Gender Identification from Text

Arjun Mukherjee
Assistant Professor
Department of CS
University of Houston
arjun@cs.uh.edu

Vishal Pallerla
Graduate Student
Department of CS
University of Houston
vpallerla@uh.edu

Guru Pavan Kamakolanu
Graduate Student
Department of CS
University of Houston
gkamakolanu@uh.edu

Abstract— The identification of an author's gender from a text has turned into a prominent research area within the scope of text categorization. The number of users of social network applications based on text, such as Twitter, Facebook, and other text messaging services, has grown swiftly over the past few decades. Subsequently, text turned out to be a standout amongst the most essential and predominant media sorts on the Internet. This project aims to determine the gender of an author from an arbitrary piece of text such as, for example a blog article. This field of research has gathered the enthusiasm of the specialists since a few people fake their gender in Internet. The psychology of linguistics indicates how intently the words and writing styles people use correlate with their gender. Different feature sets have been used by researchers in recent periods to identify the gender of an author but identifying feature sets is a field on which still research is going on. In our implementation, various feature sets were carefully chosen for the gender identification problem. They can be categorized as character-based features, word-based features, syntactic-based features, structure-based features, and the function words that an author used in a text. Machine learning algorithms (like support vector machine, Naïve Bayes and AdaBoost decision tree) are then designed for gender identification based on the proposed features. Experiments show that function words, word-based features and structural features are substantial gender classifiers.

Index Terms— Classifier, Gender Identification, Machine Learning, Stylistic Features, POS tagging

1 Introduction

The fast growth of Internet has created numerous ways for sharing information in cyberspace. The number of social network users, ecommerce and other web applications are increasing day-to-day. Web Blogs are one such applications where people write informally but they are becoming a major source of information over web. This evolution has given rise to a variety of misuses, as anonymity is a significant characteristic of Internet-based communities [1]. Users might not disclose their real identities in terms of name, age, gender and address in Internet communities. Therefore, it has become important to stratetize an effective method for identity-tracing online.

Gender identification always plays a major role, as it can be misused for email forgery, online communities and disguising as opposite gender for money [2]. Gender is used for experimentation using stylistics as the blogs generally contain this information provided by the author. Style in writing is a result of the subconscious habit of the writer of using one form over several available options to present the same thing [3].

Research shows that literature used by an individual in writing specifies their mental health and emotion. Also, everyone has their own writing style which is referred to as author profile [4]. Advancement of computers led to wide use of stylometry for identifying author ownership. This project focusses on investigating blog articles and extract features from them that can potentially divide authors into males and females.

Finding gender of author is a classification problem with two classes. When submitting an article, it should be assigned to class F if the author is female or class M if the author is male. To design such a classifier, we need to get feature sets from the text that remain the same for most authors of the same gender. Gender identification process can be divided as collecting suitable articles to make up dataset, identifying features that indicate gender, exctracting features from articles and building a model.

2 Related Work

The research in last few decades on usage of language pattern by different social groups was constrained due to unavailability of sufficient annotated data. Analysis of effects of bloggers age and gender from weblogs,

- *Arjun Mukherjee is with the Department of Computer Science at University of Houston, Houston, TX 77004. E-mail: arjun@cs.uh.edu*
- *Vishal Pallerla is with the Department of Computer Science at University of Houston, Houston, TX 77004. E-mail: vpallerla@uh.edu*
- *Guru Pavan Kamakolanu is with the Department of Computer Science at University of Houston, Houston, TX 77004. E-mail: gkamakolanu@uh.edu*

Manuscript received 11 Dec. 2016

based on usage of keywords, parts of speech and other grammatical constructs, has been presented in Learning Age and Gender of Blogger. Age linked variations had been reported by Pennebaker, et al., Pennebaker and Stoneand Burger and Henderson, 2006. J. Holmes distinguished characteristics of male and female linguistic styles.

