Authorship Identification System Using Word2Vec Word Embedding Model

Kyi Pyar Zaw
University of Computer Studies, Mandalay (UCSM), Myanmar

Abstract: *Authorship identification is a field of study where researchers investigate for various approaches to identify an unknown text's author. The aim is to determine who wrote anonymous texts by providing writing examples from potential authors. Authorship identification system identifies the most possible author of texts such as articles, news, books, blogs, emails and messages, etc. Applications for this evaluation include news stories, forensic science, plagiarism detection, and responsibility for published work. Source code authorship identification has grown in significance over time due to the prevalence of online academic examinations, malware, and other types of code-based plagiarism. This system examines the authorship identification of news articles written in English. Extraction of features that indicate an author's writing style is the primary goal of the authorship identification challenge. In traditional methods of authorship identification, hand-crafted features are used to represent the text. Unlike traditional approaches, this system proposes the authorship identification by investigating the use of Word2Vec that performs automatic feature extraction. In the subject of authorship identification, deep layers of neural networks can employ word embedding to extract characteristics from them and learn the patterns of authors based on context and word co-occurrence.*

Keywords: *Authorship Identification, CBOW, Logistic Regression, Skip gram, Word Embedding, Word2Vec.*

1. Introduction

In the modern world, everything is accessible online, which encourages malevolent and criminal conduct. The identification of content that is already available in text data must therefore be resolved. Applying authorship identification, which determines the true author of anonymous work based on writing samples, solves this issue. Theoretically, the study of information trustworthiness and verification is affected by these discoveries. This study aims to identify the elements that influence people's inclination to check news authorship in addition to looking at the effects of news authorship verification. To be clear, news brands and journalists/reporters are both considered news authors in this study [1].

When it comes to the disputed authorship of some literary works, author identification is a critical issue. The problem is also well-known in the digital forensics community. Several approaches, including those from machine learning and statistic-specific computation, have been proposed to address this issue. The training phase typically kicks off the author identification procedure. Stop words are eliminated from used texts by well-known authors throughout the training phase. The verification phase comes next, where machine learning methods are used to compare the unattributed texts to the previously computed text. The author that most closely fits is then chosen from the pool of authors that are readily available. In a nutshell, author identification is comparing a writer's writing style to a corpus of texts (the text body used for linguistic analysis) by writers whose identities are known. It is important to confirm the authorship of the news because when a variety of news pieces authored by numerous authors are broadcast, fake news and low-quality news may predominate. Press credibility affected news authorship identification rather than the other way around. This detrimental influence suggests that persons who are more inclined to believe the press is trustworthy are those who are less likely to verify the authorship of news stories. This finding suggests the peril of news intake without critical thought. The correlations between press credibility and other factors were also partially mediated by news authorship verification. As a multi-class categorization problem where the authors serve as the class labels, authorship identification can be formulated [2]. The choice of categorization methods is thus the second challenge of the authorship identification task.

2. Literature Review

Numerous natural language processing (NLP) projects have made extensive use of word embedding. These embeddings carry valuable semantic information and can be pre-trained on a big corpus. A word embedding format essentially aims to represent words as vectors in a space. In actuality, this typically implies that word embeddings are placed in a high-dimensional space where they are separated from one another and like or related words are placed close together. Word embedding was the chosen method since machine learning, even deep learning, cannot comprehend strings or plain texts and needs a vectorized representation of the texts for operations. The different types of word embedding can be broadly classified into two categories: Frequency based Embedding and Prediction based Embedding.

The known word embedding techniques such as TF-IDF, Bag of Words and the relatively recently used Word2Vec are used to perform text classification and compare the results. Many studies have been undertaken on word embedding techniques, but few researchers have studied the impact of each method in text classification.

