**Chapter 1**

**INTRODUCTION**

This chapter introduces the problem of author ship identification in online Hindi texts and motivation for it. Section 1 will tell you about authorship identification. Section 2 will shed light on the earlier work done in this field and their success in this field.

* 1. **Context**

Authorship identification is the task of identifying of most likely author of a document based on the collection of already labelled documents. It uses the relationship between an author and his writing style. Writing styles are different from person to person since different persons are different in their word choices, sentence structures and usage of punctuations.

Doing it for online text has many reasons for it. Online text can be blogs, forums, newspaper and emails, messages. Internet has become a place of fraud or anonymous abuse .Every day we hear about online anonymous abuse. In Anonymous abuse, culprits can be discovered with their unique writing style. Plagiarism detection is another advantage of doing it. Often, there is dispute for plagiarism or copyright issues. One who has done Plagiarism can easily be detected with the help of the writing style in their documents. Unique writing style works in the same way as our fingerprints. As fingerprints can identify a person, writing style can too identify the author of an unknown document.

* 1. **Characteristic of Online Text**

Unlike text in books, literary works ,novels, magazines, where we can get lot of text for a particular author, in online texts like blogs, discussion forums, email ,etc. we can get only a small amount of information. The type of information which can trace out the author also differs. Greetings, Signature, smileys, short abbreviations occur in an online text which is different from our text in books.

# 1.3 Motivation

A lot of work has been done in author attribution when it comes to languages like English, Chinese, and Dutch. But no work has been done for Hindi which is one of the most widely used languages. Hindi as a language is more rich if compared to languages like English. To get an idea, Hindi has alphabets whereas English has only 26 alphabets.

**Chapter 2**

**Literature Review**

A lot of work has been done by the researchers and scholars in authorship identification in the past. Authorship identification has been the subject of study by the scholars from a very long time. Every researcher has used different features, different datasets and different techniques for classifying the author.

**2.1** **Corwin Mendelhall**

The first ever work in author identification was done by Corwin Mendelhal who used average word length as a measure of author style. He successfully classified the documents written by Shakespeare and Bacon using this approach.[1]

**2.2 Identifying author of federalist papers by Mostellar and Wallace**

The Federalist papers were a series of political papers published in year 1788. There was a long time dispute regarding the authorship of these papers. Mosteller and Wallace (1964) used the known texts of Hamilton and Madison to find the word length variation and word usage in their texts. They then used these criteria as the features and used the Bayesian probability to classify the disputed papers. These papers were identified to be written by Madison.[2]

**2.3 Corney work on Email data**

When we talk about online text, we cannot leave email message. Email data has 2 characteristics. Firstly, content of email data is very small as compare to literary works. Secondly, it has some unique features such as author signature, greetings and farewell messages. Corney worked on the email dataset. A total of 170 style marker attributes and 21 structural features were used in his experiment.

**Structural Attribute Type**

* Has a greeting Acknowledgement
* Uses a farewell acknowledgment
* Contains signature text
* No of Attachments.
* Position of requoted text within email body
* Html tag frequency distribution/total no of Html tags

**Style Marker Attribute Type**

* Average sentence length
* Average word length
* Vocabulary richness
* Total no of function words/total no of words
* Total no of short words/total no of words
* Hapax legonema - words which occur only once
* Total no of upper-case characters in words/total no of characters in the email body
* Word length frequency distribution
* Total no of punctuations/total no of characters in the email body

A corpus of 156 email documents written by the 3 authors was taken. They used Support Vector Machine as the classifier with RBF and linear kernel. They concluded that structural features alone are not good for identifying authors but if combined with other style markers they can certainly help in improving the accuracy.

**2.4 Rong Zheng work on two different languages**

In 2006, Rong Zheng carried out the work in two different languages Chinese and English. Oriental languages such as Chinese, such word boundaries often do not exist and words are adjacent to each other in a sentence. In case of English, data was collected from English Internet Newsgroup Message while In case of Chinese data came from Chinese Bulletin Board system. For both languages, total no of authors were 20. Four types of features were used lexical, syntactical, structural and content-specific. He called those features F1 (lexical), F2 (syntactical), F3 (structural), and F4 (content-specific. The three methods of classification used were Decision Tree Classifier (C4.5), neural networks and SVM. They concluded that combination of F1,F2,F3 and F4 gives the best accuracy.

