I.J. Information Technology and Computer Science, **2018, 10, 11-21**
Published Online October 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2018.10.02

Authorship Attribution for Bengali Language Using the Fusion of N-Gram and Naive Bayes Algorithms

D. M. Anisuzzaman
Ahsanullah University of Science and Technology, Department of Computer Science and Engineering, Dhaka, Bangladesh
E-mail: rajon99@gmail.com

Abdus Salam
American International University-Bangladesh, Department of Computer Science, Dhaka, Bangladesh
E-mail: salamt2@aiub.edu

Received: 08 February 2018; Accepted: 12 August 2018; Published: 08 October 2018

***Abstract*—**This research shows the authorship attribution for three Bengali writers using both Naïve Bayes method and a new method proposed by us which performs better than Naïve Bayes for authorship attribution. Though a lot of works exist in the field of authorship attribution for other languages (especially English); the amount of work in this field for Bengali language is very low. For this experiment, we make our own dataset having 107380 words and 21198 unique words. For both methods, we pre-process our dataset to be compatible to work with the method experiments. For our dataset, Naïve Bayes gives an accuracy of 86% while our method gives an accuracy of 95%. The main inspiration behind our method is that every author has a nature to write some adjacent words and some single words repeatedly.

***Index Terms*—**Naive Bayes, n gram, authorship attribution, bengali language, natural language processing.

I. Introduction

The main idea behind authorship attribution is that we can distinguish among texts written by different authors by measuring some textual features. In the general authorship attribution problem, a text of unknown author is assigned to one author from a given set of candidate authors for whom text samples are available. In our experiment the training dataset is the text samples for the corresponding authors and a text of unknown author means the test set. From the predefined sample texts of three different authors we determine which author is more probable to write an unknown authorship sentence. Authorship attribution has a large area of applications: author verification, plagiarism detection, author profiling or characterization, detection of stylistic inconsistencies, forensic linguistics etc. There exists a lot of works for English literature in the field of authorship attribution but the work done in this field for Bengali literature is very few and that is why our research has a special importance in this field. In this paper, we are not only doing authorship attribution for Bengali literature with a traditional (naïve bayes) method but also proposing a new method that can outperform this traditional method for authorship attribution problem.

Here we want to classify a given input to a class that represents a predefined set of authors in Bengali literature. For this we use a fusion of N-Gram and Naïve Bayes algorithm. The main reason behind the fusion is that – every author has a writing pattern and they use some words (both adjacent and single) more than other authors. The dataset includes several lines written by some authors. The authors are the classes.

Using this fusion classifier, a new test string will be classified into a class, in other words it will show the highest probability of the new string to be written by one of the predefined authors. Though there exist a lot of work in the field of authorship attribution it is relatively very small for Bengali language. To our knowledge none have done this work using Naïve Bayes method for Bengali language. Moreover, we are proposing our own method for authorship attribution in Bengali language.

II. Related Works

Automatic authorship detection is done by using integrated syntactic graph (ISG) feature extraction methodology in [1]. The ISG is built with Lexical level, Morphological level, Syntactic level and Semantic level. They use two approaches: Profile-based approach and Instance-based approach for solving both authorship verification and authorship attribution problems. And they propose two methods for the authorship verification problem: Extrinsic approach and Intrinsic approach. Finally, they observed that for automatic authorship detection, the best results are achieved while more features are added to the graph. They achieved an

accuracy of 71% and 71.5% for English essay and English novel respectively.

A novel method for computer-assisted authorship attribution based on character level n-gram author profiles is proposed in [2]. Their approach is based on byte-level n-grams. It is language independent, and the generated author profiles are limited in size. They performed experiments on English, Greek, and Chinese data. For the experiment, the authors have used "The Perl packageText::N grams[Keselj2003]" to produce n-gram tables. They didn't do any preprocessing but simply use byte n-grams by treating texts as byte sequences. The experiment is done on three kinds of data sets: English Data Set, Greek Data Set and Chinese Data Set and for each they obtain highest 100%, 97% and 89% accuracy respectively.

An unsupervised learning approach - a hierarchical Naive Bayes mixture model is used for name disambiguation in author citations in [3]. Authors have used both the web collected datasets and the DBLP datasets and manually labeled the canonical name entities. Their method partitions a collection of citations into clusters. Each cluster containing only citations written by the same author. Three types of citation features have been used: co-author names, paper title words, and journal or proceeding title words. They achieved an accuracy of 63.2% on an average as their best.

Authorship attribution for Arabic is done by using Naïve Bayes classifier in [4]. They have used different event models, namely, simple Naive Bayes (NB) [5], multinomial Naive Bayes (MNB) [6], multi-variant Bernoulli Naive Bayes (MBNB) [7] and multi-variant Poisson Naive Bayes (MPNB) [7]. The experimental results show that multi-variant Poisson Naive Bayes (MBNB) provides the best results with an accuracy of 97.43%.

Hidden Markov Model (HMM) has been used for Bengali authorship identification in [8]. This work tries to identify the trigram pattern and the nominal and verbal chunks of Bengali language. Authors claimed that they solve some basic HMM problems by using forward algorithm, viterbi algorithm and forward-backward algorithm. They choose 15 authors and encoded their literature using UTF-8 encoding in python environment. The system is trained with approximately 50,000 chunks and reported an accuracy of over 90%.

A set of fine-grained stylistic features (unique linguistic styles and writing behaviors of individuals) for the analysis of the text is used to develop two different models for authorship identification in Bengali literature in [9]. The models are: statistical similarity model that consist of three measures and their combination, and machine learning model with Decision Tree, Neural Network and Support Vector Machine. Their experimental results show that SVM outperforms other state-of-the-art methods after 10-fold cross validations. Their SVM model achieved an accuracy of 83.3%.

An end-to-end system for authorship classification for Bengali literature is developed in [10]. It is based on character n-grams which uses a new corpus of 3,000 passages written by three Bengali authors (Rabindranath Tagore, Sarat Chandra Chattopadhyay and Bankim Chandra Chattopadhyay). Their work also includes: feature selection for authorship attribution, feature ranking and analysis, and learning curve to assess the relationship between amount of training data and test accuracy. They achieve state-of-the art results on their dataset.

Some other existing works have used random forest classifier [11], established and modified stylometric features [12] and graph based models [13] for Bengali authorship attribution. In these works they have used their own developed dataset. Authorship attribution for some other languages like: Arabic has also been done by using extended version of the probabilistic context free grammar language [15] and [16]. Authorship attribution has also been done for English literature by using author based rank vector coordinates (ARVC) with an accuracy of 96.43% in [14].

From the above review, it can be observed that very little work has been done in authorship attribution in Bengali language especially using Naïve Bayes classifier and N-gram algorithm. Again, Bigram count in Naïve Bayes classifier has not been used in any research in this field. In this paper, we have shown how Naïve Bayes with bigram count can be used for authorship attribution. Furthermore, we have also proposed a fusion of N-gram and Naïve Bayes algorithms for the same purpose which shows better performance.

