European Journal of Computer Science and Information Technology
Vol.9, No.1, pp.1-26, 2021
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

AUTHOR IDENTIFICATION BASED ON NLP

Noura Khalid Alhuqail
Dublin City University and Princess Nourah bint Abdulrahman University
Noura.alhuqail@gmail.com

ABSTRACT: *The amount of textual content is increasing exponentially, especially through the publication of articles; the issue is further complicated by the increase in anonymous textual data. Researchers are looking for alternative methods to predict the author of an unknown text, which is called Author Identification. In this research, the study is performed with Bag of Words (BOW) and Latent Semantic Analysis (LSA) features. The "All the news" dataset on Kaggle is used for experimentation and to compare BOW and LSA for the best performance in the task of author identification. Support vector machine, random forest, Bidirectional Encoder Representations from Transformers (BERT), and logistic regression classification algorithms are used for author prediction. For first scope that have 20 authors, for each author 100 articles, the greatest accuracy is seen from logistic regression using bag-of-words, followed by random forest, also using bag-of-words; in all algorithms, bag-of-words scored better than LSA. Ultimately, BERT model was applied in this research and achieved 70.33% accuracy performance. For second scope that increase the number of articles till 500 articles per author and decrees the number of authors till 10, the BOW achieves better performance results with the logistic regression algorithm at 93.86%. Moreover, the best accuracy performance is with LR at 94.9% when merged the feature together and it proved that it is better than applied BOW and LSA individual, with an improvement by almost 0.1% comparing with BOW only. Ultimately, BRET achieved result by 86.56% accuracy performance and 0.51 log los.*

KEYWORDS: author, NLP, identification, data analytics, analysis

INTRODUCTION

What if one could determine who wrote a piece of text? Reveal the writers behind the texts? Was Shakespeare the real author of his plays? If there were a system that allowed us to identify the primary author, such a system would enable us to answer those questions. Author identification works to preserve intellectual property rights, and prevent theft of articles, attributing each article to its primary author. It would enable governments or institutions to give authors credit where credit is due.

Problem Statement

Lately, there has been increased literary theft, loss of literary rights, and concealment of the original author of a particular article or paper. Anybody can take a copy of anybody else's work and put it on a website or in a paper with his or her name on it. The author identification process is significant for determining who deserves recognition for the text. It is not very easy to see an article in the name of another. It would be perfect if there were a system that could analyze and discover the unstructured article to assign the text to its primary author. As a result, NLP

analysis has emerged to analyze articles and extract features to predict author name. This study will focus on NLP analysis of given articles and how the NLP, based on machine learning algorithms, will help to predict the author's name.

Research Questions

The research will answer the following questions:

- How can the models predict the author's name from a published article?
- Which model of feature generation, between Bag-of-Words (BOW) and Latent Semantic Analysis (LSA), performs the best for the task of author identification?

Aims and Objectives

The aims and objectives of this study are as follows:

- Predicting the author's name from a given article.
- Comparing BOW and LSA, to find which performs best for the task of author identification.
- Using different classifier models to predict the author's name.
- Comparing the performance of multiple classifiers.

Scope of the study

The search scope is as follows:

- The research concentrates on studying author identification analysis based on NLP for published articles.
- The scope is for twenty authors in two newspapers.
- The research concentrates on English articles only.

LITERATURE REVIEW

Natural Language Processing (NLP)

Natural Language Processing is the method used to aid machines to understand human natural language. It is a section of artificial intelligence that deals with the interaction between machines and humans using natural language. NLP aims to read, decode, analyze, understand, and make sense of human languages to derive meaning. Authorship identification is an essential topic in the field of NLP. It enables us to identify the most likely writer of articles, news, text, or messages. Authorship identification can be used to identify anonymous writers or detect plagiarism.

Authorship analysis

Authorship analysis is a challenging field that has evolved over the years. It is the procedure of finding the characteristics of a text in order to draw conclusions and analyze its authorship. Stylometry is the root of authorship analysis, which means the statistical way to analyze the text style in order to characterize the author. The concept of authorship analysis can be defined and divided into three sections as follows:

- Authorship identification (authorship attribution): Finding the real writers of an article or document and the possibility of an author having written some text.
- Author profiling (characterization): getting the writer's profile or characteristics; for example, gender, age, background, and language.
- Similarity detection: Finding the similarity between the texts to determine the possibility of them having been produced by a single writer, without necessarily finding the real author. Commonly used in plagiarism detection.

Data gathering

The authorship identification dataset includes varied sources of previous work, including books, scientific papers, articles, and even emails. Still, ultimately, the focus was on the text regardless of its type.

PAN dataset: In studies [1] [2], the dataset comprised documents from the PAN competition dataset for Authorship Attribution. It is a publicly available dataset, focused on Authorship analysis. The author analysis in [3] was carried out by using the same dataset. They focus on author multi-genre and multi-language problems. It has a combination of genres, like essays, novels, articles in Spanish, English and Greek, and the total number of documents is 7,044. Two datasets were used in [4] The first is PAN 2012; the second is the Urdu articles dataset, which has 4,800 articles written in twelve well-known Urdu newspapers, with 400 articles by each author.

Reuter dataset: Study [5] worked on two different datasets. One is the Reuters news dataset, which is widely using for authorship identification; it is an archive of over 800,000 newswire stories. The second dataset is the Gutenberg dataset that was established by the author, containing 53,000 e-books on the Internet. Study [6] wrote that they used a subset of the Reuters dataset, including 50 authors, who have each written 100 articles. Study [7] used 21 English books, written by ten different authors, as well as a collection of news stories from the Reuters dataset. Likewise, the dataset used in [8] was based on the Reuters dataset; they chose all authors who had 200 or more articles. The collected dataset contained 114 authors who wrote 27,342 articles in total. Two types of text corpora have been used in [9]; one in English, the Reuters newswire stories dataset; and the other in Arabic (newspaper reportage from Al-Hayat website). Both contain different authors, with 100 texts for each author. The Reuter_50_50 Dataset was applied in [10]; it contains 50 authors and 50 texts per author.

Articles: In study [11] the dataset was manually gathered from several Arabic websites. The dataset consists of 10 authors, with 10 articles for each author, while [12] developed a dataset containing text from different newspapers. The topics of these articles are about current events, political and medical issues. There are 20 authors, with 20 texts for each author in the training set, while, for the test set, there are 20 authors with five different texts for each. In [13], the research consists of approximately 145 student essays of about 1400 words for each essays. The essays are a real description of the Artificial Life documentary and the students' opinions about it. Thereby the topic, age, and level of education are constant. In study [14], the dataset contains 20 different authors who write about Economics, Sports, Literature, and miscellaneous subjects. The articles were obtained from two Brazilian publications. Each writer has 30 pieces. Work [15] is based on thirteen selected Nigerian writers from a Nigerian national daily. They harvested articles published from 2014 to 2016, and collected a total of 20 articles per author, so the total is 260 articles.

Papers: Study [16] used the ACL anthology network corpus dataset, which contains 23,766 papers and 18,862 authors. Also in the field of scientific papers, [17] used ACL papers. The dataset includes scientific papers published in several conferences and workshops. The selected papers were from 1965 to 2007; they classified all 2006 papers as development data and all 2007 papers as test data and the remaining papers were used for the training set.

Emails: In [18] [19] [20] a real-life dataset was obtained from the email records of Enron, which is an energy company. The employees' emails were made public, and the dataset

contains approximately 200 thousand emails from about 150 employees. The average word count per email is 200. The emails are simple texts and cover several topics ranging from business connection to technical reports and personal conversations.

Other data types: In [21], the authors gathered a collection of 23 novels. They selected six of them as the experimental dataset. There are approximately 22,000 texts. For each author, they selected text from random novels to ensure they were cross-topic.The method of collecting data in [22] was different: the author relied on only two books, and each book was divided into parts and saved in different files. Testing can be executed with various training sets from the book chapters of both authors then different book chapters by the same author are tested to determine the accuracy of the prediction. In [23], the author collects six types of text in different languages (Dutch, English, Greek, and Spanish), and genres (essays, reviews, novels, and articles). A Greek blogs dataset was created in [24] from scratch.They manually collected 100 Greek authors' blogs. In their study, they used 20 blogs with a common topic of personal affairs. The total is 1,000 blog posts with a total of 406,460 words. For each author, they collected 50 recent blogs. In [25], the writer collected online messages for Cyber Forensics Analysis. The messages were written by the author who tried to hide their real identity to void detection.

