kernels

Model	**Accuracy**
SVM-Linear	76.7%
SVM- Poly2	77.68%
SVM- Poly3	77.08%
SVM-RBF C=1,Gamma=0.01	76.21%
SVM-RBF C=3,Gamma=0.5	78%

The criteria we chose to examine the system were: first, we compared two different classifiers for the entire training dataset and then evaluated the results by specifying a limitation on the number of words in each text. We implemented the feature extractor in Java; we used output of this java code, which was a CSV file, as input into Weka.

6.1 Evaluating the Performance of SVM and Bayesian Logistic Regression by Applying a Complete Data Set as Input

In this part, we applied six different datasets to the support vector machine and Bayesian-based logistic regression. Each time, we randomly selected a training dataset containing a different number of emails, which were 2500, 5000, 7500, 10000, 12000 and 14000, respectively. We applied each dataset separately to the support vector machine and Bayesian logistic regression to evaluate the performance of each classifier when dealing with different training corpus sizes.

As can be seen in Figure 17, when submitting datasets containing 2500, 5000, 7500, 10000, 12000 and 14000 emails, their accuracies were 59%, 62%, 65%, 63%, 65% and 65%, respectively.

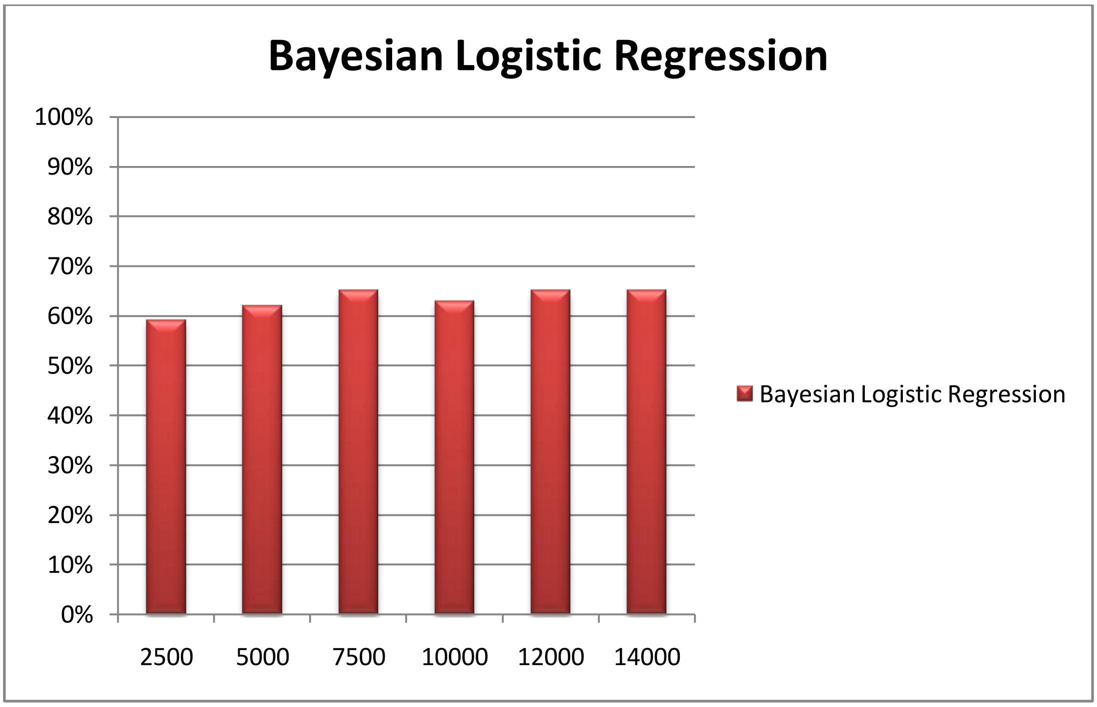


Figure 17: Comparison the Performance of Bayesian Logistic Regression Classifiers Based on the Size of Training Dataset

Figure 18 shows results were refined by submitting the larger dataset to the support vector machine. The accuracy for the 2500 emails was 63%, for 5000 emails 68%, for 7500 emails it was 75%, for a 10000 emails 78%, for 12000 emails 81% and for 14000 emails accuracy was 83%.