F1 > F1+F2 > F1+F2+F3 > F1+F2+F3+F4

SVM outperformed Neural Network and Decision Tree classifier when all the features were included.

**2.5 Marcia Fissete work on Dutch Language**

In 2010, Marcia Fissete carried worked on short texts written in Dutch language. Data was collected from Dutch message board. About 25 messages per author were taken. The total number of authors was 40.

**Features used were:**

* Unigrams excluding smileys
* Unigrams including smileys
* Bigrams excluding smileys
* Bigrams including smileys
* Triplets
* Unigrams +triplets
* Bigrams + triplets

SVM was used as the classifier. Unigrams performed better than bigrams and result became better when smileys were added to the unigrams. For unigrams and bigrams the performance improves when smileys are included.

.

**Chapter 3**

**Design**

# 3.1 General Procedure: predicting the author of an unknown document goes through many steps. The main steps which come are:

**Data Collection:**

In this step, data needed for Authorship identification is collected. Data is collected from email message, online blogs and some discussion forum.

**Feature Generation:**

For machine learning algorithm to understand our data, Data needs to be converted into numerical values. Raw data is converted into a feature vector. These features in case of authorship identification are lexical, structural, syntactical and content specific features.with these features, the writing style of different authors is presented in a tabular form.

**Feature Normalisation:**

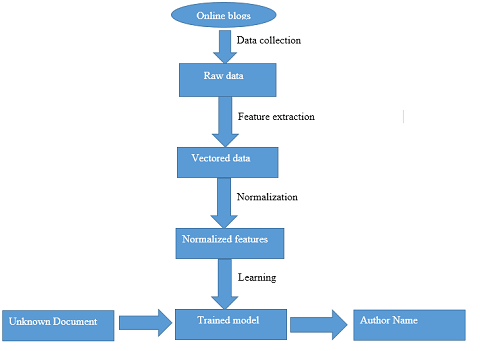
After feature generation, feature Normalisation is carried out so that our machine learning algorithm can learn better and in a small time**.**

**Generating Model**

The data is divided into 2 parts. Training part which is given to the machine learning algorithm for learning the pattern inside it. Testing part which is used to test our model

**Identifying the author of Unknown document**

Each sample in the testing set is tested to predict the author of unknown document and performance of the system is checked over the parameters of accuracy.



**3.2Features**

The features which are used in authorship identification are called Stylometric features. A lot of researchers have carried their work in finding out the features which characterize an author style. These features are grouped into four categories.

**3.2.1** **Lexical Features:**

Lexical features can further be divided into 2 types

**Character Based**:

* Total no of characters in a document
* Total no of alphabetic characters
* Total no of digit characters
* Total no of uppercase letters
* Total no of lowercase letters
* Frequency of characters
* Unigrams: Sequence of 1 character
* Bigrams: Sequence of 2 characters
* Trigrams: Sequence of 3 characters
* Tetra grams: Sequence of 4 characters

**Word Based:**

* Word unigrams
* Word Bigrams
* Word Trigrams
* Total no of words in the document
* Total no of unique words in the document (HAPAX)
* Average word length
* Most frequently used word in the document
* Word length frequency distribution

**3.2.2 Syntactical features**

Syntactical features are considered more reliable than the lexical features in text classification. However they are language dependent as they require the use of a parser.

Part of speech tagging is a syntactical feature where each token in the document is labelled with its tag. The tag here refers to whether the particular word used in that context is used as a verb, noun, preposition, etc. There are two main approaches to POS tagging:

* Rule based tagging

Rule based taggers depend on a dictionary to get possible tags for each word to be tagged. Hand-written rules are used to identify the correct tag when a word has more than one possible tag. Disambiguation is done by analysing the linguistic features of the word, its preceding word, its following word and other aspects. For example, if the preceding word is article then the word in question must be noun. This information is coded in the form of rules.

* Stochastic tagging

Stochastic taggers such as HMM based tagger choose the tag sequence which maximizes the product of word likelihood. HMM based taggers need a training phase during which each tag is represented as a state. State transition probabilities are learned which are later used for calculating the likelihood of a given tag sequence.