Recent papers on gender classification of blogs (e.g., Schler et al., 2006, Argamon et al., 2007; Yan and Yan, 2006; Nowsonet al., 2005). These systems use function/ content words, POS tag features, word classes (Schler et al., 2006), content word classes (Argamon et al., 2007), results of dictionary based content analysis, POS unigram (Yan and Yan, 2006), and personality types (Nowson et al., 2005) to capture stylistic behavior of authors writings for classifying gender. these works use only one or a subset of the classes of features. None of them uses all features for classification learning. Given the complexity of blog posts, it makes sense to apply all classes of features jointly to classify genders. Improving gender based classification and age based classification is also done in paper Improving Gender Classification of Blog Authors by Bing Liu and Arjun Mukherjee.

3 Methodology and Implementation

3.1 Data Set

The goal of this project was to determine whether the author of a submitted text was male or female. In this regard, there was a need for dataset containing various writing samples divided per gender type, which makes it easier for us to extract each gender writing styles. Different datasets were available to the researchers, each of which had different categories considering age, gender, the type of text, etc. Additionally, texts that are written in a more informal style are more suitable for this type of research. Our experimental results are based on a real-life blog data set collected from many blog hosting sites. We collected dataset used by Bing Liu and Arjun Mukherjee for their paper Improving Gender Classification of Blog Authors [3], that dataset is having around 3200 entries

Dataset consists of articles and the gender of the author for each row.

Article	Gender

Composition of articles by gender in the dataset is also very good as it consisted of approximately equal male and female articles.

Gender	No of Articles
Male	1678 (52.01%)
Female	1548 (47.99%)

Table 1. Distribution of Articles

3.2 Methodology

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. 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 occupations. As there is no real way of establishing an author's authenticity, research has sought to find a means for detecting fake authors. To hide their identity, fake authors tend to portray themselves as other gender and in those cases, the only way of establishing the original author is to do so through the writing style and structure of the article. In this research, some structural and linguistic features have been applied in a bid to train a learning engine.

In the process of gender identification, 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 then predict the gender of the author after applying this labelled dataset by comparing the test article style to the features of writing styles that are already available. This can result in the system making a judgment about whether a text was written by male or female.

3.2.1 Feature Set Description

Variations in the systems that ladies and men use to express a similar subject have been important to numerous specialists in the field. Past few decades have shown critical changes as far as how ladies and men utilize dialect. For instance, in a content containing sport-related words, for example, 'cricket', 'beat', 'champion", mentor' and 'class', it has been found that the creator of a content containing these words will most probably male rather than a female. Then again, for a content containing words, for example, "pink" and 'boyfriend', the likelihood of the essayist being female has more chances than male [5].

Distinctive genders utilize many similar words in their written work, yet with different context. For instance, when guys discuss 'day to day life', they tend to mean their work; when females utilize a similar expression, they will probably be talking about adoration and the more profound parts of life. Another case concerns the utilization of "dress", which guys tend to use for tuxedos, while females utilize the word when discussing marriage outfits and night dresses.

Robin Lakoff, a contributor to feminist linguistics, trusts that females utilize weaker and even more sweet-sounding words, for example, "dear" and 'oh my goodness', while guys tend to utilize more grounded words, for example, "damn". Then again, there are words that both sexual orientations utilize, however with various recurrence.

Ladies tend to utilize escalating adverbs like "extremely" or "truly" and question marks in their written work. For the most part, in their discussions, ladies make aberrant requests while men tend to utilize more orders; ladies tend to chat more intently to standard syntactic dialect than men, who talk more persuasive. For instance, when a lady needs to ask others out to supper, she may compose, "Does anyone needs to go out for dinner???" On the other hand, a man may compose, "Let‟s go out for supper". The length of the sentence is another element that can be utilized as a measure to differentiate genders; sentences composed by females are longer than those composed by men. As far as subject, ladies speak more about individual and enthusiastic viewpoints than men, who tend to speak more about actuality based and less emotional subjects [6].