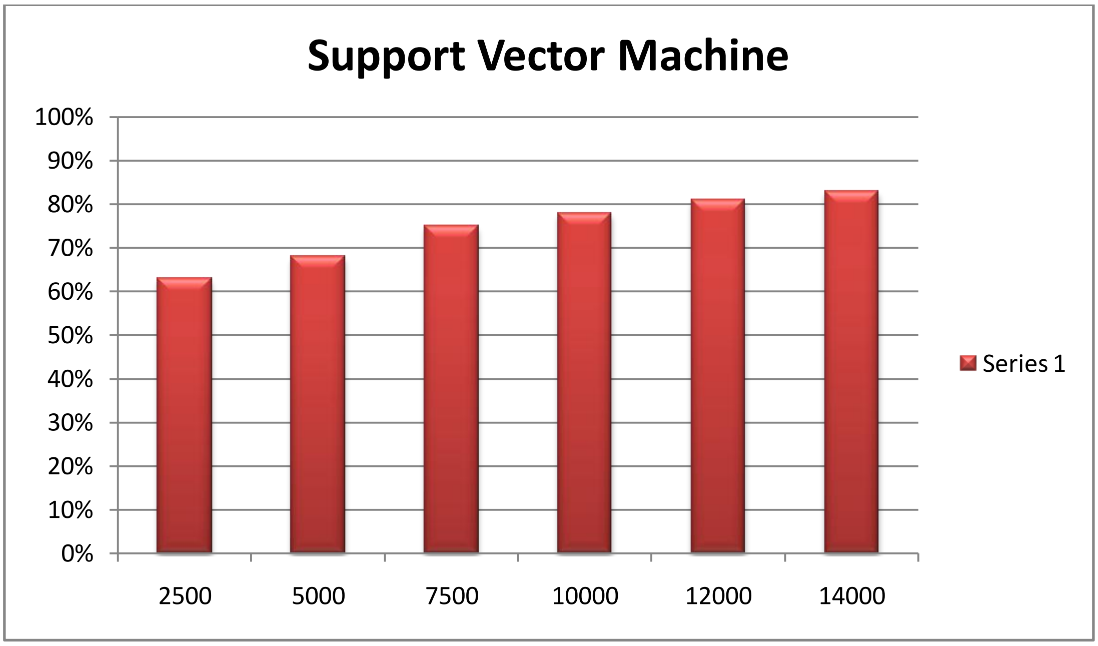


Figure 18: Comparison the Performance of Support Vector Machine Classifiers Based on the Size of Training Dataset

Comparing the performance of the two classifiers, the final result implied that the support vector machine classifier outperformed Bayesian logistic regression by 83%. Additionally, Figure 19 indicates that the performance of Bayesian logistic regression did not change significantly according to the size of the dataset, compared to support vector machines, which showed improvement when applying a bigger dataset.

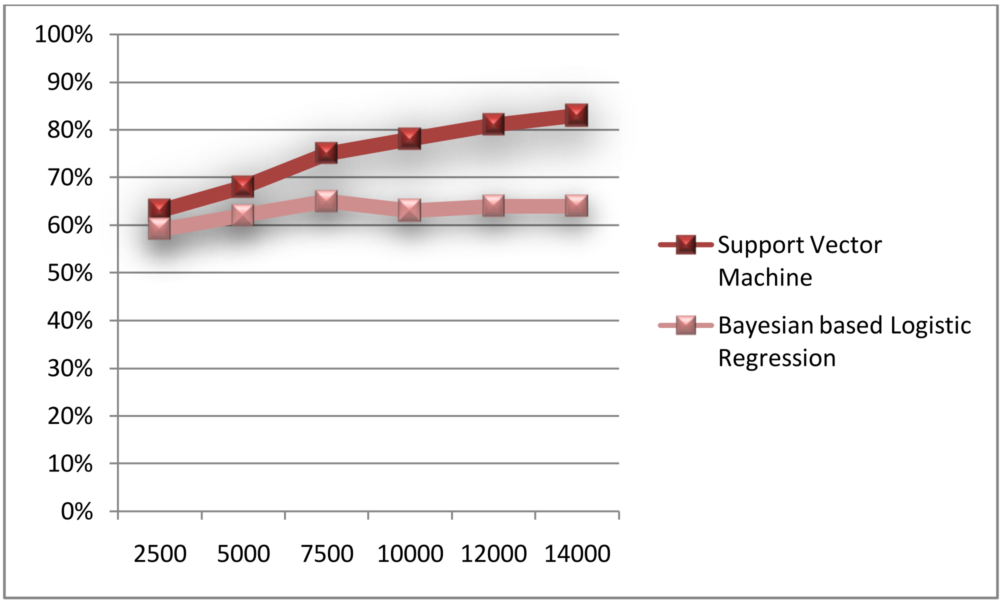


Figure 19: Comparison the Performance of Two Classifiers

In other research[41], another feature set selection was applied. The researchers used word class frequency as a feature set. Each word class consisted of words related to synonyms and hypernyms. The researchers set nine classes: money, job, sports, television, sleep, eat, sex, family and friends. Each of the lists contained an average of 1400 unique words. The final results of this research showed 57% accuracy, compared to the 83% accuracy of our research. It can thus be observed that feature set selection is a critical function of identifying the author's gender from a text.