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 [3]. The BOW approach is easier to use computationally and conceptually than many other categorization strategies. Because of this, BOW-based systems were able to record new and improved performance scores on standard metrics for text and image categorization techniques [4]. Word frequency in a document is represented by TF, and frequency inside a word in a document is defined by IDF. For the analysis, words with a high TF-IDF weight are more significant than terms with a low TF-IDF weight. Word2Vec, an algorithm first put out by the Google team, led by Mikolov, has improved word embedding in text mining research [5]. The similarity between the two words can be determined following the computation of Word2Vec's weight. The Word2Vec approach is separated into Skip-gram and CBOW (continuous bag of words) [6]. For supervised classification, a variety of algorithms have been created using artificial intelligence (logical algorithms like decision trees), perceptron-based methods (single-layered perceptrons, multilayered perceptrons), and statistical learning techniques (Bayesian networks, instance-based techniques) [7]. Vapnik was the first to propose the Support Vector Machine (SVM), which has since generated a lot of interest among researchers studying machine learning. [8]. The SVM is generally capable of producing superior performance in terms of classification accuracy than the other data classification algorithms, according to a number of recent researches. It was discovered that, if the activation functions of neurons are linear, multilayer networks do not offer an improvement in processing power compared to networks with a single layer since a linear function of linear functions is likewise a linear function [9]. In Classification, different characteristics determine the class to which the unlabeled data belongs. KNN is mostly used as a classifier. It is used to classify data based on closest or neighboring training examples in a given region. This method is used for its simplicity of execution and low computation time. For continuous data, it uses the Euclidean distance to calculate its nearest neighbors. For a new input the K nearest neighbors is calculated and the majority among the neighboring data decides the classification for the new input [10].

The Naive Bayes classifier performs best in two scenarios: first, when the features are functionally dependent, and second, when they are fully independent. Between these two extremes is where the worst performance can be found [11]. In an effort to enhance the performance of the naive Bayesian classifier, numerous extensions and improvements have been made. In [12], The authors propose a Tree Augmented Naive Bayes (TAN) classifier, which outperforms the naive Bayesian classifier significantly. The authors of [13] demonstrate that while setting parameters based on maximum likelihood also produces better results, picking structures by maximizing conditional likelihood does.

Because of its simplicity, computational effectiveness, and strong performance for real-world issues, the Nave Bayes classifier has gained appeal and is being used by many. In literature, conditional Gaussian distributions for each attribute probability given the class are used to manage continuous attributes [14].

3. Authorship identification

Authorship identification is the process of identifying the original author by scrutinizing a text's characteristics and writing style. It is not a field of study that has grown as a result of rising internet usage. It was employed to identify the author of a piece of news. The text's grammatical structure and word choice are utilised in author identification research.

In traditional methods of authorship attribution, independent features such as lexical n-grams or frequency-based word embedding is used to represent the text, which are very similar to one-hot encodings. As well as the methods perform, in such approaches, the word representations are created independent of each other's meanings and words of similar contexts seem to be represented in different vector spaces, which is problematic for detecting the semantic values of words. The overview of the system is shown in Fig 1.

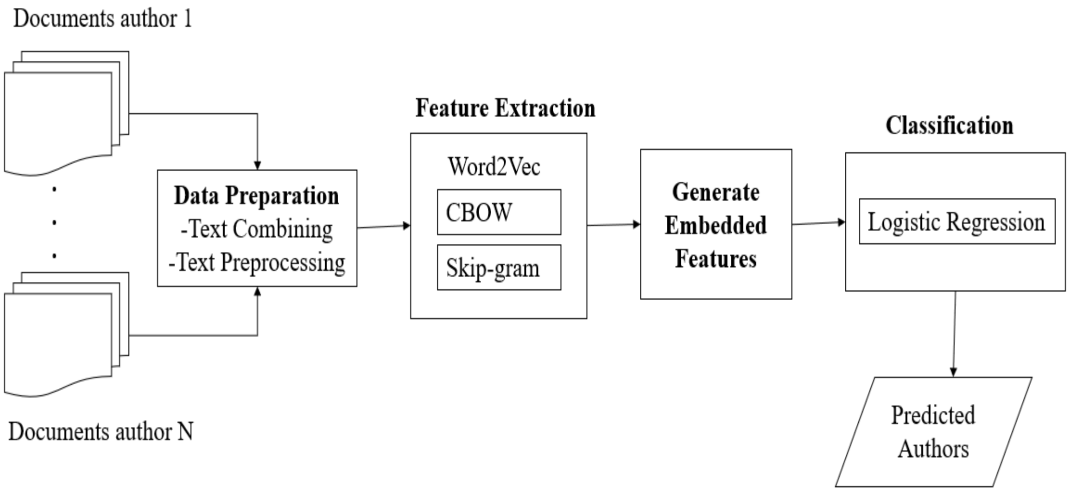

Figure 1: Flow of the system

3.1 Data Preprocessing

Any type of processing done on raw data to get it ready for another data processing operation is referred to as data preprocessing, which is a part of data preparation. It has historically been a crucial first stage in the data mining process. It changes the data's format so that data mining, machine learning, and other data science jobs can process it more quickly and efficiently.