Stylometry is the investigation of how individuals judge others as indicated by their composition style. Stylometry can not just be utilized to recognize a written work style, yet can likewise help with distinguishing the gender of the writer [7]. Next, we discuss the features that helps us in separating writing based on gender.

a. Character based features:

This section discusses the text analysis by considering each of the characters included therein. First, we counted the total number of characters, including all the letters, digits, punctuations, spaces, etc. Then we calculated number of words used in a review, distinct number of words, number of special characters used, occurrence of sequence of words.

Algorithm:

for each review:

Pre-process the review

words ⟵ review.split(“ “)

count ⟵ length(words)

b. Word based features:

This part discusses the analysis of words including total number of words, the average number of characters per word, the total number of different words that are available in an article.

We observed a new word based feature which is excitement feature i.e using yeahhhhhhhh instead of yeah, oh my god!!!!!!!!!! Instead of oh my god! This observation was made when going through many articles and result was this excitement and stressing of words was found in female authors mostly than male. So, we made use of this feature which played a significant role in improving the accuracy.

Algorithm:

Distinct word count

for each word in words:

  if word in distinct_words:

    distinct[word] ⟵ distincy[word]+1

  endif

endfor

  else:

    distinct.update({Word:1})

Special character count

Create a list of special characters

for each review:

  for each special character

    count⟵ length(special character)

    save the count as a feature

  endfor

endfor

c. Bag of words:

Bag of Words (BoW) is an algorithm that counts how many times a word appears in an article/sentence/document. Those word counts allow us to compare them and gauge their similarities for applications like search, document classification and as one of the features in machine learning. BoW is a method for preparing text for input in a deep-learning net. This specific strategy involves tokenization, counting and normalization known as the Bag of Words or “Bag of n-grams” representation. Word events depict records while totally overlooking the relative position data of the words

in the archive. Bag of Words are implemented for different ngrams such as bigrams, trigrams etc. For bag of words, we have removed the punctuations, spaces so that only words are considered.

d. POS tagging:

Automatic assignment of descriptors to the given tokens is called Tagging. The descriptor is called tag. The tag may indicate one of the parts-of-speech, semantic information, and so on. So, tagging is a kind of classification**.**

The process of assigning one of the parts of speech to the given word is called Parts of Speech tagging. It is commonly referred to as POS tagging. Parts of speech include nouns, verbs, adverbs, adjectives, pronouns, conjunction, and their sub-categories.

Parts of Speech tagger or POS tagger is a program that does this job. Taggers use several kinds of information: dictionaries, lexicons, rules, and so on. Dictionaries have category or categories of a word. That is a word may belong to more than one category. For example, run is both noun and verb. Taggers use probabilistic information to solve this ambiguity.

There are mainly two types of taggers: rule-based and stochastic. Rule-based taggers use hand-written rules to distinguish the tag ambiguity. Stochastic taggers are either HMM based, choosing the tag sequence which maximizes the product of word likelihood and tag sequence probability, or cue-based, using decision trees or maximum entropy models to combine probabilistic features.

1. Tokenization: The given text is divided into tokens so that they can be used for further analysis. The tokens may be words, punctuation marks, and utterance boundaries.
2. Ambiguity look-up: This is to use lexicon and a guessor for unknown words. While lexicon provides list of word forms and their likely parts of speech, guessors analyze unknown tokens. Compiler or interpreter, lexicon and guessor make what is known as lexical analyzer.
3. Ambiguity Resolution: This is also called disambiguation. Disambiguation is based on information about word such as the probability of the word. For example, power is more likely used as noun than as verb. Disambiguation is also based on contextual information or word/tag sequences. For example, the model might prefer noun analyses over verb analyses if the preceding word is a preposition or article. Disambiguation is the most difficult problem in tagging.