In another study[42], results showed 82% accuracy when applying a dataset containing 8970 e-mails from the Enron dataset. The lengths of the texts were almost twice as long compared to the texts we chose as our dataset.

6.2 Comparing the Performance of Classifiers when Submitting a Limited Number of Words for Each Email

In another attempt to measure the accuracy for each classifier's results we provided three different training datasets containing 6354 emails, all containing less than 40 words; 2737 emails ranged between 41 and 70 words and 909 emails ranged between 71 and 100 words. Generally, as the results indicated, accuracy increased as the number of words per message increased, since more words in one message may contain more information about the author's personal writing style and the corresponding gender influence.

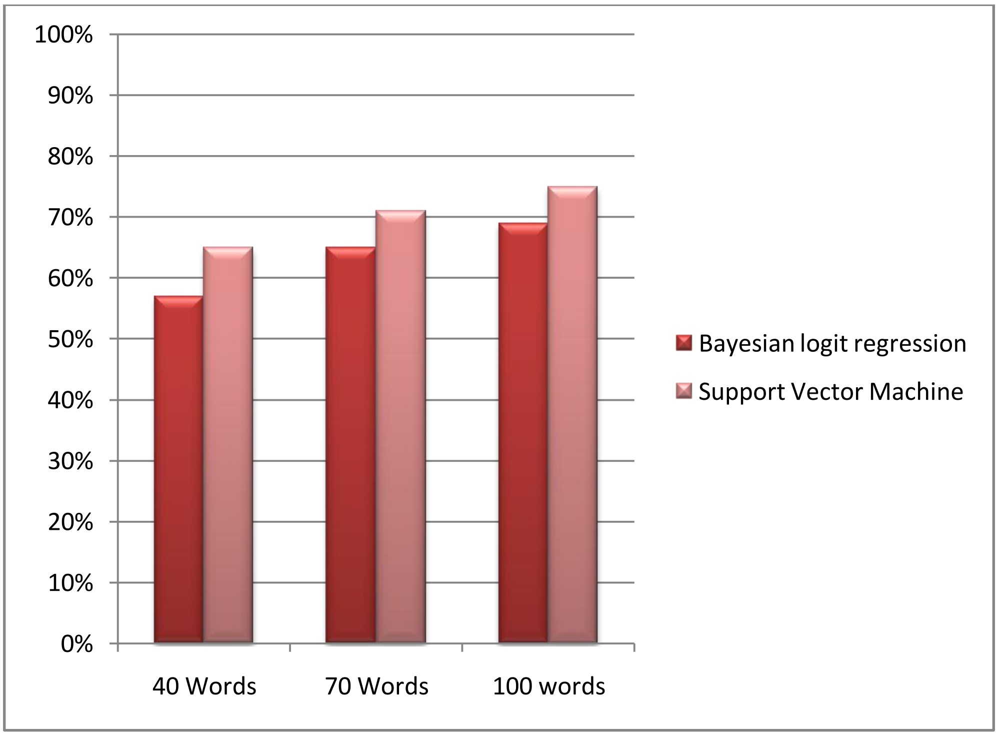


Figure 20: Comparison the Performance of Two Classifiers Submitting Different Length of Text

As can be seen in Figure 20, the performance of the Bayesian logistic regression classifier has been increased in almost all groups of emails, considering the number of submitted emails. Studying the chart by taking into account the support vector machine classifier, the results appear more accurate; however, in the previous section, we have seen that results are more accurate when the size of the training dataset is bigger. The SVM results can be improved by submitting the bigger training dataset with a limited number of words in each email. However, the aim of this thesis was to verify the gender of short textual messages and both classifiers produced moderate results in these cases, for example, in online conversations.

Chapter 7

CONCLUSION

In this thesis, we introduced a classifier-based implementation of author gender identification from text. Gender identification from text concerns the interplay between linguistic and writing styles, as well as those words that are commonly used by one gender. The results yielded by various experiments demonstrated the advantages of the different classifiers, as well as feature set selection.