Random Forest: Pre-processing is a significant step in text mining. It means turning the text into a form that is predictable and analyzable. In [1], the pre-processing was divided into two types of features, depending on the requirement of the model. In the Bag of Words model for extracting content-based features, the author applied stop word removal and stemming, then extract the most frequent specific terms and consider them as a bag of words. The Bag of Words model is used for extracting n-gram features that tend to appear in the author's writing style and can be used to compare the writing style of one author to another, Therefore, as the first step, the author removes punctuation marks and extracts the most frequent character n-grams, word n-grams, and POS n-grams. Once the dataset is prepared and pre-process works done, feature extraction is needed to convert the data to vectors. In this step, the author uses the bag of words to represent the data vector, then uses classification algorithms; specifically, the Naive Bayes Multinomial (NBM) and Random Forest (RF). The author compares predictions with the most frequent content-based features with the accuracies of the most frequent character, word, and POS n-grams. The best results of author prediction are achieved when the author uses a combination of content-based features and n-grams, using the Random Forest classifier algorithm, with 91.87% accuracy. [3], on the other hand, evaluates the extracted features - unigram features, Latent semantic features, and similarity - by producing a supervised machine learning algorithm comparing Logistic regression, Random Forest, and SVM. The Random Forest tree produces higher performance than other models. with accuracy of up to 80.12%.

Support vector machine: Some authors, like [11], did not do any pre-processing work. In their view, the reason for this is to keep the text as it is to indicate the unique writing style of each author. Moreover, they classify and rank tasks and use SVM-Light, which is an open-source tool common in the machine learning community, with an interface to train and test a model. The author extracts the features and bundles them into five groups: F1: lexical; F2: lexical and syntactic; F3: lexical, syntactic, content-specific; F4: lexical, syntactic, content-specific, structural; and F5: lexical, syntactic, content-specific, structural, and semantic. They test all five groups of features, and the accuracy calculated as a total can correctly identify the author of an article from the test sample. The accuracy performance of this bundle of feature set is successively; F1: 88%; F1+F2: 92%; F1+F2+F3: 95%; F1+F2+F3+F4: 96%; and

F1+F2+F3+F4+F5: 98%. In [21], on the other hand, the authors focused on four essential features, which are:

1. Character-based features used to clarify the style of writing. For example, if there are many commas, then the author is more formal, and if there are many questions, the author is more emotional.
2. Word-based features commonly used in author identification, referring to word statistics information, rather than using words directly. In addition, the word-based features analysis applies the standard deviation of the word length; average word length; the difference between the maximum word length and the minimum word length.
3. Sentence-based features, fundamental to describing the construction of the text or article. Different authors use different constructions to write their articles. Some authors' styles are simple, so the sentences in their articles are possibly shorter, while other authors possibly prefer long sentences. The author of this study used the mean length of sentences, the standard deviation of sentence length, and the difference between the maximum and minimum values of the sentence length.
4. Syntactic features which are analyzed by syntactic analysis tools. Syntactic features refer to the grammatical relationship between words in sentences. Therefore, they use a support vector machine (SVM) and a linear kernel. They use the tools released by HIT: pylyp function for the segments of Words and parts-of-speech. They use two types of performance measurement; accuracy and PRF scores. They show that the accuracy and the f1-score of using the syntactic features alone rises about 12%, indicating the efficiency of using syntactic features alone, rather than other essential features. The author concluded that using syntactic features alone reduced the size of the feature set to decrease the computational overhead, and showed the high possibility of the syntax tree for author identification.

In study [7], the text was analyzed in several ways: tokenizing, part-of-speech tagging, phrase parsing, and typed dependency parsing. Then they identified pronouns, function words and non-subject stylistic words. Therefore, they used k-nearest neighbors (KNN), support vector machine (SVM), and latent Dirichlet allocation (LDA) and made comparisons between the performance of different selected feature sets. They used the LIBSVM package for SVM. and a fivefold cross-validation way to select it from the candidate dataset. The core approach is a collection of n-gram features and SVM, excluding PCA feature extraction, and n is a positive integer. The LDA achieves higher performance by 98.45%.

In [24], a Greek blogs dataset used a set of stylometric features. The features include classic stylometric features, such as lexical, word length measures, and features extracted from n-grams. They use the extracted features and the Support Vector Machines algorithm to reach 85.4% accuracy in authorship attribution. The feed-forward neural network was used in [25], with a radial basis function network, and Support Vector Machines applied to predict the authorship of anonymous online text. They begin by extracting features for each unstructured text, which appear as a vector of writing-style features.

Study [26] investigated authorship identification of Telugu text by using several features: average number of words, sentences, syllables per word, word length, sentence length, parts of speech, bigrams and trigrams of the word. They used a support vector machine classifier for

feature vectors. The accuracy performance of the SVM model for authorship identification was measured, and the results showed that character n-gram features occurred at a higher rate than all other features. The combination of several features, like word grams mixed with lexical and vocabulary features, reached a higher rate than applying the features separately.
In [15], the authors performed experiments, applying five different subsets of the main attributes by using a rough mechanism. The results of the experiment showed that using a rough set mechanism improved the accuracy performances for both neural network algorithms and the Supported Vector Machines algorithm. The classification model accuracy increased to 50.505% for the NN and 28.662% for the SVM algorithm. However, the NN algorithm performed better than the SVM algorithm.

In [10], the author introduces the Stylometry approach and n-gram features for the authorship identification task. They achieved 85% performance accuracy for the SVM classifier model. In [27], the authors suggested using a support vector machine model for heterogeneous documents. Moreover, experimental results showed that applying both n-grams and sequential word patterns together achieved better accuracy performance than n-grams alone.

Latent semantic analysis: In [23]**,** the author focused on representing the writing style of the authors; they used lexical-syntactic features. The work is divided into two levels; the first one is Phrase-level features: Word prefixes, Word suffixes, stop words, Punctuation marks, Word n-grams and Skip-grams; the second one is Character-level features: Vowel combination and Vowel permutation. The reason for using lexical-syntactic features is due to the ease of identification. The writer treated them with unsupervised classification, using several metrics to determine the similarity of the feature vectors of the documents of the known author against the documents of the unknown author. To establish the similarity of the documents, they used Latent semantic analysis (LSA), Jaccard similarity, Euclidean distance, Cosine similarity and Chebyshev Distance.
In [2], they create n-grams by sliding a window along the document to use as the features for the Document matrix. Thus, they develop an effective LSA by the use of character n-gram-based analysis. In terms of accuracy, the percentage reached 75.68% for Dutch reviews in the PAN dataset.

Cosine similarity: In [16]**,** the authors used the n-gram frequency technique and reflected unigram, bigram, and trigram technique and implement stop-word removal, then used the Python NLTK package for Porter Stemming. There are two kinds of features extracted to characterize the paper or the author. They apply LINE heterogeneous network embedding adapted to suit author identification. They apply the trained model by taking both the input paper embedding vectors and the author embedding vector and then create embedding for the test papers of the anonymous author. They compare the embedding of the trained author vectors with the test paper embedding results. They use cosine similarity over the embedding vector values to assess the distance between a potential author and the test paper, with the accuracy of results reaching 66.66%.

In [4], the author decided that there was no need for robust pre-processing in authorship attribution. Spelling errors, letter abbreviations, and letter capitalization are an essential part of writing style, so they decided not to fix grammatical errors or word stems; such actions may reduce the number of features for a writer. Nevertheless, they carried out the following for document pre-processing: tokenization, lowercasing, n-gram generation, and stop-word removal. For syntax analysis and feature extraction, they carried out TF-IDF and bag-of-words

extraction. The LDA approach is focused on instance-based and profile-based classifications of author identification. LDA is an unsupervised methodology that can handle a variety of writing, high-dimensional and sparse datasets by allowing more text. The author used cosine similarity alongside n-gram-based LDA to measure similarity in vectors of text. They achieved overall 84.52% accuracy in the first dataset and 93.17% accuracy on the second dataset without applied any labels to identify author tasks.

Bayesian Classifier: In [22], they extracted lexical features, such as the number of words, statistical information, and syntactic features for classification. These are essential features because different authors mean a different level of vocabulary. The vocabulary level of the author can be decided by the total number of single words they use in the text. Although the dataset is small, the predictive accuracy performance, which is measured by using k-fold cross-validation, is low when using the Bayesian Classifier. They proved that the accuracy decreases if the dataset volume decreases. Also, as the data volume decreases, the uncertainty of the actual predictive accuracy increases; the results from a small dataset will not be as accurate as of the large dataset. In [8], the author used Bayesian multinomial logistic regression to build classifiers on various data sets.

N-gram: While classical documents are very well structured and provide various stylometric features, an e-mail [18] consists of a few paragraphs, written by an employee quickly, and frequently with syntactic and grammatical mistakes. All the sample e-mails are divided into groups to build a given author profile into one document that is subsequently divided into small blocks. They applied these processes: replace all numbers with 0; normalize the emails to printable ASCII; converted the emails to lowercase characters; remove white space; remove any punctuation; group all emails by author, to make a document that is divided into blocks.
In the Enron email dataset, the Equal Error Rate (EER) was14.35% for 87 employees for small block sizes. While the acquired results are hopeful, the author decides that more effort must be made to be usable in the real world. They discussed the limitations of their approach. The accuracy decreased, not only when the number of authors number increased, but also when the number of blocks per employee decreased. They applied 5-grams that achieve better results than 3- and 4-grams for a large number of blocks per user.