Algorithm:

```
for each review:
   text◄— tokenize(review)
   text◄— pos_tag(text)
   for each (w1, t1)…(w3,t3) in trigrams(text):
     if (w1◄—"extracted w1" &..w3◄—"extracted w3)
        extract word-sequence features that are
        ranked high among top ten.
     endif
   for each word tag (three in sequence) in ngrams:
     if (word sequence matches the extracted se-
     quence):
        feature_list.append(count of word sequence)
     endif
   endfor
 endfor
endfor
```

e. TF-IDF:

TF-IDF (Term Frequency, Inverse Document Frequency) is a basic technique to compute the relevancy of a document with respect to a particular term. "Term" is a generalized element contains within a document. A "term" is a generalized idea of what a document contains. (E.g. a term can be a word, a phrase, or a concept). Intuitively, the relevancy of a document to a term can be calculated from the percentage of that term shows up in the document (ie: the count of the term in that document divide by the total number of terms in it). We called this the "term frequency" On the other hand, if this is a very common term which appears in many other documents, then its relevancy should be reduced. (ie: the count of documents having this term divided by total number of documents). We called this the "document frequency" The overall relevancy of a document with respect to a term can be computed using both the term frequency and document frequency.
relevancy = term frequency * log (1 / document frequency) This is called tf-idf. A "document" can be considered as a multi-dimensional vector where each dimension represents a term with the tf-idf as its value.

The very first step which is performed to find tf-idf is to generate a set of text files for each document separately and perform STOP word removal on them. STOP words such as for, a, the, to etc. In the next step, we perform stemming. For e.g., we have words such as care and careful the stemming is performed and these are counted as one word.

step1() gets rid of plurals and -ed or -ing.
e.g. caresses -> caress
ponies -> poni
cats -> cat
agreed -> agree
disabled -> disable

step2() turns terminal y to i when there is another vowel in the stem. step3() maps double suffices to single ones. so -ization (= -ize plus -ation) maps to -ize etc. step4() deals with -ic-, -full, -ness etc. similar strategy to step3. step5() takes off -ant, -ence etc., in context <c>vcvc<v>.

For a set of 100 blogs we found 4523 words which were present. Once the task of stemming is complete, we obtain the raw data then the number of occurrences of each word is counted for all the documents. Now tf-idf is performed as mentioned above and weight of every word is calculated for each blog entry. A vector matrix is arranged in the format of .arff file which is fed to WEKA for classification. WEKA trains on the first 20% of data and for then next 80% it classifies the dataset and gives out the result. Accuracy we got for TF-IDF is 60%

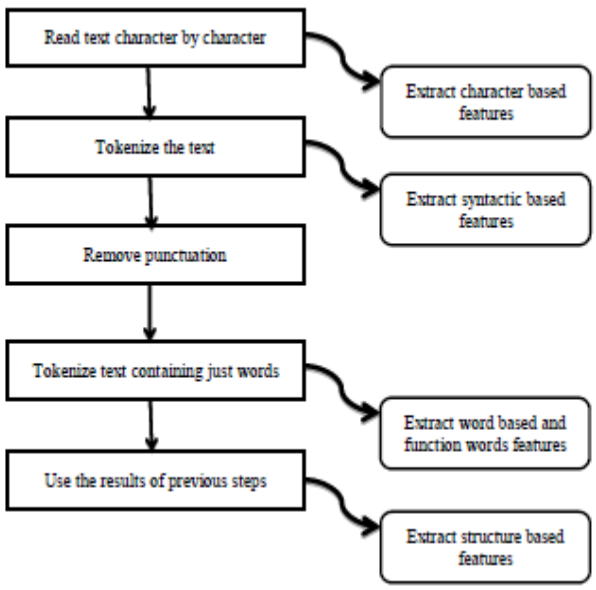

Fig 1. Extraction of Features

3.2.2 Classifiers

After the features are extracted, a maching learning algorithm needs to be implemented on the training data with extracted features. 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 per the defined feature sets.