Stop words, often referred to as function words or non-content words, are printed punctuation that are removed during pre-processing because they create useful characteristics. Every written document contains a list of tokens that are often used. Auxiliary verbs, prepositions, conjunctions, grammatical articles, and pronouns are some of those that are eliminated despite having no bearing on the classification procedure.

All of the text's punctuation is eliminated during the removal process. One of the most popular preprocessing stages is called "lowering," which involves changing the text's case, preferably to lower case. However, it is not required to complete this step each time you work on an NLP problem because lower casing can result in information loss for some situations.

While stemming the word, lemmatization ensures that its meaning is retained. A pre-defined dictionary used in lemmatization stores word context and verifies the term while decreasing.

A token is a representation of how characters are arranged in a particular document and are processed as a single unit based on their semantic relationships. Through the use of symbols and punctuation like as bullets, colons, exclamation points, hyphens, parenthesis, numbers, and semicolons, the text is broken into tiny units known as tokens and given a meaningful semantic meaning.

3.2 Word2Vec

By using word embedding, words can be represented as vectors, usually as real-valued vectors. Word embedding's primary objective is the transformation of words' high-dimensional feature space into their low-dimensional feature vectors. From the training text corpus, the word2vec model may produce numerical vector representations of words while preserving their semantic and syntactic relationships. When the term Man is removed from the word King and Woman is added, one of the most closely related results is Queen, which is a fairly well-known illustration of how word2vec keeps the semantics. The demonstration of the word to vector model is shown in Fig 2.

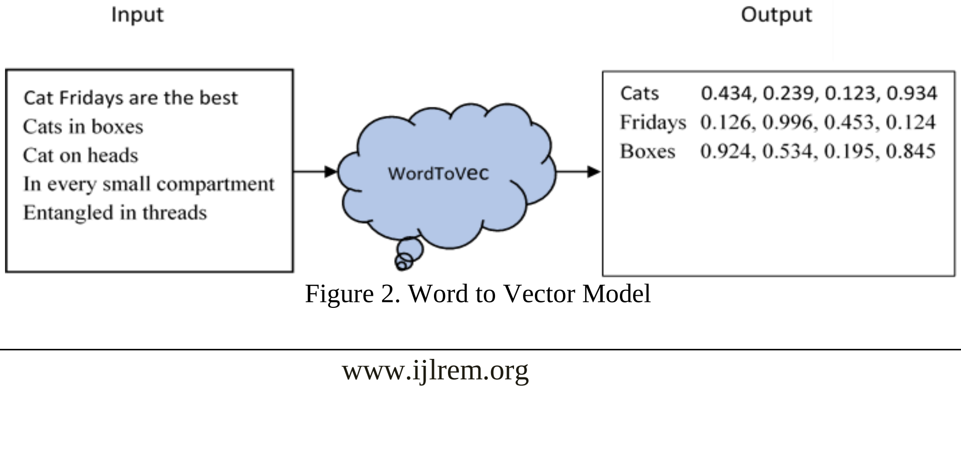
Input
Output
Cat Fridays are the best
Cats in boxes
Cat on heads
In every small compartment
Entangled in threads
WordToVec
Cats 0.434, 0.239, 0.123, 0.934
Fridays 0.126, 0.996, 0.453, 0.124
Boxes 0.924, 0.534, 0.195, 0.845

Figure 2. Word to Vector Model

Word2Vec can use one of two distinct model designs to produce these word embedding representations. These consist of

1. A target word can be predicted from context using the Continuous Bag of Words (CBOW) model.
2. The Skip-gram model, which forecasts context from the target word.

3.2.1 Continuous Bag of Words (CBOW) model

The CBOW model seeks to forecast a word's probability in light of its context. A fixed-sized window around the target word contains a bag of the contained words as the context is represented. In essence, CBOW is similar to word prediction when the context is known.