However, in [28], pre-processing was required to produce the character n-gram profile. The author removed numerals from the text, eliminated all punctuation marks, partitioned the text into separate tokens, locating all possible n-gram for N = 2, 3, then making sure that each output n-gram in the list has its frequency, sorting the n-gram frequencies in descending order, and for each author, they build a profile size for bi-grams, tri-grams and quad-grams. The authors create the bi- & tri-grams from the author's text called Author's Profile. n-gram was used to calculate the dissimilarity between the frequency of the n-gram in the Author's Profile and the frequency in the test data.

Other methods: The pre-processing in [5] primarily consists of two parts. The first one is word representations, where the authors used the GloVe word vectors to initialize the word embeddings and excluded the occurrences of numbers and special characters to match the features of the word representations. During the pre-process, the author trimmed each word to ensure that it did not include any number or special character. The second technique is Input Batch Alignment; there is a fixed-length batch as input, with the input truncated if it exceeds the fixed length. If there are words that cannot be found in the GloVe, they are replaced by a magic word that is created by the author; it is a word that does not exist in the real world. The

magic word was hidden to remove its effect on the output and only the actual words were extracted. They implemented four authorship identification deep learning models. The best two were: Article-level GRU, which achieved 69% on the Reuters news dataset and 89% on the Gutenberg dataset; and the Siamese network, which performed with 99.8% accuracy on both the C50 and Gutenberg datasets.

The author of [6] suggested randomly partitioning the dataset into three groups: 60% for the training set and the validation set, with 10% of the dataset and applied hyperparameters to choose the best order, and finally they used 30% of the dataset for evaluating the classification performance test set. They applied deep learning for author identification, and they compared the performance between the features extracted. The author showed that the chi-square-based feature produced a high performance compared to frequency-based features. To produce a high accuracy, the author applied min-max normalization. 95.12% is the systematic classification accuracy.

In the ACL papers dataset [17], they ignore the first ten lines of each paper document in order to exclude author names, publications, emails, and business information. For authorship identification prediction, they used a convolutional neural network (CNN). Each sentence appears as a padded series of word embedding vectors and POS-tag one-hot encodings. They proved that extraordinary words support system performance. They achieved 95% accurate performances on the training dataset.

[12] studied 35 style markers for an average of 20 articles of each author in the training set. For stemming words, they used the Turkish NLP library, Zemberek, which is a Turkish Natural Language Processing system, to identify 35 style markers; with this model, they achieved an accuracy of 70.75%. Then they selected 22 style markers, which were the most effective ones. They obtained the best success with Naive Bayes Multinomial, that was 80% after attributes were extracted using the CFS Evaluator with the Search of the Rank method.

In [9], the author decided not to perform pre-processing of texts for the dataset, apart from deleting XML and HTML labels irrelevant to the text content. The author represents four methods. The first is under-sampling of the classes, based on the training set; the second method is under-sampling of the classes, based on training set lines; the third method is re-balancing the set by document samples of variable length; the fourth method is re-balancing the set by document re-sampling.

In [14], they discuss applying compression algorithms for authorship identification. They apply three types of compressors: statistical type, Lempel-Ziv type, and block-sorting type. The Normalized Compression Distance (NCD) and Conditional Complexity of Compression (CCC) were applied to compute the dissimilarity between two documents. For the instance-based approach, NCD is suitable, while CCC gives better results when used with the profile-based approach.

In [29], the naive Bayes classifier was shown to be unusable in author identification, despite the simplicity of the model, but the author applied naive Bayes and proposed two types of feature selection process, based first on a univariate feature extraction and then feature clustering. They prove the effectiveness of their method by evaluating and comparing 13 datasets. The performance refinement thus achieved makes the proposed algorithm comparable with other classifiers.

In study [19], the author applied a model for email authorship identification by using a Cluster-based Classification (CCM) technique. Stylometric features were used, which were extended to get more useful features for email authorship identification. In addition, they used Info Gain featuring extraction-based content features. There was a positive impact on the accuracy by using these features. The results show that the suggested model of CCM-based email authorship identification outperforms the models based on the support vector machine (SVM). The suggested model gets 94% accuracy for ten authors and 89% for 25 authors on the Enron dataset.

Study [20] focused on three types of authorship analysis problems: authorship identification one time with large training samples; a second time with small training samples; and authorship characterization. Furthermore, they proposed a unified data mining process based on the novel notion of frequent-pattern-based writing to solve the three problems.

In [30], the author proposed an algorithm for authorship identification of poems and editorial documents. The research results showed the effectiveness of the Author-based Rank Vector Coordinates (ARVC) model to recognize poets and [27] authors. The authors applied lexical features, removing common words. They believe that the suggested model can help in tracking the author's identification in cyberspace, online messages, books and novels.

In the thirty previous studies on author identification task analysis, the authors collected data from various sources, such as the PAN dataset, Reuter dataset, articles, papers and emails. After that, the pre-processing stage starts by preparing the data; for example, they removed stop words, converted all upper-case words to lowercase, and removed special characters. Then, they implemented several types of feature extractions, such as TF-IDF, LSA, and Bag-of-words. Moreover, the authors applied a variety of machine learning algorithms, such as Random Forest, SVM, Naïve Bayes, Logistic regression.

METHODOLOGY

Feature Extraction

The news articles dataset is textual, so it is necessary to extract and produce feature representations that are convenient for the author's identification task. Moreover, the textual dataset cannot be entered directly into the model as it is: the dataset needs to be transformed into a numeric feature in a meaningful way. Feature extraction was applied by using the following methodologies:

Bag-of-Words (BOW)

Bag-of-words is a way to represent textual data for a machine-learning algorithm that supports the author identification task. The BOW involves term frequency that calculates how often the word is present within a given article and extracts features from the articles to enter in algorithms. Moreover, bag-of-words takes the articles as input; it counts how repeatedly each word appears in the dataset. BOW works as follows: (i) it splits each article into words (tokenization); (ii) it creates vectors by converting words that appeared in all the documents, and numbers them to be used in the algorithm; (iii) it checks how often each word in the vocabulary appeared in each document. The final output is a matrix representing each word and how much it is present in each document.

The motivation for using bag-of-words:
The bag-of-words algorithm has seen great success in so many cases, for instance, author identification and document classification. The approach of BOW is flexible and can use in several ways for extracting features from the text. However, BOW different from the rest because the structure of words in the file is not essential. The model is concerned with whether words occur in the file, not where in the file.

Latent Semantic Analysis (LSA)
Latent Semantic Analysis is an automated, unsupervised statistical-algebraic summarization method. LSA follows an extractive approach to analyzing documents and finding unobserved semantic relations between words and sentences of the articles (text). LSA is popular in the scope of natural language processing for information retrieval and textual mining, document comparison, and returns semantic similarity between words and documents. LSA supposes that semantically similar words will occur in the same article. It analyzes the relations among a set of words and documents containing them by producing a collection of concepts. Every document appears as a two-dimensional matrix with $m \times n$ dimensions, where m is the number of words, and n is the number of sentences in preprocessing status text. Internally, and to clarify, LSA uses singular value decomposition (SVD) to reduce the number of rows of the matrix by applying cosine similarity comparison of words and decreasing noise [31].
Singular Value Decomposition (SVD)

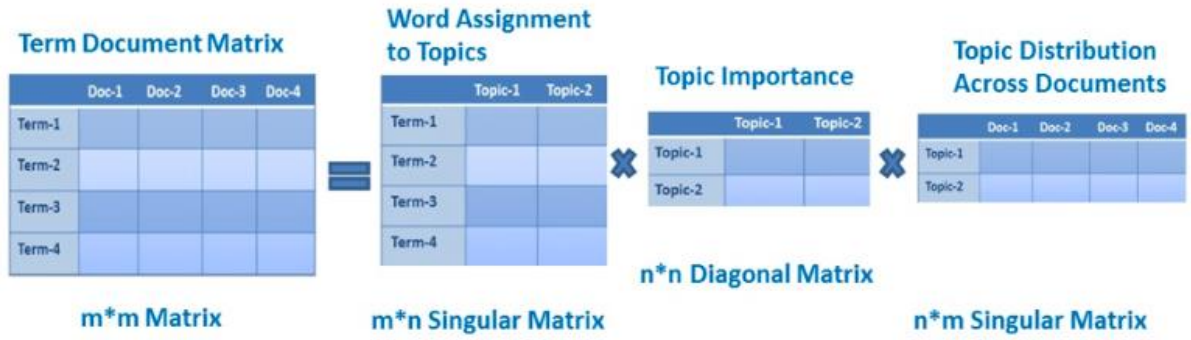

Figure 1. Singular Value Decomposition Process [32]

Singular Value Decomposition is a matrix factorization technique that creates one matrix from the product of two matrices:

$$M = U\Sigma V^* \qquad (1)$$

M: is $m \times m$ matrix
U: is $m \times n$ which stores left singular Matrix, where m act words and n act concepts or topics.
Σ: is $n \times n$ diagonal matrix with non-negative value.
V: is $m \times n$ matrix which stores the correct singular value, where n is the number of sentences in a document and m represents the concepts discovered
V*:is $n \times m$ matrix, which is the conjugate (transposition) of the V.
A diagonal matrix is a matrix in which the entries are all zero except the main diagonal. A singular matrix is 0 or a square matrix that does not have a matrix inverse.