**3.2.3** **Structural features:**

Structural features show the layout of writing. The following are some features which have been found out by Researchers**.**

* Total number of lines
* Total number of sentences
* Total number of paragraphs
* Number of sentences per paragraph
* Number of characters per paragraph
* Number of words per paragraph
* Has a greeting
* Has quoted content
* Position of quoted content
* Indentation of paragraph

**3.2.4** **Content –Specific features**

Sometimes, we use typical words in our document which just relate to document topic only. Some authors will write documents relating to certain topics like corruption while some will write regarding like engineering. It is most probable that author writing documents for corruption will use words like growth, economy etc. while author associated with engineering will use words like students, ,placements .As a result , such words become characterizer of a particular author .These words are therefore called content –specific features

**3.4 Machine Learning Algorithm**

**3.4.1 Logistic Regression:**

Logistic regression is one of the simplest machine learning algorithms. The underlying assumption used in the algorithm is that there is a linear relationship between target and predictor variables. The value of the target variable is given by



where θ is a vector containing the weight corresponding to each feature.

The error associated with each training example is given by



The cost function is given by



This cost function is optimized by using Gradient descent algorithm.

**3.4.2 Naïve Bayes**

Naïve Bayes uses the Bayes theorem to calculate the value of the target variable from predictor variables. It makes a naive assumption that the predictor variables are independent of each other.

Given a test example where the predictor variables are X1, X2 …XN, the value of target variable is calculated by the formula



All probabilities on the right hand side are calculated using the training data set. The target variable is assigned that class corresponding to which the numerator on the right hand side is maximum.

Since the predictor variables are independent of each other



**3.4.3 SVM**

Support vector machines or in short SVM use a hyperplane to separate the classes from each other. They are also known as large margin classifier since they try to maximize the distance between the hyperplane and each class. The equation of the hyperplane is



where *w* is the weight matrix for the features.

The equation to be optimized is



subject to constraints



The above equations when followed lead to a hard margin classifier. A hard margin classifier finds a hyperplane which completely separates the two classes. The problem with a hard margin classifier is that a single training example can completely shift the hyperplane leading to a plane which separates the two classes but has a narrow margin. A narrow margin ultimately leads to overfitting. We therefore need a classifier which also pays due importance to large margin. This type of classifier is called soft margin classifier. For a soft margin classifier the equation to be optimized is



Subject to constraints



where ε is slack variable and C is the regularisation parameter.

The parameter C controls the trade-off between large margin(less overfitting) and narrow margin(more overfitting).

**3.4.4 Random Forest**

Random forest is an ensemble classifier. Multiple decision trees are trained in parallel and the final decision is made by taking the mode of the decisions given by all trees. During the training phase different decision trees are trained on different subsets of data. As a result of this the complete model is not prone to overfitting. This is the main advantage of RandomForest over a single decision tree. RandomForest classifier takes certain parameters like number of trees to be trained, max depth of each tree, number of features to consider at each split. All these features have to be tuned to get the best possible results.

**3.4.5 K nearest neighbours**

This is a non-parametric model as it does not have a learning phase. A test example is classified by taking the mode of classes for k nearest neighbours. The optimum value of k can be found out by parameter tuning.

**Chapter 4**

**Implementation**

**4.1 Data**

Data was collected from various Hindi blogs. Blog posts of 18 authors have been taken. For each author 14 blog posts have been taken. For a large part of our project we have worked with only 5 authors. The entire set of 18 authors is only used to show how our algorithms fare when the number of authors starts increasing.

**4.2 Formation of feature vector**

Raw data as such cannot be used by learning algorithms. It needs to be converted into a feature vector. A feature vector is a representation of data which can be used by machine learning algorithms. Various models can be used for generating the feature vector. The model used depends upon the type of style marker we want to use for identifying the author.

We carried out our work using different feature models:

**4.2.1 Bag of words Model**

Frequency of each word in the document is considered as a feature. To identify Hindi words we build a regular expression and passed this regex to the CountVectoriser function present in python sklearn library.



0x900 is the unicode point for first character of Hindi alphabet. 0x97F is the unicode point for the last character of the Hindi alphabet. The regular expression identifies words which have characters only between these two codepoints.