The experimental results showed that designing an appropriate feature set by considering linguistics and features that correlate to gender is of high importance. It should be noted that some features such as certain interjection words were not common in any gender, while some words were mostly used by one gender. Furthermore, by removing words that were uncommon among both genders we can improve the feature set.

Choosing a particular classifier is of critical importance in this subject area. The results identified in the previous chapter showed that support vector machines outperform Bayesian logistic regression. Moreover, accuracy improvement was clearly observed in the support vector machine classifier by submitting the larger training dataset when compared to Bayesian logistic regression. The results indicate that, after a certain point, applying a bigger dataset does not improve the performance of Bayesian logistic regression, as was the case for support vector machines. After evaluating the advantages and disadvantages of each classifier, the support vector machine classifier appears to be the best candidate for author gender identification from text.

REFERENCES

[1] Valdemir Vapnik. (1979). Retrieved from SVM: ttp://www.svms.org/history.html

[2] Reuters corpora. (2000). Retrieved from http://trec.nist.gov/data/reuters/reuters.html

[3] Enron Email Dataset. (2009, August 21). Retrieved from Carnegie Mellon University: http://www.cs.cmu.edu/~./enron/

[4] PAN. (2009). Retrieved from http://www.webis.de/research/corpora

[5] A dictionary of first names. (2014). Retrieved from Oxford reference: http://www.oxfordreference.com/

[6] A dictionary on first names. (2014). Retrieved from Oxford reference: http://www.oxfordreference.com/

[7] Hapax Legomenon. (2014, March). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Hapax_legomenon

[8] Name. (2014). Retrieved from Behind the names: http://www.behindthename.com/

[9] A. Abbasi, H. Chen. (2012). A stylometric approach to identity-level identification and similarity detection in cyberspace. World Academy of Science, Engineering and Technology, 26(2), 1-29.

[9] Alpaydin, E. (2003). Introduction to Machine Learning. mitpress.

[10] Benno Stein · Nedim Lipka · Peter Prettenhofer. (2010). Intrinsic Plagiarism Analysis. Language Resources and Evaluation.

[11] Braja Gopal Patra,Somnath Banerjee,Dipankar Das,Tanik Saikh,Sivaji Bandyopadhyay. (2013). Automatic Author Profiling Based on Linguistic and Stylistic Features . *PAN*.

[12] Breiman, L. (2006). Random forests. Pattern Recognition in Remote Sensing (pp. 5-32). *ELSVIER*.

[13] Chen, H. (2005). Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems: special issueon AI for Homeland Security* , 67–75.

[14] Chen, H. (2005). Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems: special issue on AI for Homeland Security*, 67-75.

[15] Corney, M. d. (2002). Gender-preferential text mining of e-mail discourse. *Proceedings of the 18th Annual Computer Security Applications Conference*. Washington, DC, USA: IEEE Computer Society.

[16] DJ. Hand, K. Yu. (n.d.). Idiot's Bayes - not so stupid after all? *International Statistical Review*, 69(3), pp. 385-399.

[17] Dmitriy Fradkin,Ilya Muchnik. (2000). Support Vector Machines for Classification. *Mathematics Subject Classification*.

[18] E. STAMATATOS, N. FAKOTAKIS, G. KOKKINAKIS. (2001). Computer-Based Authorship Attribution Without. *Computers and the Humanities. Netherland: Kluwer Academic Publishers*.

[19] Edwill Nel, C.W. Omlin. (2004). Machine Learning Algorithms for Packet Routing in Telecommunication Networks. *SATNAC.*

[20] Emad E Abdallah, A.F. Otoom, ArwaSaqer, Ola Abu-Aisheh, Diana Omari, Ghadeer Salem. (2012). Detecting Email Forgery using Random Forests. *World Academy of Science, Engineering and Technology,* 6, pp. 03-23.

[21] F. Iqbal, H. Binsalleeh, B.C.M. Fung, M. Debbabi. (2010). Mining writeprints from anonymous emails for forensic investigation. *Digital Investigation,* (pp. 1-9).

[22] F. Mosteller, D. L. Wallace. (1984). Applied Bayesian and Classical Inference.

[23] F. Peng, D. Schuurmans, V. Keselj, S. Wang. (2003). Automated authorship attribution with character level language models. *The European Chapter of the Association for Computational Linguistics.*

[24] F.J.Tweedie, S.Singh, D.I.Holmes. (1996). Neural network applications in stylometry. The federalist papers,Computers and the Humanities, 30(1), 1-10.