III. Proposed Method

We use both Naïve Bayes classifier and my method to predict the author. The fusion method that we use to find the author is given below:

Step 1:

First, we collect some writing of renowned authors of Bengali literature and categorize their writing as different classes. For this work, we have to collect raw writings of our three selected authors: 'Humayun Ahmed', 'Rabindranath Tagore' and 'Shamsur Rahman' and then have to pre-process their writing to an acceptable format that can be used to our proposed algorithm. The pre-processing step involves: removing punctuations; removing numbers and removing unnecessary whitespaces. The testing data are carefully chosen separately from the training data so that the method cannot be biased.

Step 2:

Then we convert the Bengali language into English phonetics. This work has been performed in python platform using Unicode converter. We convert both the training and testing data by using this Unicode converter. For our method, we collect the training data in three different files named: "H.txt", "R.txt" and "S.txt" that represents 'Humayun Ahmed', 'Rabindranath Tagore' and 'Shamsur Rahman' respectively.

Step 3:

After that we find the bigram and unigram count table for each class. We do this work by using the words tokenize method in python platform. The unigram and bigram method are then saved in separate files for each input files. For "H.txt" the unigram and bigram count are saved in "uni1.txt" and "bi1.txt" respectively and that will be our new corpus to work with. That is, we look into this unigram file and bigram file for our calculation.

Step 4:

Then we classify the given input into one class according to the class's probability (highest) using the following fusion equation:

$$c_{NB} = \underset{c \varepsilon C}{\operatorname{argmax}}\ P(c) \prod P(w/c)$$

Where,

$$P(w/c) = \frac{\text{count}(w_{n-1}w_n, c)+1}{count(w_{n-1}, c)+V} + \frac{\text{count}(w_{n-1}, c)+1}{\sum count(w,c)+V} + \frac{\text{count}(w_n, c)+1}{\sum count(w,c)+V}$$

$$= \frac{\text{count}(w_{n-1}w_n, c) + \text{count}(w_{n-1}, c) + \text{count}(w_n, c)+1}{\sum count(w,c)+V}$$

Here,

P(c) = probability of the classes.

P(w/c) = probability of the adjacent words for each classes

count($w_{n-1}w_n$, c)= bigram count of the test input from the bigram file generated in the previous step.

count(w_{n-1}, c)= unigram count of the first bigram word of the test input from the unigram file generated in the previous step.

count(w_n, c)= unigram count of the second bigram word of the test input from the unigram file generated in the previous step.

$\sum$ *count(w, c)* = The total number of words in the current corpus.

V = Vocabulary of the current corpus.

Here, we have used the N-gram formula with Laplace smoothing in the first part of our equation ((count($w_{n-1}w_n$,c)+1) ÷ (count(w_{n-1},c)+V)) for the bigram count; which has helped us to understand the adjacent words used by a specific author. The rest of the equation contains Naïve Bayes formula with Laplace smoothing ((count(w_{n-1},c)+1) ÷ ($\sum$count(w,c)+V) and (count(w_n,c)+1) ÷ ($\sum$count(w,c)+V)); which has helped us to understand the probability of using each words of the bigram separately by a specific author.

In the combined equation we have ignored the count(w_{n-1}, c) in the denominator of our calculation as it will be very small in comparison with the summation of $\sum$ count(w, c) and V.

Previous works have used Naïve Bayes or modified Naïve Bayes formulas and N-gram formulas separately for authorship detection. With this equation our work shows a new way for authorship attribution by combining N-gram with Naïve Bayes algorithm.

Step 5:

Finally, we compare the probabilities that we get in our fourth step among our three selected authors and show the output as: which class does the given input belongs. In other words, among a pre-defined set of authors who has the highest probability to write the given sentence.

IV. Experiment

A. Dataset

Step 1:

For this experiment, we make our own dataset. We collect the Bengali literature of three different authors named: 'Humayun Ahmed', 'Rabindranath Tagore' and 'Shamsur Rahman'. Then we do some transformation on our dataset. The transformation process includes: remove punctuations; remove numbers and remove unnecessary whitespace. An example of data transformation is given in Table 1.

Table 1. Dataset Transformation Example

Original Data	Pre-processed (transformed) Data
সুরাইয়া অবাক হয়ে তার ছেলের দিকে তাকিয়ে আছে। ছেলের নাম ইমন। বয়স পাঁচ বছর তিনমাস। মাথা ভর্তি কোকড়ানো চুল। লম্বাটে ধরণের মুখ। মাঝে মাঝে সেই মুখ কোন এক বিচিত্র কারণে গোলগাল দেখায়, আজ দেখাচ্ছে। ইমন তার মায়ের বিস্মিত দৃষ্টির কারণ ধরতে পারছে না। সে ভুরু কুঁচকে মায়ের দিকে তাকিয়ে আছে। ভুরু কুঁচকানোর এই বদঅভ্যাস সে পেয়েছে তার বাবার কাছ থেকে।	সুরাইয়া অবাক হয়ে তার ছেলের দিকে তাকিয়ে আছে ছেলের নাম ইমন বয়স পাঁচ বছর তিনমাস মাথা ভর্তি কোকড়ানো চুল লম্বাটে ধরণের মুখ মাঝে মাঝে সেই মুখ কোন এক বিচিত্র কারণে গোলগাল দেখায় আজ দেখাচ্ছে ইমন তার মায়ের বিস্মিত দৃষ্টির কারণ ধরতে পারছে না সে ভুরু কুঁচকে মায়ের দিকে তাকিয়ে আছে ভুরু কুঁচকানোর এই বদঅভ্যাস সে পেয়েছে তার বাবার কাছ থেকে

From Table 1, we can see that the stop notation of Bengali language '।' has been removed after the words: 'আছে।', 'ইমন।', 'তিনমাস।', 'চুল।', 'মুখ।', 'দেখাচ্ছে।', 'না।', 'আছে।' and 'থেকে।'. The gap between the two words of 'আছে' and 'ছেলের' is also removed. The other punctuations like: ',', ':', ';' are also removed from our original dataset to the pre- processed dataset. This transformation process is done to make this dataset fit for our next step.

We mainly collect our data from Bengali Unicode library. They have a huge collection of Bengali writing of renowned authors. But the problem is that they are in raw format. That means they include all the punctuations, numbers and other unnecessary things that are not important in our experiment. So we manually remove those things to make it compatible for our experiment.

Step 2:

Then we convert these Bangla words to English phonetics and use it for our experiment. We do the conversion using Unicode converter in python platform. We give the Bengali file input in our system and get the phonetics to use it in our experiment. An example of data transformation is shown in Table 2.