The motivation for using Latent Semantic Analysis

The reason for use LSA because it is working by identifies thematic components present in some text instead of counting words. LSA analyzing relationships between the words in documents and assumes that terms that are close in meaning will occur in similar pieces of articles (distributional hypothesis).

Classification Algorithms

Classification algorithm techniques were applied in this project for learning from past data to classify future information. Figure 2 illustrates how machine learning algorithms work for the author identification task.

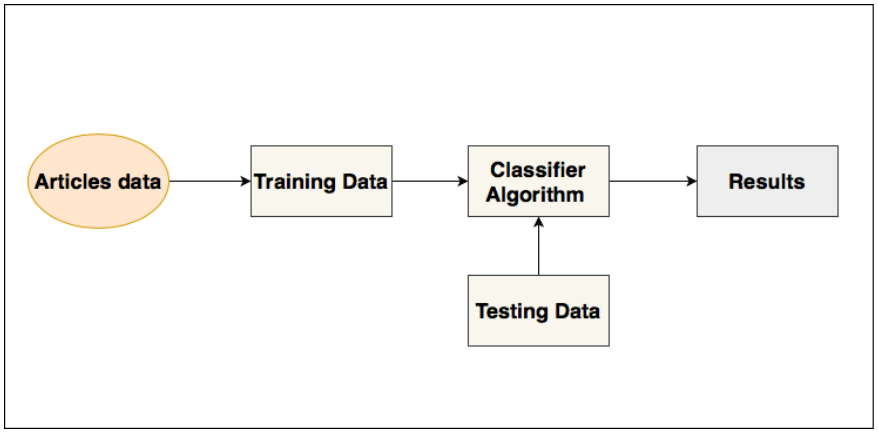

Figure 2. Classification steps for NLP

Logistic regression (LR)

Multinomial Logistic Regression (MLR) studies the probability of the D-dimensional point $xi \in \{x\,1\,, x\,2\,, x\,3\ldots, x\,N\,\}$ pertinent to a class $k \in \{1, 2, \ldots, K\}$. The data of the shape $(x\,i, y\,i\,)\; i = 1, \ldots, N$ wherever $x\,i \in R$, d is a d-dimensional representation of the feature vector and $yi \in \{1, 2, 3, \ldots, K\}$ is a label connected with it; K is the number of class labels. Let $yik = I\,(y\,i = k)$ indicate the membership of data point x i to class k. $W = \{w\,1\,, w\,2\,, \ldots, w\,K\,\}$ indicates the parameter vector for all of K classes. The probability that x i is pertinent to class k is given by:

$$p(y = k|x\,i\,) = \frac{exp(\,W_K^T\,xi\,)}{\sum_{j=1}^{K} exp(W_K^T\,xi\,)} \qquad (2)$$

The motivation of using Multinomial Logistic Regression

Logistic regression widely used algorithm; it is very active and interpretable and very efficient to train. As well, the outputs well-determine predicted probabilities.

Random Forest (RF)

Random Forest is a classification algorithm that works by making several independent decision trees that operate as an ensemble, then decides by taking votes of those trees. Moreover, making several separate decision trees decreases the chance that it overfits and the chance that it makes mistakes, because usually, the majority of trees will point in the right direction. Figure 3 is a

visualization of a random forest model making a prediction. Three decision trees voted 1, and one decision tree voted 0, and thus result in prediction number 1.

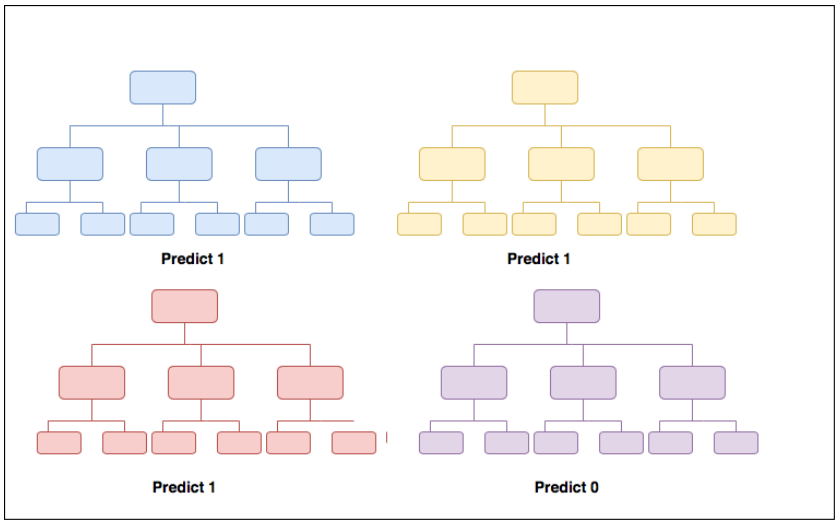

Figure 3. Visualization of a Random Forest model making a prediction

The motivation for using Random Forest

Random Forest approach used to increase the performance of individually prosaic Decision Tree algorithms. The principle is that a group of "prosaic learners" can come together to shape a "powerful learner." Each Decision Tree, individually, is a "prosaic learner," while all the Decision Trees taken together are a "powerful learner."

Support Vector Machine (SVM)

A support vector machine is a non-probabilistic classification algorithm that works by trying to find a decision boundary between two classes by maximizing the distance. Moreover, to use SVM in nonlinear regression, a Kernel trick is used to take low-dimensional input space and convert it into high-dimensional space.

The motivation for using Support Vector Machine

It is supposed to be a binary algorithm that has been a great success in many researches. However, it is more often to solve an issue that has a multiclass because it is work by divide the multiclass issue into multiple binary classification sub-issues [33].

Bidirectional Encoder Representations from Transformers (BERT)

BERT is a mechanism for Natural Language Processing pre-training, created by "Google. BERT". The BERT model is designed to help the understanding of how to represent words and sentences with the best captures of meanings and word relationships. The model has two sizes; BERT base, and BERT large. Since BERT is a pre-trained transformer encoder stack, both model sizes have many encoder layers, or transformer blocks. The base version has twelve encoder layers, and the large version has twenty-four encoder layers.

BERT takes a series of words as input to the transformer blocks. Each encoder layer in the transformer blocks makes a self-attention, and pushes the results through a feed-forward network, and then delivers it off to the next encoder. The output is a vector (a list of numbers) that can use it as the input for a classifier, such as a single-layer neural network. For standard word embedding, the output is a vector that shows the words in a method that captures semantic or concept-related relationships. In BERT, embeddings are pre-trained on large amounts of text data, instead of training the data alongside the model on a dataset. However, in BERT, a list of words is downloaded and their embeddings created by pre-training with Word2Vec or GloVe. Embeddings from Language Models (ELMo), which is a deep contextualized word representation. It examines the entire sentence before specifying each word for an embedding. It applies a bi-directional long-short-term memory (LSTM), which is an artificial recurrent neural network architecture. The LSTM is trained on a particular task to be capable of creating those embeddings. In addition, that language model has a sense of both the next and previous words [34].

The motivation for using Bidirectional Encoder Representations from Transformers (BERT)

BERT's critical technical innovation is that the first in applying deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. However, it is in contrast to previous efforts that looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The results for BERT show that a language model that bidirectionally trained can have a more profound sense of language context.

Evaluation technics

Evaluation parameter (EP)

An evaluation parameters matrix was employed in this study to examine the performance of machine learning algorithms. The output results from the algorithm measures are: Accuracy, Recall, Precession, and F1. All of them were used in the study, and the meaning of each one will be clarified:

Accuracy shows whether a model is being trained correctly or not. The accuracy percentage of authors that are identified correctly, divided by the total number of authors identified.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (3)$$

Recall is the percentage of the authors identifed and classified as positive divided by the total number of authors identified that are truly positive.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative\ (Predicted\ Results)} \quad (4)$$

Precession measures the percentage of positive authors identified and classified correctly divided by the total number of authors identified that are classified positively.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive\ (Actual\ Results)} \quad (5)$$

F1 is a metric that takes into account both precision and recall with the following equation.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

Log loss (LL)
Log loss measures the model's performance where the prediction output is a probability in the domain between 0 and 1. The model is supposed to minimize the value of log loss. Therefore, if the probability result is close to 1, it means that the prediction value is far from the true value and it is a poor model. Instead, if the model supplies values close to zero, it is a preferable model. The feature extraction methods were discussed, such as bag-of-words and LSA. In addition, the classification approaches were explained; for instance, logistic regression, random forest, and support vector machine. Finally, the evaluation technics were explained

Data Collection & Analysis
The amount of textual content is increasing exponentially, especially through published articles. The author identification process is significant for determining who deserves recognition for the text. This chapter will describe the dataset used for the author identification task. Moreover, it will explore the data representing the number of publications, number of authors and articles, and the average number of words per article.