Forman, G. I. (2004). Learning from little. 161-172.

[25] Genkin, Alexander, Lewis, D. David, Madigan, David. (2007). Large-scale bayesian logistic regression for text categorization. *Technometrics,* 49(3), 291-304.

[26] H. Baayen, H. van Halteren, A. Neijt, F. Tweedie. (2002). An experiment in authorship attribution. *6th International Conference on the Statistical Analysis of Textual Data*.

[27] H. Deng, G. Runger, E.Tuv. (2011). Bias of importance measures for multi-valued attributes and solutions. *Artificial Neural Networks* . ICNN.

[28] Holmes, D. (1992). A stylometric analysis of mormon scripture and related texts. *Royal Statistical Society*, 91-120.

[29] Investigation, D. (n.d.). Marlowe's hand in Edward III revisited. *Literary and Linguistic Computing*, 11(1), pp. 19-22.

[30] J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, R. J. Booth. (2007). *The Development and Psychometric Properties of LIWC2007*. Texas: LIWC Inc.

[31] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. *Springer*.

[32] Joachims, T. (1999). Making large-scale support vector machine learning practical. USA.

[33] Jordan, M. I. (2004). Soft Margin SVM. *Berkeley University of california*.

[34] K Santosh, Romil Bansal, Mihir Shekhar, Vasudeva Varma. (2013). Predicting Age and Gender from Blogs. *Notebook for PAN at CLEF 2013*.

[35] Kaizhu Huang,Zhangbing Zhou,Irwin King, Michael R. Lyu. (2003). Improving Naive Bayesian Classifier by Discriminative Training.

[36] L. Breiman. (2001). "Random forests" Machine Learning. 5-32.

[37] L.A.Gottschalk and G.C.Gleser. (1969). The measurement of psychological states through the content analysis of verbal behavior. *Berkeley*.

[38] L´eon Bottou,Chih-Jen Lin . (2013). Support Vector Machine Solvers. *csie*.

[39] Lakshmi,Pushpendra Kumar Pateriya. (2012). A Study on Author Identification through Stylometry . *International Journal of Computer Science & Communication Networks . IJCSCN*.

[40] Latent Dirichlet allocation. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

[41] M. L. Newman, J. W. Pennebaker, D. S. Berry, and J. M. Richards. (n.d.). Predicting deception from linguistic.

[42] Mulac, A. (1998). The gender-linked language effect. Do language differences really make a difference.

[43] Na Cheng, Xiaoling Chen, R. Chandramouli, K. P. Subbalakshmi. (2013). Gender Identification from E-mails.

[44] Ng, Andrew Y., Michael I. Jordan, Yair Weiss. (2001). On spectral clustering Analysis and an algorithm.

[45] O. D. Vel, M. Corney, A. Anderson, G. Mohay. (2002). Language and gender author cohort analysis of e-mail for computer forensics. *digital forensic research workshop*.

[46] P. Domingos M. Pazzani. (2001). On the optimality of the simple Bayesian classifier under zero-one loss. In G. Provan (Ed.), *Machine Learning*, (pp. 103-137). Irvine.

[47] P. Domingos, M. Pazzani. (2001). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 103-137.

[48] Ratsch, G. (2004). A Brief Introduction into Machine Learning. *Machine learning*.

[49] S.D.Rosenberg, G.J.Tucker. (1978). Verbal behavior and schizophrenia. *Verbal behavior and schizophrenia*, (pp. 1331-1337).

[50] Simon Tong,Daphne Koller. (2001). Support Vector Machine Active Learning with Applications to Text Classification. *Machine Learning Research*, 45-66.

[51] Tweedie, F., Baayen, H. (1998). How variable may a constant be? Measures of lexical richness in perspective. *Computers and the Humanities*, (pp. 323-352).

[52] Udhyakumar Nallasamy, Florian Metze,Tanja Schultz. (2012). ACTIVE LEARNING FOR ACCENT ADAPTATION IN AUTOMATIC SPEECH RECOGNITION. *interACT*.