We save this converted dataset in one file for the Naïve Bayes experiment and in three separate files named: 'H.txt', 'R.txt' and 'S.txt' which represents 'Humayun Ahmed', 'Rabindranath Tagore' and 'Shamsur Rahman' respectively for our experiment.

Table 2. Dataset Conversion Example

Author Name ⬇	Pre-processed (transformed) Data	Phonetically (English) converted Data
Humayun Ahmed	সুরাইয়া অবাক হয়ে তার ছেলের দিকে তাকিয়ে আছে ছেলের নাম ইমন বয়স পাঁচ বছর তিনমাস মাথা ভর্তি কোকড়ানো চুল লম্বাটে ধরণের মুখ মাঝে মাঝে সেই মুখ কোন এক বিচিত্র কারণে গোলগাল দেখায় আজ দেখাচ্ছে ইমন তার মায়ের বিস্মিত দৃষ্টির কারণ ধরতে পারছে না সে ভুরু কুঁচকে মায়ের দিকে তাকিয়ে আছে ভুরু কুঁচকানোর এই বদঅভ্যাস সে পেয়েছে তার বাবার কাছ থেকে	suraiya obak hoye tar cheler dike takiye ache cheler nam imon boyos pacboC bochor tinomas matha vorti kOkoRanO Cul lombaTe dhoroNer mukh majhe majhe sei mukh kOn Ek biCitr karoNe gOlogal dekhay aj dekhaCche imon tar mayer bismit drriShTir karoN dhorote paroche na se vuru kucboCoke mayer dike takiye ache vuru kucboCokanOr Ei bodoovzas se peyeche tar babar kach theke
Rabindranath Tagore	রমেশ এবার আইনপরীক্ষায় যে পাস হইবেসে সম্বন্ধে কাহারো কোনো সন্দেহ ছিল না বিশ্ববিদ্যালয়ের সরস্বতী বরাবর তাঁহার স্বর্ণপদ্মের পাপড়ি খসাইয়া রমেশকে মেডেল দিয়া আসিয়াছেন স্কলারশিপও কখনো ফাঁক যায় নাইপরীক্ষা শেষ করিয়া এখন তাহার বাড়ি যাইবার কথা কিন্তু এখনো তাহার তোরঙ্গ সাজাইবার কোনো উৎসাহ দেখা যায় নাইপিতা শীঘ্র বাড়ি আসিবার জন্য পত্র লিখিয়াছেন রমেশ উত্তরে লিখিয়াছে	romesh Ebar ainoporIkShay ze pas hoibese sombondhe kaharO kOnO sondeh chil na bishbobidzaloyer sorosbotI borabor tacbohar sborNopodmer papoRi khosaiya romeshoke meDel diya asiyachen skolaroshipoO kokhonO facbok zay naiporIkSha sheSh koriya Ekhon tahar baRi zaibar kotha kintu EkhonO tahar tOroNgg sajaibar kOnO uTHosah dekha zay naipita
	পরীক্ষার ফল বাহির হইলেই সে বাড়ি যাইবে	shIghr baRi asibar jonz potr likhiyachen romesh uttore likhiyache porIkShar fol bahir hoilei se baRi zaibe
Shamsur Rahman	ফিরে যাবো কেন ফিরে যাবো বারবার জানি সিংহদ্বার পেরুলেই পেয়ে যাবো আকাঙ্ক্ষিত সব উপচার যার জন্যে ভীষণের স্তব করেছি সকালসন্ধ্যা পেরিয়েছি ঝড়মত্ত নদী কতো সিঁড়ি রক্তাপ্লুত বারংবার নেমেছি খনিতে আজো ফিরি পথে পথে কেইনের মতো ফিরে যাবো প্রহরীর রক্তচক্ষু দেখে ফিরে যাবো তুমি তবুও বধির হয়ে থাকবে সর্বক্ষণ ডাকবে না সেখানে যেখানে আমার ব্যাকুল পদচ্ছাপ পড়েছিলো স্বপ্নে	fire zabO ken fire zabO barobar jani singohodbar perulei peye zabO akaNgkShit sob upoCar zar jonze vIShoNer stob korechi sokalosondhza periyechi jhoRomott nodI kotO sicboRi roktaplut barongobar nemechi khonite ajO firi pothe pothe keiner motO fire zabO prohorIr roktoCokShu dekhe fire zabO tumi tobuO bodhir hoye thakobe sorbokShoN Dakobe na sekhane zekhane amar bzakul podoCchap poRechilO sbopne

Step 3:

Then we prepare our test set using the same process described in step 1 and 2. An example of test case is shown in Table 3.

Table 3. Test-case Processing Example

Author Name ⬇	Original Data	Pre-processed (transformed) Data	Phonetically (English) converted Data
Humayun Ahmed	ফরিদা তৃপ্তির নিঃশ্বাস ফেলে বললেন- "তোরতো দেখি পান খাওয়া অভ্যাস হয়ে যাচ্ছে।"	ফরিদা তৃপ্তির নিঃশ্বাস ফেলে বললেন তোরতো দেখি পান খাওয়া অভ্যাস হয়ে যাচ্ছে	forida trriptir ni:oshbas fele bololen tOrotO dekhi pan khaOya ovzas hoye zaCche
Rabindranath Tagore	রামমোহন মাল যখন অন্তঃপুরে আসিয়া বিভাকে প্রণাম করিয়া কহিল – "মা তোমায় একবার দেখিতে আসিলাম।"	রামমোহন মাল যখন অন্তঃপুরে আসিয়া বিভাকে প্রণাম করিয়া কহিল মা তোমায় একবার দেখিতে আসিলাম	ramomOhon mal zokhon onto:opure asiya bivake proNam koriya kohil ma tOmay Ekobar dekhite asilam
Shamsur Rahman	কে আমাকে নিয়ে যাচ্ছে অজানা পথের ধুলোবালি চোখে-মুখে ছড়িয়ে; সন্ধ্যায় কেন যাচ্ছি	কে আমাকে নিয়ে যাচ্ছে অজানা পথের ধুলোবালি চোখে-মুখে ছড়িয়ে সন্ধ্যায় কেন যাচ্ছি	ke amake niye zaCche ojana pother dhulObali COkhe-mukhe choRiye sondhzay ken zaCchi

We select the test-cases carefully and separately from our training cases. So our system does not show any biased result. A summary of both our experimental data and test case data is shown in Table 4. We take 100 inputs for each author which means a total of 300 inputs; where an input means a chunk of words written by a single author.