The Word2Vec family of models are unsupervised, which entails that all you need to do for them to create dense word embeddings from a corpus is to provide them a corpus without any labels or additional information. Once you have this corpus, however, you will still need to use a supervised classification algorithm to reach these embeddings. However, we shall carry out that task directly from the corpus, without the aid of any additional data. In order to forecast the target word, Y, we can now model the CBOW architecture as a deep learning classification model by using the context words as our input, X. Building this architecture is really easier than creating the skip-gram model, in which we attempt to forecast a large number of context words.

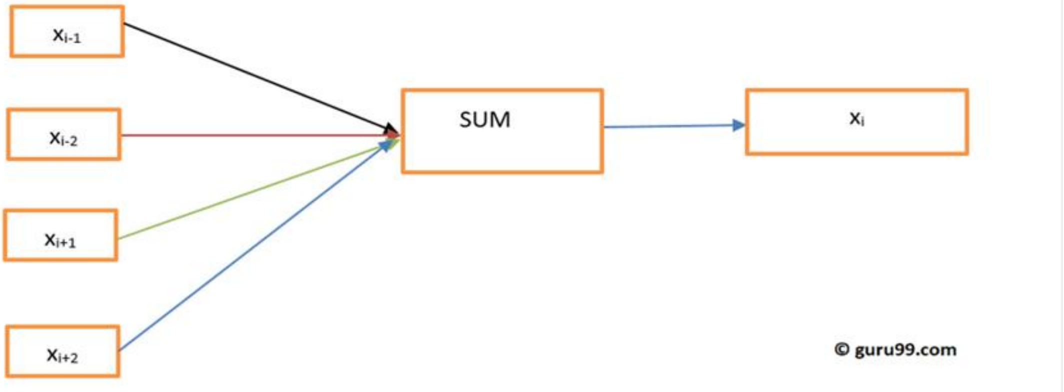

Figure 3: Continuous Bag of Word Architecture

3.2.2 Skip-gram

In reality, the Skip-Gram design is a CBOW inverted architecture. The skip gram model is essentially like guessing the context if a word is given, in contrast to the CBOW model. Here, by picking them at random, more distant terms are given less weight. Only the maximum window size can be defined when specifying the window size parameter. Actual window sizes are determined at random and range from 1 to maximum size. Both the CBOW and the skip-gram models will produce two one-hot encoded target variables and two related outputs if a context window size of 1 is specified. A final error vector is created by adding element-wise the two error vectors that are obtained by computing two different errors with respect to two target variables. Following training, the word vector representation will be determined by the weights between the input and hidden layer. The objective or loss function is essentially of the same sort as that of the CBOW model.

An architecture for calculating word embeddings is called skip-gram Word2Vec. Unlike CBOW Word2Vec, which predicts the center word using the surrounding words, Skip-gram Word2Vec predicts the surrounding words using the center word. The log probabilities of the words to the left and right of the target word are added by the skip-gram goal function.

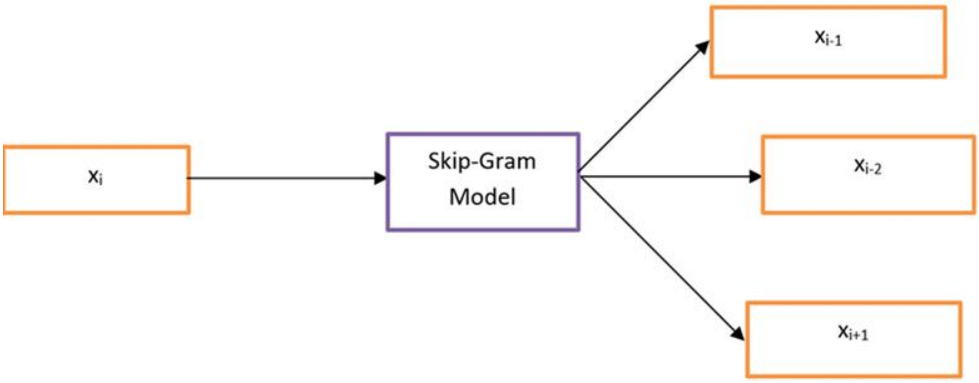

Figure 4: Skip-gram Architecture

3.3 Multinomial logistic regression

The Multinomial logistic regression is used to forecast categorical placement in or the likelihood of category membership on a dependent variable. The independent variables may either be continuous or dichotomous (i.e., binary) (i.e., interval or ratio in scale). There can be more than two categories of the dependent or outcome variable in multinomial logistic regression, which is a straightforward extension of binary logistic regression. Multinomial logistic regression, like binary logistic regression, estimates the likelihood of categorical membership via maximum likelihood estimation.