**4.2.2** **Character n grams**

A character n gram is a sequence of n characters. For bigrams we take every two adjacent characters as a feature. Similarly for trigrams we take a group of 3 characters. In our work, we have three types of n grams:

* Bigrams
* Trigrams
* Bigrams + Trigrams

**4.2.3 Function words**

Function words are words that have little lexical meaning. Their sole role is to express grammatical relationship between other words in a sentence. Prepositions, Pronouns and Conjunctions are examples of function words. There are three main reasons for using function words in lieu of other markers. First, because of their high frequency in a written text, function words are very difficult to consciously control, which minimizes the risk of false attribution. The second is that function words, unlike content words, are more independent from the topic or the genre of the text, so one should not expect to find great differences of frequencies across different texts written by the same authors on different topics. All authors writing in the same language and period are bound to use the very same function words. Function words are therefore a reliable base for textual comparison.

**Non availability of Hindi function words**

Unfortunately no reliable collection of Hindi words is available. In the absence of any collection, the only choice was to manually collect each function word. But this was too much time consuming. So we adopted a different strategy. Our strategy is based on the principle that function words are the most frequent words across a set of documents. This can be expressed as

*If x is a function word → x is a frequent word*

The above implication basically means that if x is a function word then it is also a frequent word. This implication always holds. Now we took the converse of this implication which formulates as

*If x is a frequent word → x is a function word*

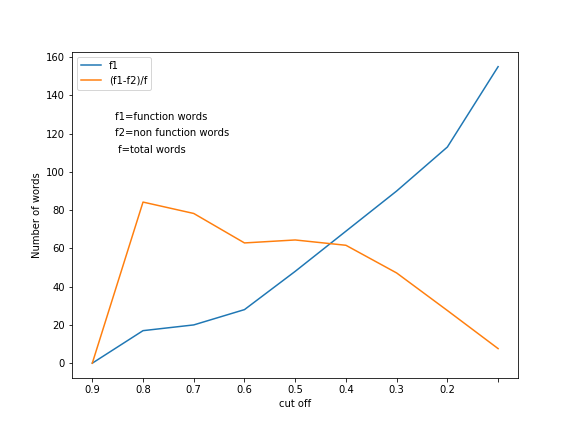
But does this implication always hold?

No. The implication does not hold for all cases. For example if a set of authors write on the same

topic then the topic itself would be a frequent word but not a function word. In spite of this, the above implication holds most of the times. Therefore we could extract the most frequent words across the entire dataset and a large portion of these words would be function words. Thus it would be a good approximation for function words.

Now the problem was how to collect the most frequent words across the entire dataset. We decided to use a threshold value £. If a word appeared in more than £ documents then we would regard it as a frequent word. A value of £=1 means that the word must have appeared in all the documents. Even for a function word there is no guarantee that it would appear in all documents. So a value of £=1 was too harsh. But we could not have selected a very low value of £ such as 0.25. A value as low as this would introduce more number of non-function words in the collected frequent words. To get the optimum value of £ we plotted the graph as shown below

|  |  |  |  |
| --- | --- | --- | --- |
| £ | Total number of words | Function words | Non function words |
| 1 | 0 | 0 | 0 |
| 0.9 | 18 | 17 | 1 |
| 0.8 | 22 | 20 | 2 |
| 0.7 | 34 | 28 | 6 |
| 0.6 | 58 | 48 | 10 |
| 0.5 | 85 | 69 | 16 |
| 0.4 | 122 | 90 | 32 |
| 0.3 | 177 | 113 | 64 |
| 0.2 | 288 | 155 | 133 |



We defined a ration which we termed as useful ratio. This ratio was equal to number of non-function words subtracted from function words and the result divided by the number of total words. Our objective was to find a value of £ such that this ratio is maximum and also the number of function words is significant. The only such value of £ was 0.5. So this means we took those words which have occured in more than half of the documents.

**4.2.4 Part of speech tagging**

In this model each word in the document is replaced by its tag. The tag represents whether the word is a noun, pronoun, conjunction etc. Then we have to calculate the frequency count of each tag and distinguish author on the basis of differing frequency counts.