Table 4. Dataset Summary

Training set for Experiment with Naïve Bayes method			
No. of words	107380		
Vocabulary	21198		
Training set for Experiment with our method			
	Humayun Ahmed	Rabindranath Tagore	Shamsur Rahman
No. of words	37536	36395	33449
Vocabulary	6355	10493	9678
Test set for both Experiments			
	Humayun Ahmed	Rabindranath Tagore	Shamsur Rahman
No. of test sentences	100	100	100

B. Experiment with Naïve Bayes method

Step 1:

For Naïve Bayes classifier we manually label dataset in three different classes. For a chunk of words written by 'Humayun Ahmed' we label it as '1' and for 'Rabindranath Tagore' and 'Shamsur Rahman' we label them as '2' and '3' respectively. There are total 879 labels which contain total 107380 words and 21198 unique words. The prior probability of class '1', '2' and '3' are $389/879$, $257/879$ and $233/879$ respectively. The sample dataset making process is shown in Table 5.

Table 5. Making Dataset for Naïve Bayes Method Example

Dataset.csv
suraiya obak hoye tar cheler dike takiye ache cheler nam imon boyos pacboC bochor tinomas matha vorti kOkoRanO Cul lombaTe dhoroNer mukh majhe majhe sei mukh kOn Ek biCitr karoNe gOlogal dekhay aj dekhaCche imon tar mayer bismit drriShTir karoN dhorote paroche na se vuru kucboCoke mayer dike takiye ache vuru kucboCokanOr Ei bodoovzas se peyeche tar babar kach theke**+1** romesh Ebar ainoporIkShay ze pas hoibese sombondhe kaharO kOnO sondeh chil na bishbobidzaloyer sorosbotI borabor tacbohar sborNopodmer papoRi khosaiya romeshoke meDel diya asiyachen skolaroshipoO kokhonO facbok zay naiporIkSha sheSh koriya Ekhon tahar baRi zaibar kotha kintu EkhonO tahar tOroNgg sajaibar kOnO uTHosah dekha zay naipita shIghr baRi asibar jonz potr likhiyachen romesh uttore likhiyache porIkShar fol bahir hoilei se baRi zaibe**+2** fire zabO ken fire zabO barobar jani singohodbar perulei peye zabO akaNgkShit sob upoCar zar jonze vIShoNer stob korechi sokalosondhza periyechi jhoRomott nodI kotO sicboRi roktaplut barongobar nemechi khonite ajO firi pothe pothe keiner motO fire zabO prohorIr roktoCokShu dekhe fire zabO tumi tobuO bodhir hoye thakobe sorbokShoN Dakobe na sekhane zekhane amar bzakul podoCchap poRechilO sbopne**+3**

Step 2:

Then we test the system with 100 sentences from each author which means with total 300 sentences. For each sentence, we find the probability of three classes using Naïve Bayes method and classify it into the class with highest probability. We do this experiment using Naïve Bayes classifier in python platform. A sample output is shown in Table 6. The result of this system is described in the 'Result Analysis' section.

Table 6. Sample Output of Naïve Bayes Method

naiveBayes.py
Sample #1 Give a sentence: forida trriptir ni:oshbas fele bololen tOrotO dekhi pan khaOya ovzas hoye zaCche The predicted author is : Humayun Ahmed
Sample #2 Give a sentence: ramomOhon mal zokhon onto:opure asiya bivake proNam koriya kohil ma tOmay Ekobar dekhite asilam The predicted author is : Rabindranath Tagore
Sample #3 Give a sentence: ke amake niye zaCche ojana pother dhulObali COkhe-mukhe choRiye sondhzay ken zaCchi The predicted author is : Shamsur Rahman

As shown in Table 6, we take inputs as a chunk of words written by a particular author to keep track of the output; that is which author has the highest probability to write this chunk of words.

Table 7. Making Dataset for Our Method Example

H.txt	R.txt	S.txt
suraiya obak hoye tar cheler dike takiye ache cheler nam imon boyos pacboC bochor tinomas matha vorti kOkoRanO Cul lombaTe dhoroNer mukh majhe majhe sei mukh kOn Ek biCitr karoNe gOlogal dekhay aj dekhaCche imon tar mayer bismit drriShTir karoN dhorote paroche na se vuru kucboCoke mayer dike takiye ache vuru kucboCokanOr Ei bodoovzas se peyeche tar babar kach theke	romesh Ebar ainoporIkShay ze pas hoibese sombondhe kaharO kOnO sondeh chil na bishbobidzaloyer sorosbotI borabor tacbohar sborNopodmer papoRi khosaiya romeshoke meDel diya asiyachen skolaroshipoO kokhonO facbok zay naiporIkSha sheSh koriya Ekhon tahar baRi zaibar kotha kintu EkhonO tahar tOroNgg sajaibar kOnO uTHosah dekha zay naipita shIghr baRi asibar jonz potr likhiyachen romesh uttore likhiyache porIkShar fol bahir hoilei se baRi zaibe	fire zabO ken fire zabO barobar jani singohodbar perulei peye zabO akaNgkShit sob upoCar zar jonze vIShoNer stob korechi sokalosondhza periyechi jhoRomott nodI kotO sicboRi roktaplut barongobar nemechi khonite ajO firi pothe pothe keiner motO fire zabO prohorIr roktoCokShu dekhe fire zabO tumi tobuO bodhir hoye thakobe sorbokShoN Dakobe na sekhane zekhane amar bzakul podoCchap poRechilO sbopne

C. Experiment with our method

Step 1:

For our method, we take three text files named ‘H.txt’, ‘R.txt’ and ‘S.txt’ for ‘Humayun Ahmed’, ‘Rabindranath Tagore’ and ‘Shamsur Rahman’ respectively. Each text file contains the writings of the corresponding authors. ‘H.txt’, ‘R.txt’ and ‘S.txt’ contain total 37536, 36395 and 33449 words and 6355, 10493 and 9678 unique words respectively. The sample dataset for our method is shown in Table 7.

Step 2:

Then we find the unigram and bigram count for each text file and save the values to the respected files. For example, we save unigram counts and bigram counts of ‘H.txt’ to ‘uni1.txt’ and ‘bi1.txt’ respectively. Some sample of this process is shown in Table 8, Table 9 and Table 10. These tables show the unigram and bigram sample count of ‘Humayun Ahmed’, ‘Rabindranath Tagore’ and ‘Shamsur Rahman’ respectively.