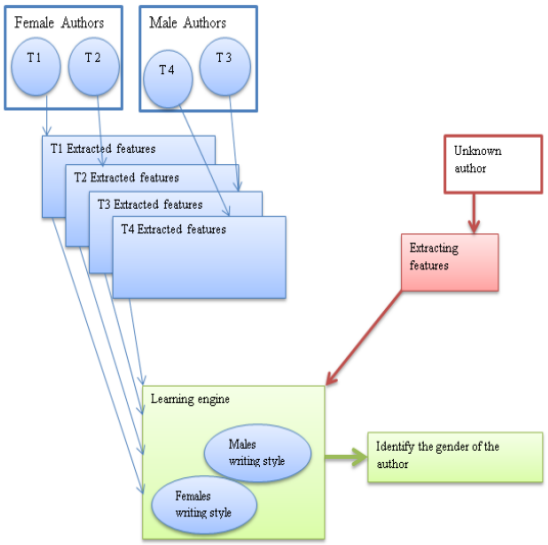

Fig 2. Process of identifying gender of an author

After training the system, the learning engine extracted female writing and male writing styles. In this stage of the analysis, when submitting an article written by a gender-unknown author, the system extracted the specific features of that text. Then, comparing the extracted features with the existing writing styles, the system could predict the gender of the author.

There are several algorithms that can be used for classification techniques. In the following section, the advantages and disadvantages of the naive Bayes classifier, logistic regression, the decision tree, support vector machine and Naïve Bayesian algorithms are briefly described.

a. Naïve 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. Naive 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.

b. Decision Trees

The reason that decision trees have become

popular is that they are fast in giving the results, can be expanded and there is no need to set many 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 must be rebuilt from scratch[8]

c. 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.

Overall, it has been stated by several 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 [9]

d. SGD Classifier

SGDClassifier supports averaged SGD (ASGD). Averaging can be enabled by setting `average=True`. ASGD works by averaging the coefficients of the plain SGD over each iteration over a sample. When using ASGD the learning rate can be larger and even constant leading on some datasets to a speed up in training time.

4 Experiments and Results

We played with combination of different features for checking the accuracy with various classifiers. Results of the different classifiers implemented are shown below and their analysis is described.

4.1 Experiment 1 - Bag of Words

This specific strategy involves tokenization, counting and normalization known as the Bag of Words or "Bag of n-grams" representation. Word events depict records while totally overlooking the relative position data of the words in the archive. Bag of Words are implemented for different ngrams such as bigrams, trigrams etc.
The results for 'Bag of Words' using different classifiers are shown below.

Classifier	Ngram (1,1)	Ngram (1,2)	Ngram (1,3)
SVC	54.91	54.44	53.82
SVC-Linear	58.53	55.91	54.91
Random Forest	50.70	52.26	56.44

Table 2. Bag of Words Experiment

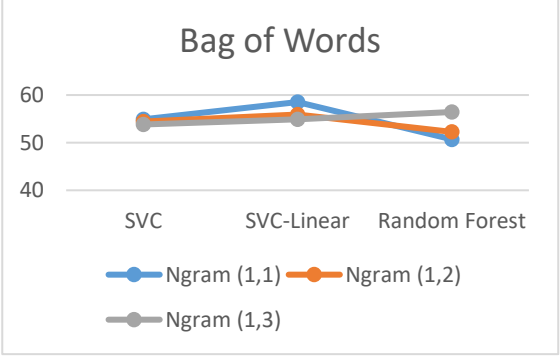

Fig 3. Bag of Words with different Classifiers

4.2 Experiment 2

The features considered for this experiment are word count and distinct word count. Accuracy has been calculated using different classifiers and the results are shown in the below table.