**4.2.5 Secondary features**

Certain features alone are incapable of identifying authors but when used with other features they may help in improving the accuracy. We have termed these features as secondary features. Secondary features that we have used:

* Number of English words used by author
* Number of words which occur only once
* Vocabulary richness
* Average word length
* Average sentence length
* Length of the document

**4.2.6 Combination of features**

Features have also been used in combination. For example trigrams and POS tags have been combined then used for training. The list of combinations used:

* Bigrams + trigrams
* Function words + trigrams
* POS tags + trigrams
* POS tags + trigrams

**4.3 Text normalisation**

**4.3.1 Term frequency normalisation**

In all the feature models, a raw count of words/n-grams is taken. This brings certain undesirable effects. The raw count is dependent upon document length but an author style should be independent of document length. To remove the effect of document length we divide the feature vector with the length of the document.

**4.3.2 IDF normalisation**

Certain words are used by all authors in almost equal amounts. These words can’t help in differentiating author style. Hence they must be given little weightage. To achieve this we use IDF normalisation which gives weight to a feature depending on how many documents the feature appeared in. Higher the number the documents lower is the weight.

**4.4 Feature normalisation**

Certain machine learning algorithms requires all features to be in the same range. To achieve this we applied min max normalisation.

**4.5 Model training and evaluation**

The dataset was split into two sets – the training set and testing set. For training set we have 12 documents for each author. For testing set we have 2 documents per author. We trained a total of 5 learning models on each feature model.

**4.5.1 Choice of evaluation metric**

Accuracy is the commonly used evaluation metric. There are certain other metrics such as precision and recall. But these metrics are needed if the dataset is highly unbalanced. Our dataset was balanced since each author had equal number of documents. Therefore we decided to keep accuracy as the evaluation metric.

**4.5.2 Approach:**

Our first goal was to develop a learning model for identifying authors. But we also wanted to know how this model would perform with increase in the number of authors. Since we had data for only 18 authors if we have used the entire set of authors we could not have achieved our second goal. Therefore we started by training our model with 5 authors and noted the results. Then we increased the number of authors to 6 and again trained the model. This step repeated for 7, 8 …18 authors. Finally we plotted a graph to see how our model scales with the increase in number of authors.

**How to choose the 5 authors?**

Our first task was to train the model with 5 authors only. But we could not have chosen any 5 random authors. If the 5 chosen authors have very similar writing style then the model would have given poor accuracy but if they have contrasting writing styles then the model would have high accuracy. We can see that the model becomes sensitive to the set of authors. To remove this undesirable effect the only solution was to take the all combinations of 5 authors from a set of 18 authors, train a model on each set of 5 authors and then take the average of all those accuracies. But training 18C5 models is too much time consuming. So we left the idea of training all combinations and took a different approach. We trained our model for 20 different sets of 5 authors and then took the model accuracy as the average of those accuracies. But this accuracy has some error value associated with it since we did not take all the combinations. To get an estimate of the error we took the standard deviation of model accuracies and subtracted this deviation from average accuracy calculated.

This approach was also used when training models for 6, 7, 8 …16 authors. For 17 authors we trained only 18C17 models. For 18 authors only 18C18=1 model was trained.

**4.5.3 Cross validation strategy**

We decided to used k fold cross validation with k=10. This is the most commonly used value of k for k fold cross validation.

**4.5.4 Learning models used**

We used the following models:

* RandomForest Classifier
* Multinomial Naïve Bayes
* Logistic Regression
* SVM with radial basis function kernel
* K nearest neighbours

The choice of these models is not arbitrary. RandomForest Classifer, Naïve Bayes and SVM have proved their effectiveness in the earlier works conducted in this field. Although Logistic Regression and K nearest neighbours have not shown their effectiveness in English texts, that does not make them unsuitable for Hindi texts.

**Chapter 5**

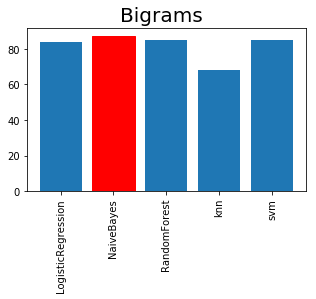
**Results**

**5.1 Bag of Words Approach:**

|  |  |
| --- | --- |
| **Algorithm Used** | **Accuracy** |
| SVM | 82.08 +- 7.72 |
| RandomForest | 89.08 +- 4.51 |
| Knn | 75.5 +- 8.11 |
| LogisticRegression | 86.35 +- 5.63 |
| NaiveBayes | 88.2 +- 7.24 |
|  |  |