With this straightforward addition, binary logistic regression may accommodate dependent variables with more than two categories. Other names are Polychotomous Logistic, Maximum Entropy Classifier, SoftMax Regression, and Multi-class Logistic Regression. The SoftMax function is used in multinomial logistic regression to compute probabilities.

$$Softmax(z_i) = \frac{\exp(z)}{\sum_{j=1}^{k} \exp(z_j)}, \text{fori} = 1, \ldots, \text{K} \qquad (1)$$

Where, $\boldsymbol{z_i}$ represents the i^{th} element of the input to SoftMax, which corresponds to class *i*, and *K* is the number of classes. The result is a vector containing the probabilities that sample *x* belongs to each class. The output is the class with the highest probability.

3.4 Gaussian Naïve Bayes

Non-maximum A straightforward probabilistic classifier based on the Bayes theorem is the naive Bayes classifier. Every feature variable is treated as an independent variable by naive Bayes. This classifier can be taught in supervised learning quite effectively and used to challenging real-world scenarios. Naive Bayes' key benefit is that it just needs a little quantity of the training data required for classification. The Naive Bayes classifier for text classification estimates the likelihood of each class based on the feature vector.

Gaussian Naïve Bayes is applied when the predictor values are continuous and are predicted to follow a Gaussian distribution. When dealing with continuous data, a usual assumption is that the continuous values associated with each class are distributed according to Gaussian distribution. The mean and variance of each class are obtained after class-segmenting the training data. As a result, the method below can be used to estimate the probabilities of a continuous dataset.

$$P(v_j|c_i) = \frac{1}{\sqrt{2\pi\sigma_{ji}}}\, e\left(-\frac{(v_j - \mu_{ij})^2}{2\sigma_{ji}^2}\right) \qquad (2)$$

Where, v is the variable, c is the class,μ is the mean of the Gaussian and σ is the variance.

3.5 Multilayer Perceptron

Multilayer Perceptron models can be used for deep learning because they are completely connected networks. They are utilized for activities and issues that are more difficult, including speech recognition or sophisticated classification. Processing and model maintenance can be resource- and time-intensive due to the depth and complexity of the model. A feed forward neural network supplement is the multilayer perceptron. There are three different kinds of layers in it: an input layer, an output layer, and a concealed layer. The logistic function is used for binary classification.

$$g(z) = \frac{1}{(1+e^{-z})} \qquad (3)$$

4. Dataset Description

This system uses the C50 (Reuter_50_50) dataset. The C50 dataset is a subset of the well-known Reuters Newswire corpus, RCV1. It is a database of 5000 news articles written by 50 different authors. The dataset contains two attributes: author name and text. The author name contains 50 attribute specifies the text written by them. The text attribute contains the unprocessed text. The common writing style (journalism) and subject matter of this corpus make it fascinating. Because each class, which represents an author, has an equal number of instances, the dataset is well balanced. With an average word count of 500, the training corpus and test corpus each contain 2,500 documents (50 documents for each author in each corpus).

Table 1: Standard variables

Language	No. of Authors	No. of Documents	Training	Testing	Language
English	50	5000	2500	2500	English

Table 2: Presents the training samples of the input document. Each training sample has its' context and target words.