[53] Waikato, U. o. (2006). A presentation demonstrating all graphical user interfaces in Weka. Retrieved from Waikato: http://www.cs.waikato.ac.nz/ml/weka/documentation.html

[54] Waikato, U. o. (2006). Weka. Retrieved from Waikato: http://www.cs.waikato.ac.nz/ml/weka/

APPENDIX

Appendix A: Extracted Features

Feature No.	Feature name
1	Total number of sentences
2	White spaces
3	Characters
4	Single quote signs
5	Comma signs
6	Colon signs
7	Semicolon signs
8	Question mark signs
9	Exclamation mark signs
10	Lowercase alphabets
11	Uppercase alphabets
12	Digits(0-9)
13	letters(all alphabets)
14	Quotation marks
15	Number signs
16	Dollar sighs
17	Percent signs
18	Ampersand signs
19	Parenthesis signs
20	Asterisk signs
21	Plus signs
22	Minus signs
23	Solidus signs
24	Less than signs
25	Equal signs
26	Greater than signs
27	At-sign signs
28	Reverse solidus sign
29	Square bracket signs
30	Circumflex accent signs
31	Lower line signs
32	Curly bracket signs
33	Vertical line signs
34	Tile signs
35	Total number of words
36	Average length of words
37	Ellipsis signs
38	Multiple question mark signs
39	Total number of tabs
40	Total number of Word longer than 6 characters

41	**Total number of words less than 3 characters**
42	**Total number of Period signs**
43	**Total number of ampesant signs**
44	**Total number of word " the"**
45	**Total number of word "a"**
46	**Total number of word "an"**
47	**Total number of word "all"**
48	**Total number of word "another"**
49	**Total number of word "any"**
50	**Total number of word "anybody"**
51	**Total number of word "anyone"**
52	**Total number of word "anything"**
53	**Total number of word "both"**
54	**Total number of word "each"**
55	**Total number of word "either"**
56	**Total number of word "everybody"**
57	**Total number of word "everyone"**
58	**Total number of word "everything"**
59	**Total number of word "few"**
60	**Total number of word "he"**
61	**Total number of word "who"**
62	**Total number of word "her"**
63	**Total number of word "hers"**
64	**Total number of word "herself"**
65	**Total number of word "him"**
66	**Total number of word "himself"**
67	**Total number of word "his"**
68	**Total number of word " I"**
69	**Total number of word "it"**
70	**Total number of word "its"**
71	**Total number of word "itself"**
72	**Total number of word "many"**
73	**Total number of word "me"**
74	**Total number of word "mine"**
75	**Total number of word "more"**
76	**Total number of word "most"**
77	**Total number of word "much"**
78	**Total number of word "my"**
79	**Total number of word "myself"**
80	**Total number of word "neither"**
81	**Total number of word "none"**
82	**Total number of word "nobody"**
83	**Total number of word "nothing"**
84	**Total number of word "one"**
85	**Total number of word "other"**

86	**Total number of word "others"**
87	**Total number of word "our"**
88	**Total number of word "ours"**
89	**Total number of word "ourselves"**
90	**Total number of word "several"**
91	**Total number of word "she"**
92	**Total number of word "some"**
93	**Total number of word "somebody"**
94	**Total number of word "someone"**
95	**Total number of word "something"**
96	**Total number of word "that"**
97	**Total number of word "their"**
98	**Total number of word "theirs"**
99	**Total number of word "them"**
100	**Total number of word "themselves"**
101	**Total number of word "these"**
102	**Total number of word "they"**
103	**Total number of word "this"**
104	**Total number of word "those"**
105	**Total number of word "us"**
106	**Total number of word "we"**
107	**Total number of word "what"**
108	**Total number of word "whatever"**
109	**Total number of word "which"**
110	**Total number of word "whichever"**
111	**Total number of word "who"**
112	**Total number of word "whoever"**
113	**Total number of word "whose"**
114	**Total number of word "you"**
115	**Total number of word "your"**
116	**Total number of word "yours"**
117	**Total number of word "yourself"**
118	**Total number of word "yourselves"**
119	**Total number of word "yes"**
120	**Total number of word "no"**
121	**Total number of word "okay"**
122	**Total number of word "amen"**
123	**Total number of word "be"**
124	**Total number of word "am"**
125	**Total number of word "is"**
126	**Total number of word "are"**
127	**Total number of word "was"**
128	**Total number of word "were"**
129	**Total number of word "being"**
130	**Total number of word "can"**