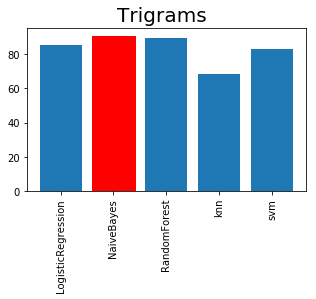
**5.2 Bigrams Approach:**

|  |  |
| --- | --- |
| **Algorithm Used** | **Accuracy** |
| **SVM** | **89. +-4.06** |
| **RandomForest** | **90.08 +- 5.17** |
| **Knn** | **75.67+- 7.28** |
| **LogisticRegression** | **89.0+- 5.04** |
| **NaiveBayes** | **92.9 +- 5.55** |

****

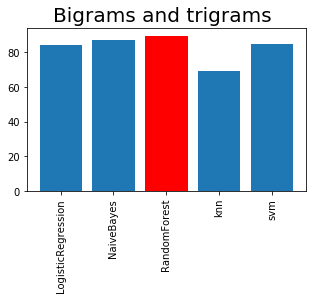
**5.3 Trigrams**

|  |  |
| --- | --- |
| **Algorithm Used** | **Accuracy** |
| **SVM** | **86. 83+-3.82** |
| **RandomForest** | **92.6 +- 3.13** |
| **Knn** | **76.33+- 7.85** |
| **LogisticRegression** | **90.5+- 4.9** |
| **NaiveBayes** | **94.65 +- 3.99** |

****

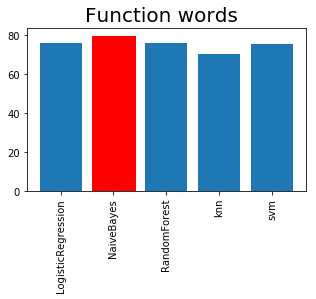
**5.4 Bigrams and Trigrams**

|  |  |
| --- | --- |
| **Algorithm Used** | **Accuracy** |
| **SVM** | **88. 83+-4.4** |
| **RandomForest** | **92.75 +- 3.35** |
| **Knn** | **76.58+- 7.52** |
| **LogisticRegression** | **89.6+- 5.35** |
| **NaiveBayes** | **92.4 +- 5.11** |

****

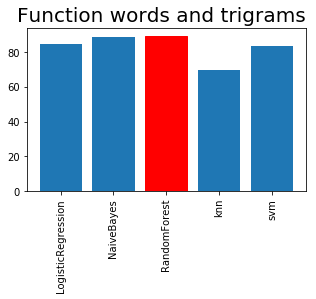
**5.5 Function Words**

|  |  |
| --- | --- |
| **Algorithm Used** | **Accuracy** |
| **SVM** | **82. 67+-7.12** |
| **RandomForest** | **83.17 +- 7.35** |
| **Knn** | **76.42+- 5.98** |
| **LogisticRegression** | **83.9+- 7.72** |
| **NaiveBayes** | **87.0 +- 7.73** |

****

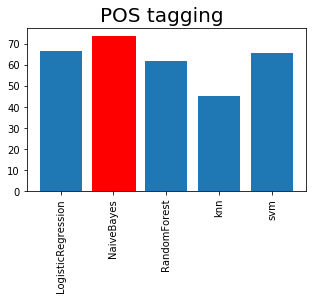
**5.6 Function Words and Trigrams**

|  |  |
| --- | --- |
| **Algorithm Used** | **Accuracy** |
| **SVM** | **88. 17+-4.92** |
| **RandomForest** | **91.83+- 2.64** |
| **Knn** | **77.0+- 7.23** |
| **LogisticRegression** | **88.6+-3.87** |
| **NaiveBayes** | **92.7+- 3.91** |

****

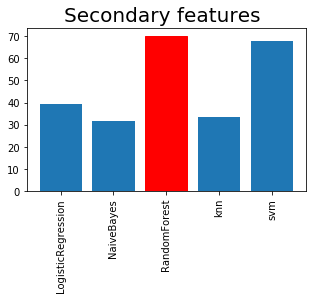
**5.7 Part of Speech tagging**

|  |  |
| --- | --- |
| **Algorithm Used** | **Accuracy** |
| **SVM** | **75.42+-9..94** |
| **RandomForest** | **72.0+- 10.27** |
| **Knn** | **58.08+- 12.97** |
| **LogisticRegression** | **75.75+-9.26** |
| **NaiveBayes** | **81.7+- 8.05** |