Classifier	Accuracy
SVC	51.32
SVC-Linear	49.76
Naïve Bayes	48.67
AdaBoost	52.73
Random Forest	51.79
Gradient Boost	48.82
SGD	49.76
Extra tree	50.39
Decision tree	49.18

Table 3. Experiment 2 Accuracies

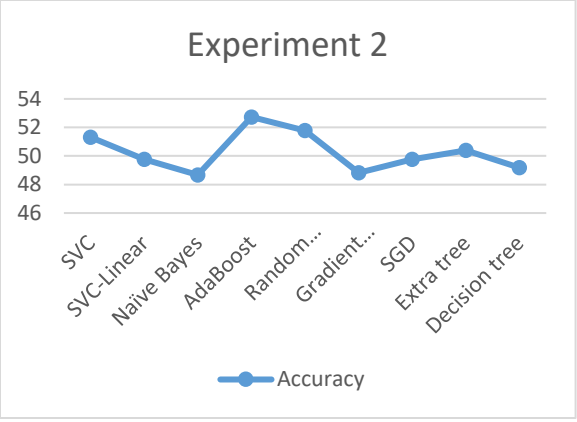

Fig 4. Accuracy of Classifiers with random features

4.3 Experiment 3

The features considered for this experiment are number of words used in a review, distinct number of words and occurrence of sequence of words extracted using POS tagging. Only, the top ranked ten sequences are used from those sequences of words that have been extracted using POS tagging. Accuracy has been calculated using different classifiers and the results are shown in the below table.

Classifier	Accuracy
SVC	55.67
SVC-Linear	55.03
Naïve Bayes	50.89
AdaBoost	51.04
Random Forest	51.79
Gradient Boost	49.73
SGD	50.23
Extra tree	52.17
Decision tree	52.70

Table 4. Experiment 3 Accuracies

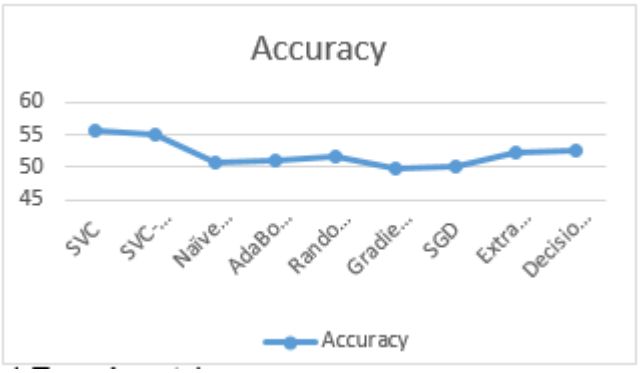

Fig 5. Accuracy of Classifiers with random features

Addition of "sequence of words" feature played a significant role in improving the accuracy of the model.

4.4 Experiment 4

The features considered for this experiment are number of words used in a review, distinct number of words, number of special characters used, occurrence of sequence of words extracted using POS tagging and excitement feature of humans. Only, the top ten sequences are used from those sequences of words that have been extracted using POS tagging. Accuracy has been calculated using different classifiers and the results are shown in the below table.

Classifier	Accuracy
SVC	63.57
SVC-Linear	66.83
Naïve Bayes	55.62
AdaBoost	60.73
Random Forest	54.07
Gradient Boost	61.08
SGD	59.13
Extra tree	55.83
Decision tree	57.05

Table 5. Experiment 4 Accuracies

The features "no. of special characters", "sequence of words pattern", "excitement feature" played a significant role in increasing the accuracy of predicting the author's gender.

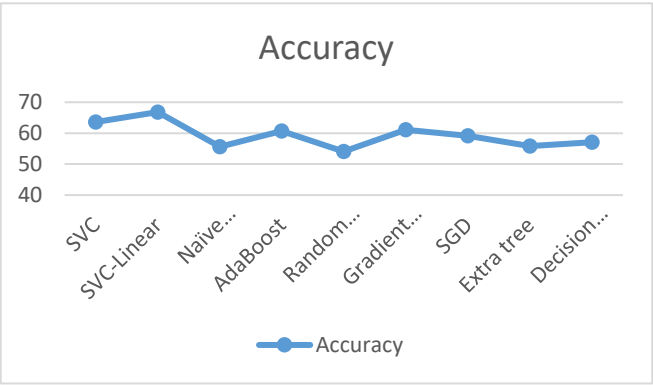

Fig 6. Accuracy of Classifiers with random features

4.5 Experiment 5

Using all the features from first experiment, SGD classifier can be implemented using combination of values for the attributes "loss function" and "penalty".