Data Collection
In this project, the "All the news" dataset used from kaggle.com is a collection of news articles published in various publications, like the New York Times, Breitbart, CNN, and Business Insider, and by various authors. Most articles were published between 2016 and 2017 and cover multiple topics. However, this project will limit the scope to two publications.

Data Description
The "All the news" dataset contains 9 features and 50,000 news articles, categorized as table 1.The 'author' is a target column to predict and 'Content' Contains the text of the article that will be used to extract features.

Table 1. Data type for all the news dataset

Features	Description	Data Type
ID	Database ID	Integer
Title	Article title	String
Publication	Publication name	String
Author	Author name	String
Date	Date of publication	Date
Year	Year of publication	Year
Month	Month of publication	Date
URL	URL for article	String
Content	Article content	String

Data Exploration

Number of publications

First, the numbers of articles from each publication is shown in this histogram. Figure 4 demonstrates the distribution for top 5 Publications.

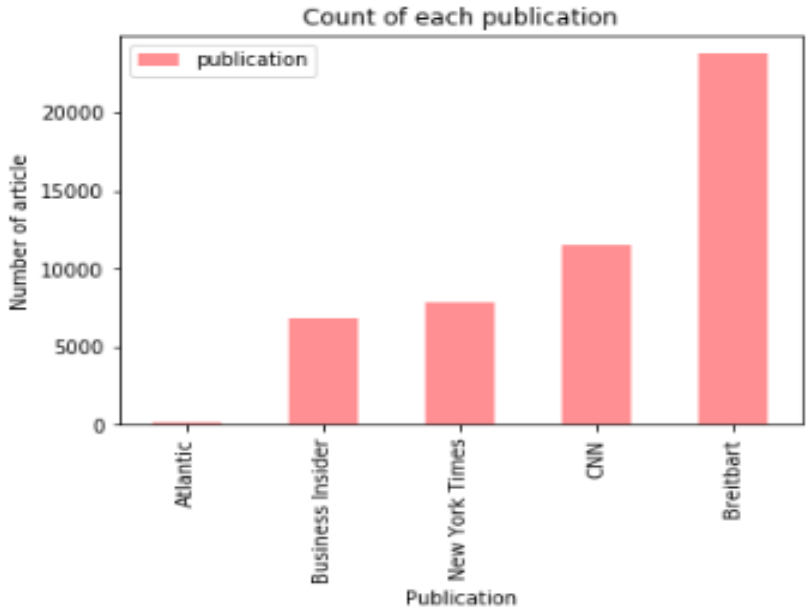

Figure 0. Number of articles per publication

The total number of publications is 50000 and from the histogram, it is clear that the top two publications are Breitbart and CNN that will be selected to limiting the scope. Table *2* shows the exact numbers of articles for each publication.

Table 2. The numbers of articles per publication

Publications	Number of articles
Breitbart	23781
CNN	11488
New York Times	7803
Business Insider	6757
Atlantic	171

Number of authors and articles

To limit the scope, two publications were selected based on the number of articles: they are Breitbart and CNN. The total articles in the two publications is 35269 from 914 authors

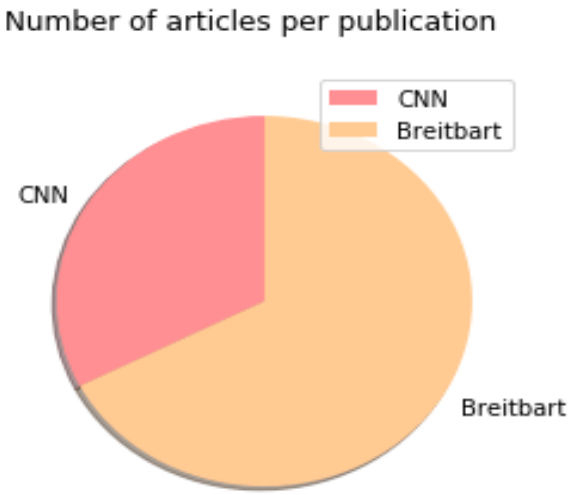

Figure 5. Number of articles for Breitbart and CNN

Next, the first scope is top ten authors with more than 100 articles were selected from each newspaper; this gave a total of 20 authors. From each author, 100 articles were selected, so that the number of articles became 2000. Figure 6 and Figure 7 shows each authors selected and their article count. The second scope is ten authors selected from Breitbart publication. From each author, 500 articles were selected, so that the number of articles became 5000.

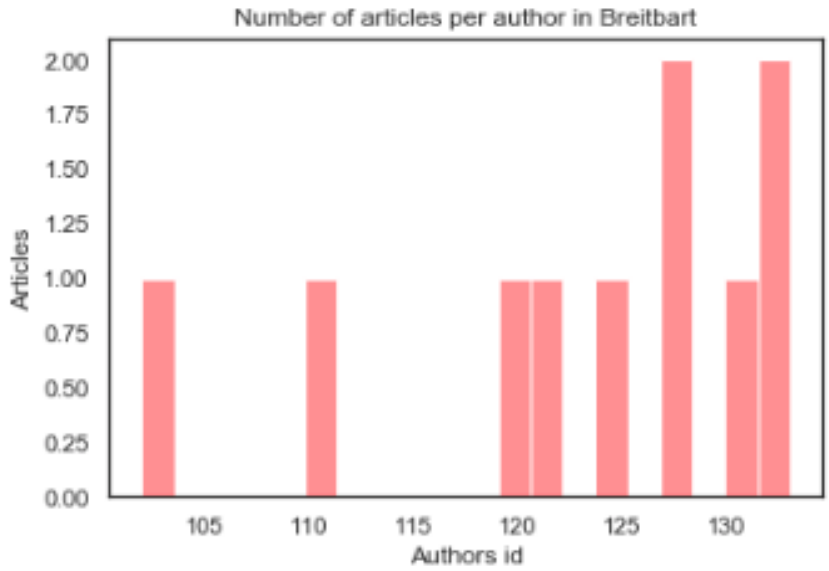

Figure 6. The number of articles per top 10 authors in Breitbart

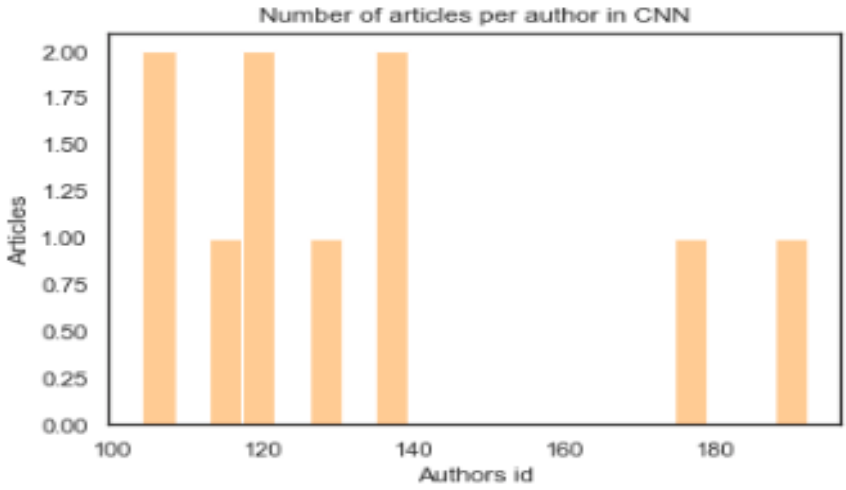

Figure 7. The number of articles per top 10 authors in CNN

Average number of words

The average number of words for the selected articles is 599.509. The word cloud image in Figure 8, represent the most repeated word and it shows that the word "Trump" is the most frequent word. The reason is that the data was collected between 2016 and 2017 (the time that Donald Trump became president).

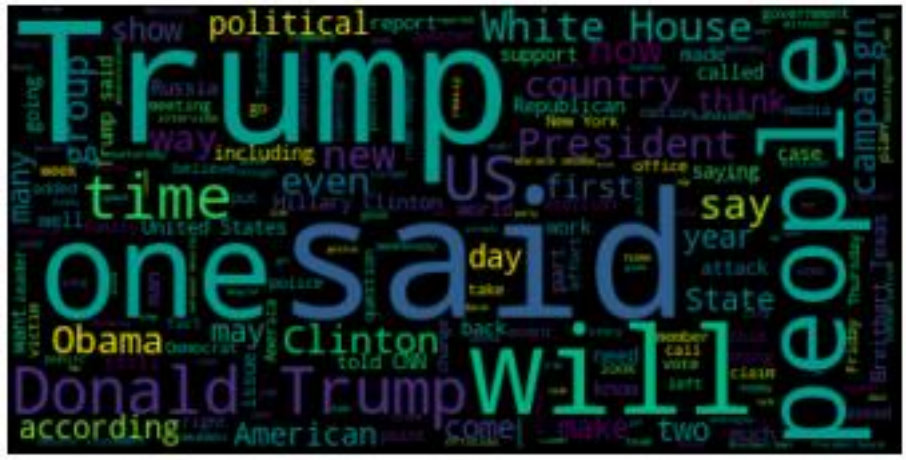

Figure 8. Word cloud image represent the most repeated word

Implementation

The pre-processing steps that were applied in the research, and how to clean and prepare the articles dataset before entering it to the algorithm. It also presents the feature extraction and output. Finally, it describes the implementation of the models and explains the results.