****

**5.8 Secondary features**

|  |  |
| --- | --- |
| **Algorithm Used** | **Accuracy** |
| **SVM** | **72.67+-5.14** |
| **RandomForest** | **76.14+- 6.07** |
| **Knn** | **42.45+- 8.99** |
| **LogisticRegression** | **49.75+-10.34** |
| **NaiveBayes** | **39.18+- 7.67** |

****

**5.9 Trigrams and Secondary Features**

**Chapter 6**

**Conclusions and Future work**

**References**

[1] Rohangiz Modaber "Authorship Attribution and Statistical Text Analysis”

[2] Mosteller, F., & Wallace, D.L. (1964). Applied Bayesian and Classical Inference: The Case of the Federalist Papers. Reading (MA): Addison-Wesley..

[3] Mosteller, F., & Wallace, D.L. (1964). Applied Bayesian and Classical Inference: The Case of the Federalist Papers. Reading (MA): Addison-Wesley.

[4] Huang, H., Guo, W., and Zhang, Y. “Detection of copy-move forgery in digital images using SIFT algorithm”, Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008.

[5] S. Alam and D. Ojha, “A Literature study on Image forgery”, International Journal of Advance Research in Computer Science and Management Studies, vol. 2, no. 10, (2014) October

[6] Jian Li, Xiaolong Li, Bin Yang, and Xingming Sun, Senior Member, IEEE’Segmentation-Based Image Copy-Move Forgery Detection Scheme IEEE Trans. Inf. forensics and security, vol. 10, no. 3, March 2015.

[7] S.-J. Ryu, M. Kirchner, M.-J. Lee, and H.-K. Lee, “Rotation invariant localization of duplicated image regions based on Zernike moments,” IEEE Trans. Inf.Forensic Security, vol. 8, no. 8, pp. 1355–1370, Aug. 2013.

[8] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, “An evaluation of popular copy-move forgery detection approaches,” IEEE Trans. Inf. ForensicsSecurity, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[9]. Detecting Multiple Copies of Copy-Move,Forgery Based on SURF K.Kiruthika, S.Devi Mahalakshmi, K.Vijayalakshmi,International Conference on Innovations in Engineering and Technology (ICIET’14)ISSN (Online) : 2319 – 8753.

[10]. A Scheme for Copy-Move Forgery Detection in Digital Images Based on 2D-DWT S.A. Fattah1, M. M. I. Ullah1, M. Ahmed1,Ahmmed2, and C. Shahnaz1,Prime Silicon Technology Inc., Santa Clara, CA, USA 978-1-4799-4132-2/14 ©2014 IEEE .

[11] H.-J. Lin, C.-W. Wang and Y.-T. Kao, “Fast Copy-Move Forgery Detection”, ISSN: 1790-5052 189, vol. 5, no. 5, (2009) May.

[12] S. Lyu and H. Farid, “How realistic is photorealistic?” IEEE Transac- tions on Signal Processing, vol. 53, no. 2, pp. 845–850, 2005.

[13] H. Farid, “Photo fakery and forensics,” Advances in Computers, vol. 77, pp. 1–55, 2009.

[14] J. Fridrich, D. Soukal and J. Lukas, Detection of copy–move forgery in digital images, Proceedings of Digital Forensic Research Workshop, IEEE Computer Society, Cleveland, OH, USA (August 2003), pp. 55–61.

[15] D.G. Lowe (2004). “Distinctive Image Features from Scale-Invariant Key points”. International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110.

[16] S. Bayram, H. Taha Sencar, N. Memon, An efficient and robust method for detecting copy move forgery, in: ICASSP ’09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE Computer Society, Washington, DC, USA, 2009, pp. 1053–1056.

[17] Harpreet Kaur, Jyoti Saxena, and Sukhjinder Singh., (2015), “Simulative Comparision of Copy–Move Forgery Detection Methods for Digital Images” International Conference on Journal of Electronics, Electrical & Computational, vol.4, pp. 62-66.

[18] Toqeer Mahmood., (2015), “A Survey on Block based Copy Move Image Forgery Detection Techniques”, International Conference on Emerging Technologies, pp.1-6.