131	**Total number of word "could"**
132	**Total number of word "dare"**
133	**Total number of word "do"**
134	**Total number of word "does"**
135	**Total number of word "did"**
136	**Total number of word "have"**
137	**Total number of word "has"**
138	**Total number of word "had"**
139	**Total number of word "having"**
140	**Total number of word "may"**
141	**Total number of word "might"**
142	**Total number of word "must"**
143	**Total number of word "need"**
144	**Total number of word "ought"**
145	**Total number of word "shall"**
146	**Total number of word "should"**
147	**Total number of word "will"**
148	**Total number of word "would"**
149	**Total number of word "can't"**
150	**Total number of word "don't"**
151	**Total number of word "won't"**
152	**Total number of word "aren't"**
153	**Total number of word "isn't"**
154	**Total number of word "wasn't"**
155	**Total number of word "weren’t"**
156	**Total number of word "couldn't"**
157	**Total number of word "doesn't"**
158	**Total number of word "didn't"**
159	**Total number of word "haven't"**
160	**Total number of word "hasn't"**
161	**Total number of word "hadn't"**
162	**Total number of word "shouldn’t"**
163	**Total number of word "wouldn't"**
164	**Total number of word "anyhow"**
165	**Total number of word "as if"**
166	**Total number of word "agreed"**
167	**Total number of word "anytime"**
168	**Total number of word "as if"**
169	**Total number of word "awful"**
170	**Total number of word "bingo"**
171	**Total number of word "bless you"**
172	**Total number of word "bravo"**
173	**Total number of word "cheers"**
174	**Total number of word "crud"**
175	**Total number of word "goodness"**

176	**Total number of word "gosh"**
177	**Total number of word "hallelujah"**
178	**Total number of word "hey"**
179	**Total number of word "hi"**
180	**Total number of word "salute"**
181	**Total number of word "chaos"**
182	**Total number of word "darn"**
183	**Total number of word "boo"**
184	**Total number of word "behold"**
185	**Total number of word "blah"**
186	**Total number of word "dang"**
187	**Total number of word "golly"**
188	**Total number of word "gracious"**
189	**Total number of word "indeed"**
190	**Total number of word "my gosh"**
191	**Total number of word "shoot"**
192	**Total number of word "please"**
193	**Total number of word "rats"**
194	**Total number of word "shucks"**
195	**Total number of word "tut"**
196	**Total number of word "ahoy"**
197	**Total number of word "alas"**
198	**Total number of word "bam"**
199	**Total number of word "Atta girl"**
200	**Total number of word "batboy"**
201	**Total number of multiple question marks**
202	**Total number of multiple exclamation marks**
203	**Total number of ellipses**
204	**Hapaxlegomena**
205	**Hapaxdislegomena**
206	**Total number of blank lines**
207	**Average words per sentence**
208	**Vocabulary richness**
209	**Average characters per paragraph**
210	**Total number of word "aah"**
211	**Total number of word "aha"**
212	**Total number of word "ahem"**
213	**Total number of word "ahh"**
214	**Total number of word "argh"**
215	**Total number of word "aww"**
216	**Total number of word "aw"**
217	**Total number of word "bah"**
218	**Total number of word "boo"**
219	**Total number of word "booh"**
220	**Total number of word "brr"**

221	**Total number of word "duh"**
222	**Total number of word "eek"**
223	**Total number of word "eep"**
224	**Total number of word "eh"**
225	**Total number of word "eww"**
226	**Total number of word "gah"**
227	**Total number of word "gee"**
228	**Total number of word "grr"**
229	**Total number of word "egh"**
230	**Total number of word "hmm"**
231	**Total number of word "humph"**
232	**Total number of word "harumph"**
233	**Total number of word "huh"**
234	**Total number of word "hurrah"**
235	**Total number of word "ich"**
236	**Total number of word "yuk"**
237	**Total number of word "yak"**
238	**Total number of word "meh"**
239	**Total number of word "eh"**
240	**Total number of word "mhm"**
241	**Total number of word "uh-hu"**
242	**Total number of word "mm"**
243	**Total number of word "mmh"**
244	**Total number of word "muahaha"**
245	**Total number of word "hahaha"**
246	**Total number of word "mwahaha"**
247	**Total number of word "bwahaha"**
248	**Total number of word "nuh-uh"**
249	**Total number of word "oh"**
250	**Total number of word "oohlala"**
251	**Total number of word "ohlala"**
252	**Total number of word "ooh"**
253	**Total number of word "oomph"**
254	**Total number of word "umph"**
255	**Total number of word "oops"**
256	**Total number of word "ow"**
257	**Total number of word "oy"**
258	**Total number of word "pew"**
259	**Total number of word "pff"**
260	**Total number of word "phew"**
261	**Total number of word "psst"**
262	**Total number of word "sheesh"**
263	**Total number of word "jeez"**
264	**Total number of word "shh"**
265	**Total number of word "shoo"**