Pre-processing

Since this task deals with a textual dataset, it requires suitable preparation before using it in machine learning algorithms. For preprocessing the data in this project, the author identification task technique was followed to indicate the unique writing style of each author by keeping rare words, without correcting the spelling and lemmatization. However, the

following preprocessing actions were taken: checking for missing data, checking for duplicated data, tokenization, removal of stop words, removal of too common words, and adjustment of all characters to lower case.

Checking for missing data and duplicates

To check for missing data in this project, the "missingno" library was installed to supply a flexible tool for a missing data visualization summary. Figure 9 shows the dataset labels. There are some missing data in the author columns. However, because of the limitation of the scope, the missing data was eliminated and authors chosen with no missing labels. It is clear that the URL content is empty, and this is not used in the project. In addition, there are no duplicates in the dataset.

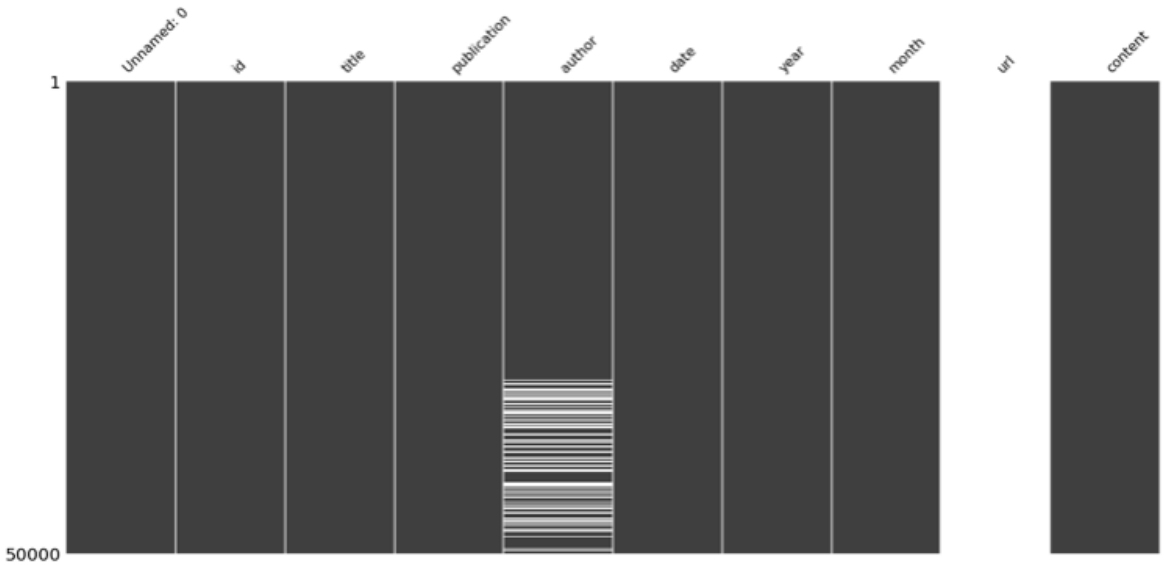

Figure 9. Missing data in the dataset

Tokenization

Tokenization is a prevalent technique in NLP; it is mainly a task of splitting a sentence into pieces, called tokens and removal of unnecessary words at the same time. Figure 10 shows an example before and after tokenization.

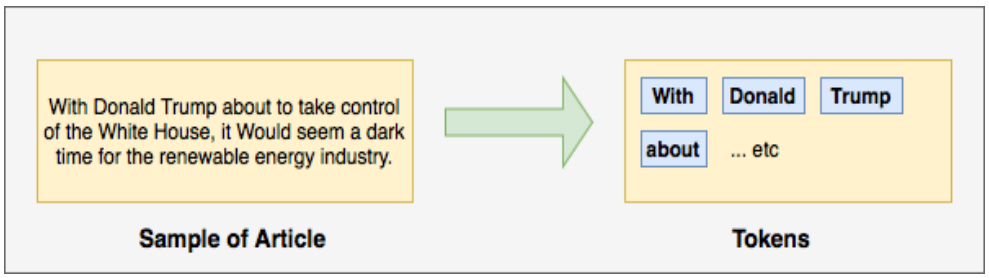

Figure 10. Example of tokenization for articles dataset

Removing Stop words

Removal of the stop words is defined by the English word class; for instance, "the", "and", "about ", that do not add meaning to articles. In this project, these words were removed because

they recur a lot in the text. In addition, the rare words that appear in the text are kept because they represent the author's characteristics and help to predict the author's identity. Figure 11 shows an example before and after removing stop words.

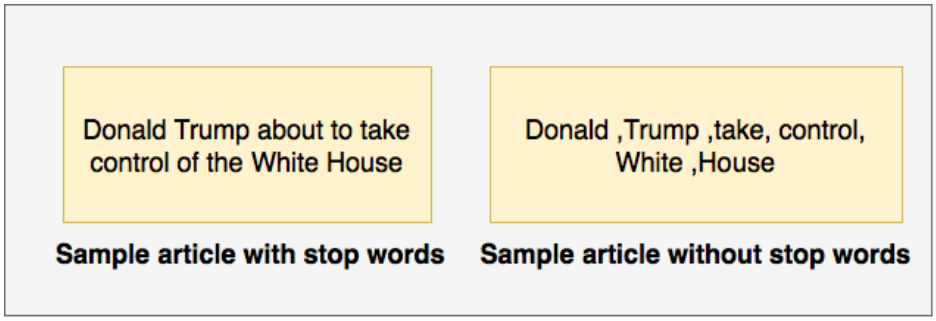

Figure 11. Removal of stop words for the article's dataset

Common words removal and Lower casing

Often, we need to remove some common words which would seem to be of less value in helping to predict the author. In this project, we removed words that were repeated in more than 70%of the articles in the dataset. All characters were adjusted and converted to lower case.

Preprocess Summarization

The important steps of preprocessing were conducted before entering the text in the machine learning algorithms. Figure 12 illustrates the steps of preprocessing in this project.

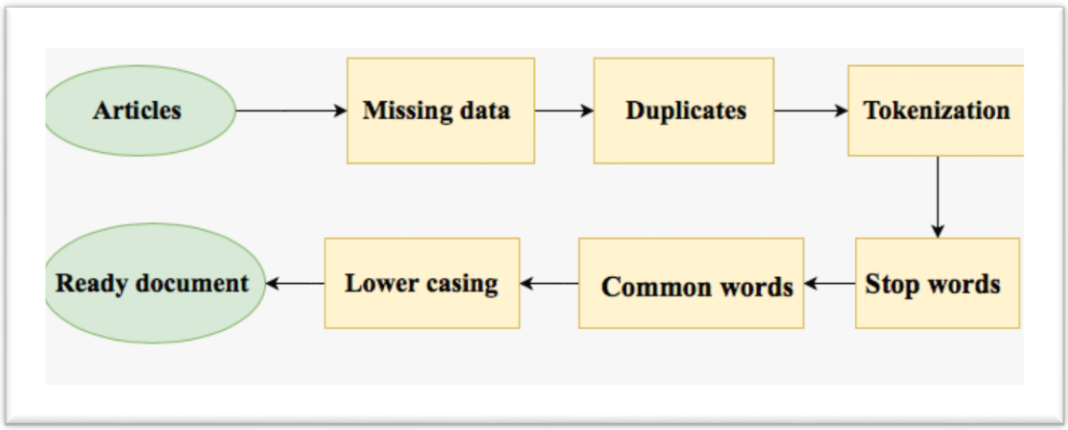

Figure 12. Summarizes pre-processing steps for the article's dataset

Feature engineering

The goal of feature extraction is to minimize the counts of features in a dataset by building up new features from the original ones. The new features can represent and summarize the data in the original features. The features extraction process used in this project is as follows:

Bag of words

Bag of words is a popular NLP technique. After converting the data to vectors, it works by counting each word occurrence. The final output is a matrix representing each word and how

much it is present in each document. Therefore, it uses the "CountVectorizer" tool in Sklearn library that has the following parameters: Tokenize; maximum features (the length of the vector: this project has specific max_features, which will be 3000); lowercase (Lower case of all text); remove stop words in the data. Consequently, the output is a matrix of 2000 x 3000 representing the top 3000 words and their count in our dataset of 2000 articles, with 295,491 stored elements in Compressed Sparse Row format. For the second scope, the output is a matrix of 5000 x 3000, representing the top 3000 words and their count in our dataset of 5000 articles, with 533,778 stored elements in Compressed Sparse Row format.