Table 8. Sample Uni-Gram and Bi-Gram Count of Humayun Ahmed Corpus

uni1.txt
{('tar',): 3, ('cheler',): 2, ('dike',): 2, ('takiye',): 2, ('ache',): 2, ('imon',): 2, ('mukh',): 2, ('majhe',): 2, ('mayer',): 2, ('se',): 2, ('vuru',): 2, ('suraiya',): 1, ('obak',): 1, ('hoye',): 1, ('nam',): 1, ('boyos',): 1, ('pacboC',): 1, ('bochor',): 1, ('tinomas',): 1, ('matha',): 1, ('vorti',): 1, ('kOkoRanO',): 1, ('Cul',): 1, ('lombaTe',): 1, ('dhoroNer',): 1, ('sei',): 1, ('kOn',): 1, ('Ek',): 1, ('biCitr',): 1, ('karoNe',): 1, ('gOlogal',): 1, ('dekhay',): 1, ('aj',): 1, ('dekhaCche',): 1, ('bismit',): 1, ('drriShTir',): 1, ('karoN',): 1, ('dhorote',): 1, ('paroche',): 1, ('na',): 1, ('kucboCoke',): 1, ('kucboCokanOr',): 1, ('Ei',): 1, ('bodoovzas',): 1, ('peyeche',): 1, ('babar',): 1, ('kach',): 1, ('theke',): 1}
bi1.txt
{('dike', 'takiye'): 2, ('takiye', 'ache'): 2, ('suraiya', 'obak'): 1, ('obak', 'hoye'): 1, ('hoye', 'tar'): 1, ('tar', 'cheler'): 1, ('cheler', 'dike'): 1, ('ache', 'cheler'): 1, ('cheler', 'nam'): 1, ('nam', 'imon'): 1, ('imon', 'boyos'): 1, ('boyos', 'pacboC'): 1, ('pacboC', 'bochor'): 1, ('bochor', 'tinomas'): 1, ('tinomas', 'matha'): 1, ('matha', 'vorti'): 1, ('vorti', 'kOkoRanO'): 1, ('kOkoRanO', 'Cul'): 1, ('Cul', 'lombaTe'): 1, ('lombaTe', 'dhoroNer'): 1, ('dhoroNer', 'mukh'): 1, ('mukh', 'majhe'): 1, ('majhe', 'majhe'): 1, ('majhe', 'sei'): 1, ('sei', 'mukh'): 1, ('mukh', 'kOn'): 1, ('kOn', 'Ek'): 1, ('Ek', 'biCitr'): 1, ('biCitr', 'karoNe'): 1, ('karoNe', 'gOlogal'): 1, ('gOlogal', 'dekhay'): 1, ('dekhay', 'aj'): 1, ('aj', 'dekhaCche'): 1, ('dekhaCche', 'imon'): 1, ('imon', 'tar'): 1, ('tar', 'mayer'): 1, ('mayer', 'bismit'): 1, ('bismit', 'drriShTir'): 1, ('drriShTir', 'karoN'): 1, ('karoN', 'dhorote'): 1, ('dhorote', 'paroche'): 1, ('paroche', 'na'): 1, ('na', 'se'): 1, ('se', 'vuru'): 1, ('vuru', 'kucboCoke'): 1, ('kucboCoke', 'mayer'): 1, ('mayer', 'dike'): 1, ('ache', 'vuru'): 1, ('vuru', 'kucboCokanOr'): 1, ('kucboCokanOr', 'Ei'): 1, ('Ei', 'bodoovzas'): 1, ('bodoovzas', 'se'): 1, ('se', 'peyeche'): 1, ('peyeche', 'tar'): 1, ('tar', 'babar'): 1, ('babar', 'kach'): 1, ('kach', 'theke'): 1}

Table 9. Sample Uni-Gram and Bi-Gram Count of Rabindranath Tagore Corpus

uni2.txt
{('baRi',): 3, ('romesh',): 2, ('kOnO',): 2, ('zay',): 2, ('tahar',): 2, ('Ebar',): 1, ('ainoporIkShay',): 1, ('ze',): 1, ('pas',): 1, ('hoibese',): 1, ('sombondhe',): 1, ('kaharO',): 1, ('sondeh',): 1, ('chil',): 1, ('na',): 1, ('bishbobidzaloyer',): 1, ('sorosbotI',): 1, ('borabor',): 1, ('tacbohar',): 1, ('sborNopodmer',): 1, ('papoRi',): 1, ('khosaiya',): 1, ('romeshoke',): 1, ('meDel',): 1, ('diya',): 1, ('asiyachen',): 1, ('skolaroshipoO',): 1, ('kokhonO',): 1, ('facbok',): 1, ('naiporIkSha',): 1, ('sheSh',): 1, ('koriya',): 1, ('Ekhon',): 1, ('zaibar',): 1, ('kotha',): 1, ('kintu',): 1, ('EkhonO',): 1, ('tOroNgg',): 1, ('sajaibar',): 1, ('uTHosah',): 1, ('dekha',): 1, ('naipita',): 1, ('shIghr',): 1, ('asibar',): 1, ('jonz',): 1, ('potr',): 1, ('likhiyachen',): 1, ('uttore',): 1, ('likhiyache',): 1, ('porIkShar',): 1, ('fol',): 1, ('bahir',): 1, ('hoilei',): 1, ('se',): 1, ('zaibe',): 1})
bi2.txt
{('romesh', 'Ebar'): 1, ('Ebar', 'ainoporIkShay'): 1, ('ainoporIkShay', 'ze'): 1, ('ze', 'pas'): 1, ('pas', 'hoibese'): 1, ('hoibese', 'sombondhe'): 1, ('sombondhe', 'kaharO'): 1, ('kaharO', 'kOnO'): 1, ('kOnO', 'sondeh'): 1, ('sondeh', 'chil'): 1, ('chil', 'na'): 1, ('na', 'bishbobidzaloyer'): 1, ('bishbobidzaloyer', 'sorosbotI'): 1, ('sorosbotI', 'borabor'): 1, ('borabor', 'tacbohar'): 1, ('tacbohar', 'sborNopodmer'): 1, ('sborNopodmer', 'papoRi'): 1, ('papoRi', 'khosaiya'): 1, ('khosaiya', 'romeshoke'): 1, ('romeshoke', 'meDel'): 1, ('meDel', 'diya'): 1, ('diya', 'asiyachen'): 1, ('asiyachen', 'skolaroshipoO'): 1, ('skolaroshipoO', 'kokhonO'): 1, ('kokhonO', 'facbok'): 1, ('facbok', 'zay'): 1, ('zay', 'naiporIkSha'): 1, ('naiporIkSha', 'sheSh'): 1, ('sheSh', 'koriya'): 1, ('koriya', 'Ekhon'): 1, ('Ekhon', 'tahar'): 1, ('tahar', 'baRi'): 1, ('baRi', 'zaibar'): 1, ('zaibar', 'kotha'): 1, ('kotha', 'kintu'): 1, ('kintu', 'EkhonO'): 1, ('EkhonO', 'tahar'): 1, ('tahar', 'tOroNgg'): 1, ('tOroNgg', 'sajaibar'): 1, ('sajaibar', 'kOnO'): 1, ('kOnO', 'uTHosah'): 1, ('uTHosah', 'dekha'): 1, ('dekha', 'zay'): 1, ('zay', 'naipita'): 1, ('naipita', 'shIghr'): 1, ('shIghr', 'baRi'): 1, ('baRi', 'asibar'): 1, ('asibar', 'jonz'): 1, ('jonz', 'potr'): 1, ('potr', 'likhiyachen'): 1, ('likhiyachen', 'romesh'): 1, ('romesh', 'uttore'): 1, ('uttore', 'likhiyache'): 1, ('likhiyache', 'porIkShar'): 1, ('porIkShar', 'fol'): 1, ('fol', 'bahir'): 1, ('bahir', 'hoilei'): 1, ('hoilei', 'se'): 1, ('se', 'baRi'): 1, ('baRi', 'zaibe'): 1}

Step 3:

Then we test the system with 50 sentences from each author which means with total 150 sentences. For each sentence, we find the probability of three classes using our method and classify it into the class with highest probability. For our calculation, we consider each bigram of a given sentence and fetch the bigram and unigram values from the corresponding text files. We do this experiment using our proposed classifier in python platform. A sample output is shown in Table 11. The result of this system is described in the ‘Result Analysis’ section.