266	**Total number of word "tsk"**
267	**Total number of word "uh-uh"**
268	**Total number of word "uh oh"**
269	**Total number of word " oh"**
270	**Total number of word "uh uh"**
271	**Total number of word "uhh"**
272	**Total number of word "err"**
273	**Total number of word "wee"**
274	**Total number of word "whee"**
275	**Total number of word "whoa"**
276	**Total number of word "wow"**
277	**Total number of word "yahoo"**
278	**Total number of word "yay"**
279	**Total number of word "yeah"**
280	**Total number of word "yee haw"**
281	**Total number of word "yoohoo"**
282	**Total number of word "yah uh"**
283	**Total number of word "yuck"**
284	**Total number of word "mwah"**
285	**Total number of word "neener"**
286	**Total number of word "zowie"**
287	**Total number of word "niner"**
288	**Total number of word "zoinks"**
289	**Total number of word "yow"**
290	**Total number of word "yikes"**
291	**Total number of word "vavavoom"**
292	**Total number of word "ugh"**
293	**Total number of word "tchah"**
294	**Total number of word "rah"**
295	**Total number of word "sis boom bah"**
296	**Total number of word "shh"**
297	**Total number of word "ole"**
298	**Total number of word "olela"**
299	**Total number of word "lah de dah"**
300	**Total number of word "hup"**
301	**Total number of word "huppy"**
302	**Total number of word "ichh"**
303	**Total number of word "hubba"**
304	**Total number of word "ho hum"**
305	**Total number of word "ho ho"**
306	**Total number of word "hist"**
307	**Total number of LIWC's leisurehome money relig death assent nonfl filler features**
308	**Total number of paragraphs**
309	**Total number of LIWC's achieve features**

310	**Average sentences per paragraph**
311	**Average words per paragraph**
312	**Yule's K measure**
313	**Simpson's D measure**
314	**Sichel's S measure**
315	**Honore's R measure**
316	**Entropy**
317	**Total number of LIWC's WC features**
318	**Total number of LIWC's Wp features**
319	**Total number of LIWC's Qmark features**
320	**Total number of LIWC's Unique features**
321	**Total number of LIWC's Dic features**
322	**Total number of LIWC's Sixltr features**
323	**Total number of LIWC's funct features**
324	**Total number of LIWC's pronoun features**
325	**Total number of LIWC's ppron features**
326	**Total number of LIWC's i features**
327	**Total number of LIWC's we features**
328	**Total number of LIWC's you features**
329	**Total number of LIWC's she he features**
330	**Total number of LIWC's they features**
331	**Total number of LIWC's ipron features**
332	**Total number of LIWC's article features**
333	**Total number of LIWC's verb features**
334	**Total number of LIWC's auxverb features**
335	**Total number of LIWC's past features**
336	**Total number of LIWC's present features**
337	**Total number of LIWC's future features**
338	**Total number of LIWC's adverb features**
339	**Total number of LIWC's preps features**
340	**Total number of LIWC's conj features**
341	**Total number of LIWC's quant features**
342	**Total number of LIWC's number features**
343	**Total number of LIWC's swear features**
344	**Total number of LIWC's social features**
345	**Total number of LIWC's family features**
346	**Total number of LIWC's friend features**
347	**Total number of LIWC's humans features**
348	**Total number of LIWC's posemo features**
349	**Total number of LIWC's negemo features**
350	**Total number of LIWC's anx features**
351	**Total number of LIWC's anger features**
352	**Total number of LIWC's sad features**
353	**Total number of LIWC's cogmech features**
354	**Total number of LIWC's insight features**

355	**Total number of LIWC's cause features**
356	**Total number of LIWC's discrep features**
357	**Total number of LIWC's tentat features**
358	**Total number of LIWC's certain features**
359	**Total number of LIWC's inhib features**
360	**Total number of LIWC's incl features**
361	**Total number of LIWC's excl features**
362	**Total number of LIWC's percept features**
363	**Total number of LIWC's see features**
364	**Total number of LIWC's hear features**
365	**Total number of LIWC's feel features**
366	**Total number of LIWC's bio features**
367	**Total number of LIWC's body features**
368	**Total number of LIWC's health features**
369	**Total number of LIWC's sexual features**
370	**Total number of LIWC's ingest features**
371	**Total number of LIWC's relative features**
372	**Total number of LIWC's motion features**
373	**Total number of LIWC's space features**
374	**Total number of LIWC's time features**