Latent Semantic Analysis

Latent Semantic Analysis is an automated, unsupervised, statistical-algebraic summarization method. Also, it is a technique that works by identifying thematic components present in the text instead of counting words. LSA helps to find unobserved semantic relations between words and sentences of the articles. Therefore, the "TruncatedSVD" tool in the Sklearn library was used to reduce the dimensions to 100. The output here is a matrix of 2000 x 100, representing 100 reduced dimensions and our 2000 articles. The output for the second scope is a matrix of 5000 x 100 representing 100 reduced dimensions and our 5000 articles.

Modelling

Data splitting

The dataset was split before modelling for testing into 70% training, 15% testing and 15% validation. It was trained only using the 70% part, and all metrics were computed using the testing and validation parts. The total data size of articles was 2000 for the first scope, divided as follows:1400 Training, 300 Testing, 300 Validation. For the second scope, the total data size of articles was 5000, divided as follows: 3500 Training, 750 Testing, 750 Validation.

Logistic regression

Logistic regression converts its products using the logistic sigmoid function to a probability that can then be mapped to more discrete classes. In the Logistic regression, the model was first applied with Bag-of-words, using the "CountVectorizer" tool mentioned in section 4.6.1. Second, Latent Semantic Analysis was conducted, using the "TruncatedSVD" function mentioned in section 4.6.2. Finally, the model was applied to the combination of the BOW and LSA matrix. To illustrate, the logistic regression model parameter C (inverse of regularization strength) was assigned the value of 20 after trying C=5, 10, 20, 40.

Random Forest

Random Forest is a classification algorithm that works by making several independent Decision Trees. Then it classifies by taking votes from those trees. The model was first applied with Bag-of-words, using the "CountVectorizer" tool. Second, it was applied with Latent Semantic Analysis, using the "TruncatedSVD" function. Then, the model was applied to the combination of the BOW and LSA matrix. The random forest parameters were as follows: 1) “n_estimators”, which is the number of trees in the forest and this project tried [50,100,400]. 2) “Min_samples_split”, which is the minimum number of samples required to split an internal node and [2,5,8,] were tried. 3) “criterion” is the function to measure the quality of a split, and the supported criteria are: "gini" for Gini impurity and "entropy" for information gain. Given the above, the best results were with n_estimators = 400, min_samples_split= 5, criterion="gini".

support vector machine

A support vector machine is a classification algorithm that works by trying to find a decision boundary between two classes by maximizing the distance. In order to use SVM in nonlinear regression, a Kernel trick is used to take low dimensional input space and convert it into high dimensional space. In this project, the model was applied for BOW, LSA and for the matrices margin between BOW and LSA. The parameter is as follows: 1) "C parameter", trying C= [1.0,2.0,4,8.0,12]; 2) "kernel", which specifies the kernel type to be used in the algorithm, trying ['rbf,' 'sigmoid,' 'poly']; 3) Degree, which is the degree of the polynomial kernel function ('poly'), trying [2,3,4,6,8]. Given the above, the best results were with C= 12, degree = 2, kernel = rbf.

Bidirectional Encoder Representations from Transformers (BERT)

To apply BERT or to use it for inference, specific data processing steps must be carried out:

- BERT Input needs a specific format, and the datasets are usually built to have the following features:
  - guide: A solitary id that shows an observation.
  - text_a: The text that is sorted into given categories.
  - text_b: Text that is applied when training an algorithm to understand the relationship between sentences.
  - Label: It is the classes or categories for a given text.
- Apply text processing, for instance: normalizing the articles by changing all whitespace characters to spaces; tokenizing the articles or splitting the sentence into the token(words); adding CLS and SEP tokens to recognize the beginning and the end of a sentence; dividing words into pieces of a word, based on similarity, for instance, "calling" becomes "call" + "ing"; applying BERT's vocabulary to map the words in the text to indexes, which are stored in BERT's vocab.txt file, by building a tokenizer object that collects a list of sentences and uses BERT's pre-trained algorithm to process the text.

The dataset was presented and explored for more understanding. Second, the pre-processing procedure was applied to prepare the dataset for entry to the algorithms. The pre-processing procedure is: checking for missing data and duplicates, tokenization, removing stop words, common words removal and converting the characters to lower case. Next, the feature selection techniques were applied: Bag-of-words, Latent Semantic Analysis and the two features merged together. Finally, all algorithms were applied, namely: logistic regression, support vector machine, random forest and BERT.

RESULTS

The results that were reached in this project. First, the various experiments will be described, then the results will be explored of each feature extraction that was applied with all the algorithms that were used in this project; namely, logistic regression, random forest, support vectors machine and BERT. The report will assess which feature analysis gave a better performance; the bag of words and latent semantic analysis. Furthermore, the results of combining the two features will be explored. Ultimately, the best feature analysis and model for predicting the author's identification will be discovered.

It can be concluded that, after applied all the experiments using several algorithms and a set of feature extraction techniques, comparing BOW and LSA, the BOW achieves the best

performance results with the logistic regression algorithm at 74%. In contrast, convolutional LSA feature extraction with support vector machine and random forest provided the lowest results. Table 3 illustrates the use of the test data of the Articles dataset and the algorithms that provided the top outcome based on the features for the author identification task. Ultimately, BERT achieved 70.33% accuracy and 0.89 log loss.

Table 3. Illustrates All algorithms and features Extraction Results for Articles Dataset

Algorithm	Feature	Accuracy %	Recall %	Precision %	F1 Score %
RF	**BOW**	72	72	73.3	72
	LSA	58	58	56.55	58
SVM	**BOW**	70	70	70.87	70
	LSA	58.33	58.33	59.85	58
LR	**BOW**	74	74	74.20	74
	LSA	65	65	64.63	65
BERT	**BERT**	70.33	66.56	70.40	67

Figure 13 illustrates a comparison of the test accuracy results of all three algorithms and the two feature extractions methods, LSA and Bag of Words. Moreover, the best accuracy is seen from logistic regression using Bag of Words, followed by random forest. In all algorithms, using Bag of Words, scored better than LSA.

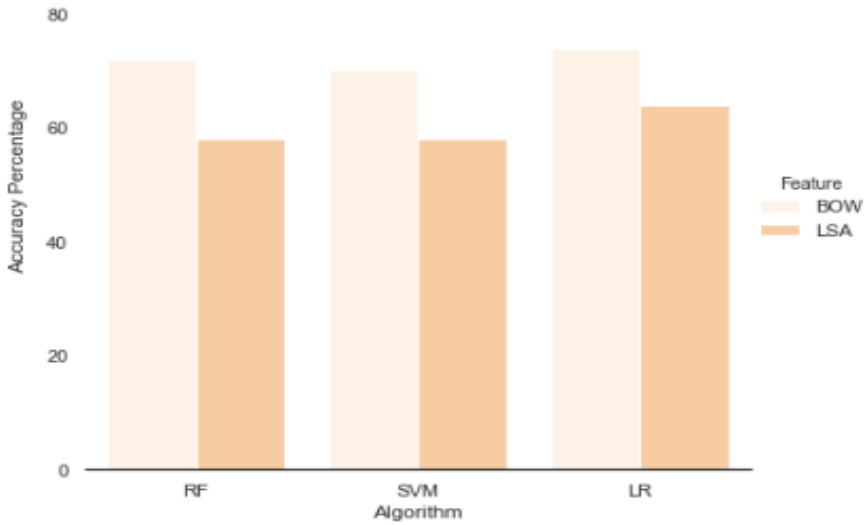

Figure 13. The Explanation That BOW Achieved Better Accuracy Results Compared to LSA Features Extraction.

After applying all the experiments on the second scope, using several algorithms and a set of feature extraction techniques. Among all, and for comparison between BOW and LSA, the BOW achieves better performance results with the logistic regression algorithm at 93.86%. Table 4 Illustrates the use of the test data of the Articles dataset and the algorithms that provided the top outcome based on the features for the author identification task. Moreover, in all algorithms, Bag of Words scored better than LSA. The best accuracy performance is with LR at 94.9% when the features were merged and it proved that it was better than BOW and LSA individually. Ultimately, BERT achieved results of 86.56% accuracy performance and 0.51log loss.