Table 10. Sample Uni-Gram and Bi-Gram Count of Shamsur Rahman Corpus

uni3.txt
{('zabO',): 5, ('fire',): 4, ('pothe',): 2, ('ken',): 1, ('barobar',): 1, ('jani',): 1, ('singohodbar',): 1, ('perulei',): 1, ('peye',): 1, ('akaNgkShit',): 1, ('sob',): 1, ('upoCar',): 1, ('zar',): 1, ('jonze',): 1, ('vIShoNer',): 1, ('stob',): 1, ('korechi',): 1, ('sokalosondhza',): 1, ('periyechi',): 1, ('jhoRomott',): 1, ('nodI',): 1, ('kotO',): 1, ('sicboRi',): 1, ('roktaplut',): 1, ('barongobar',): 1, ('nemechi',): 1, ('khonite',): 1, ('ajO',): 1, ('firi',): 1, ('keiner',): 1, ('motO',): 1, ('prohorIr',): 1, ('roktoCokShu',): 1, ('dekhe',): 1, ('tumi',): 1, ('tobuO',): 1, ('bodhir',): 1, ('hoye',): 1, ('thakobe',): 1, ('sorbokShoN',): 1, ('Dakobe',): 1, ('na',): 1, ('sekhane',): 1, ('zekhane',): 1, ('amar',): 1, ('bzakul',): 1, ('podoCchap',): 1, ('poRechilO',): 1, ('sbopne',): 1}
bi3.txt
{('fire', 'zabO'): 4, ('zabO', 'ken'): 1, ('ken', 'fire'): 1, ('zabO', 'barobar'): 1, ('barobar', 'jani'): 1, ('jani', 'singohodbar'): 1, ('singohodbar', 'perulei'): 1, ('perulei', 'peye'): 1, ('peye', 'zabO'): 1, ('zabO', 'akaNgkShit'): 1, ('akaNgkShit', 'sob'): 1, ('sob', 'upoCar'): 1, ('upoCar', 'zar'): 1, ('zar', 'jonze'): 1, ('jonze', 'vIShoNer'): 1, ('vIShoNer', 'stob'): 1, ('stob', 'korechi'): 1, ('korechi', 'sokalosondhza'): 1, ('sokalosondhza', 'periyechi'): 1, ('periyechi', 'jhoRomott'): 1, ('jhoRomott', 'nodI'): 1, ('nodI', 'kotO'): 1, ('kotO', 'sicboRi'): 1, ('sicboRi', 'roktaplut'): 1, ('roktaplut', 'barongobar'): 1, ('barongobar', 'nemechi'): 1, ('nemechi', 'khonite'): 1, ('khonite', 'ajO'): 1, ('ajO', 'firi'): 1, ('firi', 'pothe'): 1, ('pothe', 'pothe'): 1, ('pothe', 'keiner'): 1, ('keiner', 'motO'): 1, ('motO', 'fire'): 1, ('zabO', 'prohorIr'): 1, ('prohorIr', 'roktoCokShu'): 1, ('roktoCokShu', 'dekhe'): 1, ('dekhe', 'fire'): 1, ('zabO', 'tumi'): 1, ('tumi', 'tobuO'): 1, ('tobuO', 'bodhir'): 1, ('bodhir', 'hoye'): 1, ('hoye', 'thakobe'): 1, ('thakobe', 'sorbokShoN'): 1, ('sorbokShoN', 'Dakobe'): 1, ('Dakobe', 'na'): 1, ('na', 'sekhane'): 1, ('sekhane', 'zekhane'): 1, ('zekhane', 'amar'): 1, ('amar', 'bzakul'): 1, ('bzakul', 'podoCchap'): 1, ('podoCchap', 'poRechilO'): 1, ('poRechilO', 'sbopne'): 1}

Table 11. Sample Output of Our Method

ourmethod.py
Sample #1 Give a sentence: forida trriptir ni:oshbas fele bololen tOrotO dekhi pan khaOya ovzas hoye zaCche Humayun: 3.89094381752772e-39 Shamsur: 5.814247275591075e-42 Rabindranath: 5.969039931634948e-45 The predicted author is : Humayun Ahmed
Sample #2 Give a sentence: ramomOhon mal zokhon onto:opure asiya bivake proNam koriya kohil ma tOmay Ekobar dekhite asilam Humayun: 3.9107711901223146e-55 Shamsur: 2.020692576526584e-58 Rabindranath: 7.859960302963459e-44 The predicted author is : Rabindranath Tagore
Sample #3 Give a sentence: ke amake niye zaCche ojana pother dhulObali COkhe-mukhe choRiye sondhzay ken zaCchi Humayun: 1.7828944734605444e-36 Shamsur: 6.038198417702306e-35 Rabindranath: 1.6214281077478127e-39 The predicted author is : Shamsur Rahman

For a given sentence: “ke amake niye zaCche ojana pother dhulObali COkhe-mukhe choRiye sondhzay ken zaCchi”; we first separate it to bigrams like: {('ke', 'amake'), ('amake', 'niye') …… ('ken', 'zaCchi') }. Then for each bigram we search in the ‘uni1.txt’, ‘bi1.txt’, ‘uni2.txt’, ‘bi2.txt’, ‘uni3.txt’ and ‘bi3.txt’.

Here for ('ke', 'amake') we look at the bigram tables – ‘bi1.txt’, ‘bi2.txt’ and ‘bi3.txt’ and find the *count*($w_{n-1}w_n$, *c)* for ‘Humayun Ahmed’, ‘Rabindranath Tagore’ and ‘Shamsur Rahman’ respectively. And for ('ke') we look at the unigram tables – ‘uni1.txt’, ‘uni2.txt’ and ‘uni3.txt’ and find the *count*(w_{n-1}, *c)* for ‘Humayun Ahmed’, ‘Rabindranath Tagore’ and ‘Shamsur Rahman’ respectively. We apply the same process for ('amake').