Loss	Penalty	Accuracy
Hinge	L2	52.76
Log	L2	50.73
Hinge	L1	59.13
Modified_huber	L2	52.92

Table 6. Experiment 5 Accuracies

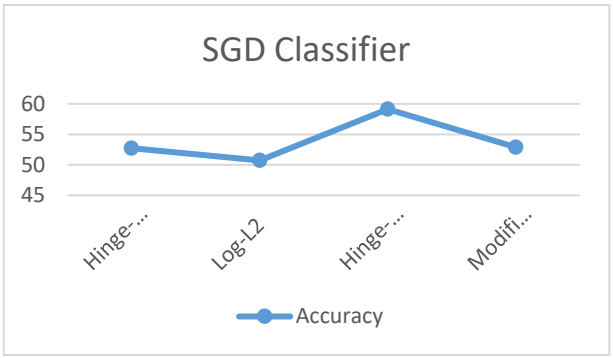

Fig 7. Accuracy of Classifiers with random features

5 Conclusion and Future Work

In this project, we introduced a classifier-based implementation of author gender identification from text. Gender identification from content concerns the exchange amongst etymological and composing styles, and in addition those words that are usually utilized by one sex. The outcomes yielded by different experiments showed the upsides of the diverse classifiers, and in addition include set determination.

The experimental outcomes demonstrated that outlining a proper list of features by considering semantics and elements that correspond to sex is of high significance. It ought to be noticed that a few components, for example, certain addition words were not regular in any sexual orientation, while one sex for the most part utilized a few words. Moreover, by expelling words that were exceptional among both sexes we can enhance the list of features.

Selection of classifier is of high importance in this subject area. The results identified showed that support vector machines outperform Bayesian classification. After evaluating the advantages and disadvantages of each classifier, the linear support vector machine classifier appears to be the best candidate for author gender identification from blog texts.

Further in future, many different methods such as MI, Chi-square (x2) and others can be incorporated with the existing one to increase the efficiency of the project. Different classification algorithms such as SVM, SVM_R can be used for better result.

6 Contribution

Vishal kumar Pallerla

- Data Cleansing & Data Pre-Processing
- Extracting character based features
- Implementing Bag of Words
- Implementing random forest classifier.
- Implementing extra tree classifier
- Implementing Adaboost and Gradient boost classifier
- Report (Equally Shared)

Guru Pavan Kumar Kamakolanu

- Importing excel data into SQL
- Extracting word count, distinct number of word features.
- Extracting excitement feature of humane.
- POS tagging.
- Using POS tagging, word sequences are extracted from reviews based on their ranking.
- Implementing Naïve bayes Classifier, SVM classifier, Linear SVM Classifier and SGD classifier
- Report (Equally shared)

7 References

[1] http://www.svms.org/history.html
[2] http://trec.nist.gov/data/reuters/ reuters.html
[3] http://www2.cs.uh.edu/~arjun/papers/EMNLP10_Gender.pdf
[4] http://www.cs.cmu.edu/~./enron/
[5] Benno Stein · Nedim Lipka · Peter Prettenhofer. (2010). Intrinsic Plagiarism Analysis. Language Resources and Evaluation
[6] Breiman, L. (2006). Random forests. Pattern Recognition in Remote Sensing

[7] Chen, H. (2005). Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems: special issueon AI for Homeland Security*
[8] Jordan, M. I. (2004). Soft Margin SVM. Berkeley University of california.
[9] Kaizhu Huang, Zhangbing Zhou,Irwin King, Michael R. Lyu. (2003). Improving Naive Bayesian Classifier by Discriminative Training.
[10] http://irep.emu.edu.tr:8080/jspui/bitstream/11129/1845/1/RezaeiAtoosa.pdf