Table 4. Illustrates All algorithms and Features Extraction Results for Articles Dataset

Algorithm	Feature	Accuracy %	Recall %	Precision %	F1 Score %
RF	BOW	92.66	95.2	92.66	93.87
	LSA	73.2	76.13	73.2	74.01
SVM	BOW	92	95.6	92	92.43
	LSA	78.66	83.86	78.66	79.27
LR	BOW	93.86	96.13	93.86	94.10
	LSA	82.53	84.62	82.53	82.70
BERT	BERT	86.56	85.19	88	86

Various machine learning algorithms were presented and evaluated; namely, BERT, Logistic Regression, SVM, and RF with BOW and LSA feature extraction. Given the above, with the Articles dataset, the results show that the logistic regression algorithm with BOW achieved the best result in the first scope compared with the other models, with 74% performance accuracy. In the second scope (after enhancement), the BOW achieves the best performance results with the logistic regression algorithm at 93.86%. Moreover, in all algorithms, Bag-of-Words scored better than LSA. The best accuracy performance is with LR at 94.9%, when the features were merged and it proved that it was better than BOW and LSA applied individually, with an improvement by almost 0.1 compared with BOW only at 0.1%. Finally, in the second scope, BERT improved significantly, due to the increase in data and the decrease in the number of authors.

CONCLUSIONS

Author identification works to preserve intellectual property rights and prevent theft of articles, attributing each article to its primary author. It authorizes the governments or institutions to give author identification credit where credit is due. The "All the news" dataset on Kaggle was used for experimentation, and it requires suitable preparation before being used in machine learning algorithms. For instance, check the missing data; check the duplicated data; tokenization; remove stop words; remove too familiar words; and adjust all characters to lower case. In addition, BOW and LSA were compared to find which gave the best performance for

the task of author identification. This research studied and analyzed several types of classifiers; BERT, logistic regression, support vector machine, and random forest. The models were applied after using the BOW and LSA feature extraction approaches.

In the first scope, the models were evaluated with 20 authors and 2000 articles, and the best accuracy was seen by logistic regression using Bag-of-Words (74%), followed by random forest (72%), also using Bag-of-Words. BERT achieved 70.33% accuracy performance and 0.89 log loss. Next, all the experiments were repeated in a second scope, with 10 authors and 5000 articles. On this scope, the BOW achieves better performance results with the logistic regression algorithm at 93.86%. Moreover, in all algorithms, Bag-of-Words scored better than LSA. The best accuracy performance was with LR at 94.9% when the BOW and LSA feature extraction methods were merged. Ultimately, BERT achieved results of 86.56% accuracy and 0.51 log loss.

Limitation and future work

In this research, several models were implemented with a set of feature extraction methods. However, there are more models that can be applied for the comparison between BOW and LSA; for instance, the Naive Bayes and KNN models. Then, the accuracy result can be compared to the result achieved in this research. Also, it is possible in future to expand the scope to more than 20 authors, with more than 10,000 articles.

The study has some limitations: the dataset is new, and there is no previous work on it; there is a dataset for articles like [11], but not the same dataset. The BERT model is relatively new, so there are few papers on it and no paper about author identification by applying BERT. Furthermore, to increase the performance and reduce time consumption, Google colab has been used, but the speed is still the same and it sometimes takes 9 hours to run the code, as in BERT.

References

[1] T. R. R. Raju Dara, "Authorship Attribution using Content based Features and N-gram features," International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, no. 1, pp. 1152-1156, 2019.
[2] D. A. K, S. S. K. Satyam Anand, "A Statistical Analysis Approach to Author Identification Using Latent Semantic Analysis," Notebook for PAN at CLEF, pp. 1143-1147, 2014.
[3] R. U. K. M. Barathi Ganesh H B, "Author identification based on word distribution in word space," in 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 28 September 2015.
[4] I. S. B. S. R. Waheed Anwar, "Design and Implementation of a Machine Learning-Based Authorship Identification Model," Hindawi Scientific Programming, vol. 2019, pp. 1-15, 2019.
[5] H. T. Z. R. Qian Chen, "Deep Learning based Authorship Identification," Department of Electrical Engineering, Stanford, CA, pp. 1-9, 2017.
[6] N. M. E.-M. G. Ahmed M. Mohsen, "Author Identification Using Deep Learning," in 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 2016.
[7] X. W, Z. N, W. D. Chunxia Zhang, "Authorship identification from unstructured texts," Knowledge-Based Systems, pp. 99-111, 2014.
[8] A. G. D. L. A. F. Y. David Madigan, "Author Identification on the Large Scale," vol. 13, 2005.

[9] E. Stamatatos, "Author identification: Using text sampling to handle the class imbalance problem," Information Processing & Management, vol. 44, pp. 790-799, 2008.
[10] D. Ms.Smita Nirkhi, "Stylometric Approach For Author Identification of Online Messages," International Journal of Computer Science and Information Technologies, vol. 5, pp. 6158-6159, 2014.
[11] A. K. Hassan Alam, "Multi-lingual author identification and linguistic feature extraction — A machine learning approach," in 2013 IEEE International Conference on Technologies for Homeland Security (HST), Waltham, MA, USA, 2014.
[12] A. K. G. Tufan TAŞ, "Author Identification for Turkish Texts," Çankaya Üniversitesi Fen-Edebiyat Fakültesi, Journal of Arts and Sciences, pp. 151-161, 2007.
[13] W. D. Kim Luyckx, "Authorship attribution and verification with many authors and limited data," in Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, 2008.
[14] E. J. a, L. O. b. W. Oliveira Jr. a, "Comparing compression models for authorship attribution," Forensic Science International, pp. 100-104, 2013.
[15] V. A. O. Ignatius Ikechukwu Ayogu, "Authorship Attribution using Rough Sets based Feature Selection Techniques," International Journal of Computer Applications, vol. 152, pp. 38-46, 2016.
[16] A. a. S. S. Adhikari, "Author Identification: Using Text Mining, Feature Engineering \& Network Embedding," pp. 1-14.
[17] E. v. d. B, R. Julian Hitschler, "Authorship Attribution with Convolutional Neural Networks and POS-Eliding," Proceedings of the Workshop on Stylistic Variation, p. 53–58, 2017.
[18] I. T. S. W. Marcelo Luiz Brocardo, "Authorship verification for short messages using stylometry," in Although the dataset is small in [], the predictive accuracy performance, which is measure by using k-fold cross-validation, is low when using Bayesian Classifier. They proved that the accuracy decreases if the dataset volume decreases. Also, as the dat, Athens, Greece, 2014.
[19] N. M. Sarwat Nizamani, "CEAI: CCM-based email authorship identification model," Egyptian Informatics Journal, pp. 239-249, 2013.
[20] H. B. B. C. F, M. D. Farkhund Iqbal, "A unified data mining solution for authorship analysis in anonymous textual communications," Information Sciences, pp. 99-112, 2011.
[21] C. a. S. W. a. L. L. a. D. C. a. Z. X. Zhao, "Research on Author Identification Based on Deep Syntactic Features," 2017 10th International Symposium on Computational Intelligence and Design (ISCID), pp. 276-279, 2017.
[22] F. S. T. Richmond Hong Rui Tan, "Authorship Identification for Online Text," in 2010 International Conference on Cyberworlds, Singapore, Singapore, 2010.
[23] C. O, D. V. D, L. S. Castillo Esteban, "Unsupervised Method for the Authorship Identification Task," CLEF, vol. 1180, pp. 1035-1041, 2014.
[24] G. K. Mikros, "Authorship attribution and gender identification in Greek blogs," Methods and Applications of Quantitative Linguistics, vol. 21, pp. 21-32, 2012.
[25] D. R. V. Nirkhi Smita, "Comparative study of authorship identification techniques for cyber forensics analysis," arXiv preprint arXiv:1401.6118, vol. 4, pp. 32-35, 2013.
[26] D. b, D. R. C, D. B. S. Naga Prasad a, "Influence of lexical, syntactic and structural features and their combination on Authorship Attribution for Telugu Text," in International Conference on Intelligent Computing, Communication & Convergence, Bhubaneswar, Odisha, India, 2015.
[27] Y. M. Yuta Tsuboi, "Authorship Identification for Heterogeneous Documents," 2002.

[28] D. S. K. Mousmi Chaurasia, "NATURAL LANGUAGE PROCESSING BASED INFORMATION RETRIEVAL FOR THE PURPOSE OF AUTHOR IDENTIFICATION," International International Journal Journal of Information of Information Technology Technology & Management Information System (IJITMIS), vol. 1, no. 0976, pp. 45-54, 2010.

[29] S. G. A. A. a. J. A. Subhajit Dey Sarkar, "A Novel Feature Selection Technique for Text Classification Using Na\ve Bayes," Hindawi Publishing Corporation International Scholarly Research Notices, vol. 2014, pp. 1-10, October 2014.

[30] D. V. V. K. O. S. R. N V Ganapathi Raju, "Author Based Rank Vector Coordinates (ARVC) Model for Authorship Attribution," I.J. Image, Graphics and Signal Processing, pp. 68-75, 2016.

[31] Y. P. Kaiz Merchant, "NLP Based Latent Semantic Analysis for Legal Text Summarization," in 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19-22 Sept. 2018.

[32] A. Navlani, "Latent Semantic Analysis using Python," 9th October 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python.

[33] X. L. Z. H. C. D. F. N. H. H. Jie Xu, "Multi-Class Support Vector Machine via Maximizing Multi-Class Margins," Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), pp. 3154-3160.

[34] M.-W. C. K. L. K. T. Jacob Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2019.