Then we go for the next bigram of the input that is ('amake', 'niye') and go through the same process. After going through all the bigrams of the input we multiply them with prior probability and find our final probability for each author.

From this probability calculation using our method we chose the author who has the highest probability as our guessed author.

V. Result Analysis

Automatic The result and evaluation for Naïve Bayes method and our method is shown in Table 12 and Table 13 respectively. In this table, ‘H’, ‘R’ and ‘S’ represents ‘Humayun Ahmed’, ‘Rabindranath Tagore’ and ‘Shamsur Rahman’ respectively.

From the result and evaluation analysis, we can see that the accuracy of Naïve Bayes method is 86% where our method gives an accuracy of 95%. So for this dataset, our method achieves an accuracy of 9% more than Naïve Bayes method.

The macroaverage precision and recall of Naïve Bayes method is 85% and 86% respectively and the macroaverage precision and recall of our method is 95% and 94% respectively. As high precision means that the classifier is returning accurate results and high recall means that the classifier is returning a majority of all positive results; for all the evaluation metrics our method works better than Naïve Bayes method.

Table 12. Result and Evaluation for Naïve Bayes Method

	System Output (Prediction)				
		H	R	S	
gold labels	H	80	8	12	Recall 'H' $= \frac{80}{80+8+12} = \frac{80}{100} = 0.80$
	R	6	90	4	Recall 'R' $= \frac{90}{6+90+4} = \frac{90}{100} = 0.90$
	S	9	4	87	Recall 'S' $= \frac{87}{9+4+87} = \frac{87}{100} = 0.87$
		Precision 'H' $= \frac{80}{80+6+9} = \frac{80}{95} = 0.842$	Precision 'R' $= \frac{90}{8+90+4} = \frac{90}{102} = 0.882$	Precision 'S' $= \frac{87}{12+4+87} = \frac{87}{103} = 0.845$	Accuracy $= \frac{80+90+87}{300} = \frac{257}{300} =$ **0.86**
Macroaverage Precision $= \frac{.842+.882+.845}{3} = 0.856$				Macroaverage Recall $= \frac{.80+.90+.87}{3} = 0.86$	

Table 13. Result and Evaluation for Our Method

	System Output (Prediction)				
		H	R	S	
gold labels	H	100	0	0	Recall 'H' $= \frac{100}{100+0+0} = \frac{100}{100} = 1.00$
	R	5	91	4	Recall 'R' $= \frac{91}{5+91+4} = \frac{91}{100} = 0.91$
	S	4	3	93	Recall 'S' $= \frac{93}{4+3+93} = \frac{93}{100} = 0.93$
		Precision 'H' $= \frac{100}{100+5+4} = \frac{100}{109} = 0.917$	Precision 'R' $= \frac{91}{0+91+3} = \frac{91}{94} = 0.968$	Precision 'S' $= \frac{93}{0+4+93} = \frac{93}{97} = 0.959$	Accuracy $= \frac{100+91+93}{300} = \frac{284}{300} =$ **0.95**
Macroaverage Precision $= \frac{.917+.968+.959}{3} = 0.951$				Macroaverage Recall $= \frac{1.00+.91+.93}{3} = 0.946$	

VI. Result Comparison

We compare our work with the existing works in the field of authorship attribution. The comparison is shown in Table 14 and Table 15, where we have compared our work with the existing works of Bengali literature and other literatures respectively.

Table 14. Comparison of Our Work with Existing Works in Bengali Literature

Methods/Models Used	Dataset	Accuracy
Hidden Markov Model (HMM) [8]	Own Developed	90%
Support Vector Machine (SVM) [9]	Own Developed	83.3%
Character n-grams [10]	Own Developed	98%
Random forest classifier [11]	Own Developed	96%
Established and modified stylometric features [12]	Own Developed	90.67%
Graph based models [13]	Own Developed	94.98%
Fusion of N-Gram and Naïve Bayes (A new approach)	**Own Developed**	**95%**

From Table 14, we can see that, in most of the cases our method worked better than the exiting works for authorship attribution in Bengali literature. Only two works show higher accuracy than ours [10] and [11] (see Table 14). The result greatly depends on the methods used and mainly on the corpus. As shown in the table all the works including ours have used their own developed dataset and for this reason the base of comparison is not uniform. We believe that if we can run our method on those datasets, we will get better or at least the same accuracy as theirs. In [10], authors have used stop words, word unigrams, word bigrams, word trigrams, character bigrams and character trigrams as their feature category and achieved an accuracy of 98% by using SVM SMO. The main reason of their better performance is that they have developed a dataset which includes 1000 passages for each of the 3 authors; which is a huge dataset comparing to ours. In [11], they have used unigram, bigram and trigram count for their feature set extraction and gained an accuracy of 96% by using random forest classifier. They get only 62% accuracy with Naïve Bayes classifier and 85% accuracy with decision tree classifier. They have also developed a dataset which includes at least 100 passages for each of the 10 authors; which is a huge dataset comparing to ours. From other works in Bengali authorship attribution [8-9] and [12-13] (see Table 14) our method performs better. So, we can say that our method performs better than the base line method of naïve bayes, which we already have established through the experiment and it performs better than that of the most existing works for authorship detection in Bengali literature.

Table 15. Comparison of Our Work with Existing Works in Other Literatures

Methods/Models Used	Language	Dataset	Accuracy
Integrated Syntactic Graph (ISG) [1]	English	Mixed Dataset	71.5%
Character level n-gram [2]	English	English Data Set	100%
Character level n-gram [2]	Greek	Greek Data Set	97%
Character level n-gram [2]	Chinese	Chinese Data Set	89%
A hierarchical Naive Bayes mixture model [3]	English	(1)Web datasets (2)DBLP datasets	63.2%
Multi-variant Poisson Naive Bayes (MBNB) [4]	Arabic	Own Developed Dataset	97.43%
Author based Rank Vector Coordinates (ARVC) [14]	English	Own Developed Dataset	96.43%
Extended version of the probabilistic context free grammar language [15]	Arabic	Articles from Felesteen newspaper	79.4%
Fusion of N-Gram and Naïve Bayes (A new approach)	**Bengali**	**Own Developed Dataset**	**95%**

From Table 15, we can surprisingly see that, though authorship attribution in English literature is a much more established field; our work outperforms some existing work of this literature [1] and [3]. Our work also shows better performance than some existing works for author recognition of Chinese [2] and Arabic [15] literature. Though our result shows less accuracy in the cases of [2], [4] and [14] (see Table 15); as we mentioned earlier it greatly depends on methods and dataset used. We believe that if we can run our method on those datasets maintaining the standard of their literature, we can get better or at least the same accuracy as they have achieved. From the above mentioned three works two are done for English literature and one is for Arabic. From the table we can see that, all these three works used their own developed dataset. In [2], authors have used approximately 670000 characters for their English dataset and approximately 1360000 characters for heir Greek dataset. In [4], authors have developed a dataset that consists of 30 Arabic books written by 10 different authors. In [14], authors have used a dataset that includes 1185 poems of 6 authors. Comparing to all their datasets richness, our dataset is a much smaller one. At last we can say that our proposed method performs better than the most of the existing works for both Bengali and other literatures and it definitely performs better than the Naïve Bayes algorithm.

VII. Conclusion

Automatic Authorship attribution is a relatively new field in Bengali language than English language and even from other languages like: Arabic, Greek, Chinese and Hindi. In spite of being the fourth most spoken language

around the world the amount of work done in the field of natural language processing for Bengali is very poor. So as a part of natural language processing the work done in the field of authorship attribution is very poor also. So we take an initiative to do some work in the field of authorship attribution for Bengali language. And to do so, we choose three renowned authors of Bengali language named: 'Humayun Ahmed', 'Rabindranath Tagore' and 'Shamsur Rahman' and apply two methods to find their authorship. The two methods are Naïve Bayes and our newly proposed method. For both methods, we have to arrange datasets separately and apply the methods in their own way of working.

For our method, we combine both n gram and Naïve Bayes methods and the main reason to do so is: every author has a style of witting and for this reason some adjacent words are more probable for writing by one author. But some words are more likely to be used by one author even if they are not adjacent. So we find the bigram count for adjacent words and unigram count for the single words and combine N-Gram algorithm with Naïve Bayes to formulate our proposed method. As the summation of vocabulary and words in each class is large, we ignore the count of previous word in the denominator in our calculation. From the result analysis, we can clearly see that our method performs better than Naïve Bayes method in all the fields of evaluation metrics which are accuracy, precision and recall. For a larger and balanced corpus, it will perform much better.

In future, we will take more data and broaden our classes. Now our dataset contains 107380 total words and 21198 unique words for three authors. We have a target of classifying the authorship among at least five to six authors with a vocabulary of around 50000 words. We have planned to apply some other classifiers like: Neural Network; Support Vector Machine; Decision Tree and Hidden Markov Model on our dataset for authorship attribution.

REFERENCES

[1] Gómez-Adorno, Helena, Grigori Sidorov, David Pinto, DarnesVilariño, and Alexander Gelbukh. "Automatic authorship detection using textual patterns extracted from integrated syntactic graphs." *Sensors 16, no. 9* (2016): 1374.

[2] Kešelj, Vlado, Fuchun Peng, Nick Cercone, and Calvin Thomas. "N-gram-based author profiles for authorship attribution." In *Proceedings of the conference pacific association for computational linguistics, PACLING*, vol. 3, pp. 255-264. 2003.

[3] Han, Hui, Wei Xu, HongyuanZha, and C. Lee Giles. "A hierarchical Naive Bayes mixture model for name disambiguation in author citations." In *Proceedings of the 2005 ACM symposium on Applied computing*, pp. 1065-1069. ACM, 2005.

[4] Altheneyan, Alaa Saleh, and Mohamed El BachirMenai. "Naïve Bayes classifiers for authorship attribution of Arabic texts." *Journal of King Saud University-Computer and Information Sciences* 26, no. 4 (2014): 473-484.

[5] Murphy, Kevin P. "Naive bayes classifiers." *University of British Columbia* (2006).

[6] Kibriya, Ashraf M., Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. "Multinomial naive bayes for text categorization revisited." In *Australasian Joint Conference on Artificial Intelligence*, pp. 488-499. Springer, Berlin, Heidelberg, 2004.

[7] Kim, Sang-Bum, Kyoung-Soo Han, Hae-Chang Rim, and Sung HyonMyaeng. "Some effective techniques for naive bayes text classification." *IEEE transactions on knowledge and data engineering* 18, no. 11 (2006): 1457-1466.

[8] Banerjee, S. "Author Identification in Bengali language." (2013).

[9] Chakraborty, Tanmoy. "Authorship identification in bengali literature: a comparative analysis." *arXiv preprint arXiv:1208.6268* (2012).

[10] Phani, Shanta, ShibamouliLahiri, and Arindam Biswas. "Authorship attribution in bengali language." In *Proceedings of the 12th International Conference on Natural Language Processing*, pp. 100-105. 2015.

[11] Islam, Nazmul, Mohammed Moshiul Hoque, and Mohammad Rajib Hossain. "Automatic authorship detection from Bengali text using stylometric approach." In *Computer and Information Technology (ICCIT), 2017 20th International Conference of*, pp. 1-6. IEEE, 2017.

[12] Hossain, M. Tahmid, Md Moshiur Rahman, Sabir Ismail, and Md Saiful Islam. "A stylometric analysis on Bengali literature for authorship attribution." In *Computer and Information Technology (ICCIT), 2017 20th International Conference of*, pp. 1-5. IEEE, 2017.

[13] Chakraborty, Tanmoy, and Prasenjit Choudhury. "Authorship identification in Bengali language: A graph based approach." In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pp. 443-446. IEEE, 2016.

[14] Raju, NV Ganapathi, V. Vijay Kumar, and O. Srinivasa Rao. "Author based rank vector coordinates (ARVC) Model for Authorship Attribution." *International Journal of Image, Graphics and Signal Processing* 8, no. 5 (2016): 68.

[15] Abuhaiba, Ibrahim SI, and Mohammad F. Eltibi. "Author Attribution of Arabic Texts Using Extended Probabilistic Context Free Grammar Language Model." *International Journal of Intelligent Systems and Applications* 8, no. 6 (2016): 27.

[16] Menai, Mohamed El Bachir. "Detection of plagiarism in Arabic documents." *International Journal of Information Technology and Computer Science* 10 (2012): 80-89.

Authors' Profiles

D. M. Anisuzzaman is a Lecturer with Computer Science and Engineering Department, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. He is pursuing his MSc in Computer Science from American International University - Bangladesh, Dhaka. He has received his BSc from Ahsanullah University of Science and Technology, Dhaka in 2013. His research interest includes machine learning, neural network, computer vision, natural language processing and algorithms.

Abdus Salam is working as an Assistant Professor at American International University-Bangladesh (AIUB). His research interest includes Data mining, Machine Learning, Semantic Web, Intelligent Systems, and Human Computer Interaction etc. He has received his B.Sc. in Computer Science and Engineering from AIUB and M.Sc. in Computer Science major in Software Technologies from University of Trento, Italy. He can be conducted at salamt2@aiub.edu.

How to cite this paper: D. M. Anisuzzaman, Abdus Salam, "Authorship Attribution for Bengali Language Using the Fusion of N-Gram and Naive Bayes Algorithms", International Journal of Information Technology and Computer Science(IJITCS), Vol.10, No.10, pp.11-21, 2018. DOI: 10.5815/ijitcs.2018.10.02