Table 2: Training Samples

Training Samples	**Context**	**Target**
1	(investment, watchdog)	britain
2	(britain, watchdog, thursday)	investment
3	(britain, investment, thursday, punish)	watchdog
4	(investment, watchdog, punish, company)	thursday
5	(watchdog, thursday, company, robert)	punish
6	(thursday, punish, robert, fleming)	company
7	(punish, company, fleming, group)	robert
8	(company, robert, group, rule)	fleming
9	(robert, fleming, rule, breach)	group
10	(fleming, group, breach, fine)	rule
11	(group, rule, fine, total)	breach
12	(rule, breach, total, pound)	fine
13	(breach, fine, pound, million)	total
14	(fine, total, million)	pound
15	(total, pound)	million

5. Experimental Results

The experiments on CBOW and Skip-gram models are made by splitting the dataset into 80 percent training (4000 samples) and 20 percent testing (1000 samples). In this system, the results are evaluated using various values of vector size (5, 10, 15), window size (5, 10, 15) and random words (5 to 9). The results are shown in Table 3, Table 4 and Table 5 using Multinomial Logistic Regression classifier, Naïve Bayes classifier and multilayer perceptron respectively.

Table 3: Experimental Results using Multinomial Logistic Regression Classifier

Vector Size	**Window Size**	**Random Words**	**Accuracy**	
			CBOW	**Skip-gram**
5	5	5	77%	47%
10	10	5	99%	85%
15	15	5	100%	98%
5	5	6	77%	47%
10	10	6	99%	87%
15	15	6	100%	99%
5	5	7	82%	50%
10	10	7	99%	88%
15	15	7	100%	99%
5	5	8	83%	54%
10	10	8	99%	89%
15	15	8	100%	99%
5	5	9	84%	53%
10	10	9	100%	90%
15	**15**	**9**	**100%**	**100%**

According to Table 3, the results of CBOW is better than skip-gram and vector-size (15), window-size (15) and random words (9) achieves the best accuracy result.

Table 4: Experimental Results using Naïve Bayes Classifier

Vector Size	**Window Size**	**Random Words**	**Accuracy**	
			CBOW	**Skip-gram**
5	5	5	40%	45%
10	10	5	82%	84%
15	15	5	95%	96%
5	5	6	43%	46%
10	10	6	85%	87%
15	15	6	97%	97%
5	5	7	45%	53%
10	10	7	88%	90%
15	15	7	97%	98%
5	5	8	50%	57%
10	10	8	88%	92%
15	15	8	98%	98%
5	5	9	53%	58%
10	10	9	88%	92%
15	**15**	**9**	**99%**	**99%**

According to Table 4, the results of skip-gram is better than CBOW and vector-size (15), window-size (15) and random words (9) achieves the best accuracy result.

Table 5: Experimental Results using Multilayer Perceptron

Vector Size	**Window Size**	**Random Words**	**Accuracy**	
			CBOW	**Skip-gram**
5	5	5	81%	55%
10	10	5	99%	90%
15	15	5	100%	98%
5	5	6	82%	57%
10	10	6	100%	92%
15	15	6	100%	98%
5	5	7	86%	62%
10	10	7	100%	94%
15	15	7	100%	98%
5	5	8	86%	62%
10	10	8	100%	95%
15	15	8	100%	99%
5	5	9	88%	63%
10	10	9	100%	95%
15	**15**	**9**	**100%**	**99%**

According to the Table 5 that used multilayer perceptron, the results of CBOW is better than skip-gram and vector-size (15), window-size (15) and random words (9) achieves the best accuracy result 100%.

6. Conclusion

The system proposes a framework for authorship identification problems of news articles with large training samples. In this system, three classifiers such as multinomial logistic regression, Gaussian naïve bayes and Multilayer perceptron are used to classify the authors. For multinomial logistic regression classifier, CBOW model is better than skip-gram model. For the Gaussian naive bayes classifier, the skip-gram model is better. For the Multilayer perceptron classifier, the CBOW model is better than the skip-gram model. When the window size is increased beyond the vector size, the accuracy of the CBOW model increases and the accuracy of the skip-gram model decreases. When the vector size is increased over the window size, both the accuracy of the CBOW model and the accuracy of the skip-gram model increase. When comparing the three classifiers, multilayer perceptron classifier. In continuous-bag-of-words and skip-gram comparison, continuous-bag-of-words with multilayer perceptron classifier is better than skip-gram with multilayer perceptron classifier. Test results using Word2Vec with the multilayer perceptron classifier for C50 dataset show that the CBOW model is better than the Skip-gram model.

References

[1] S. Choi, “Determinant and Consequence of Online News Authorship Verification: Blind News Consumption Creates Press Credibility,” p. 23, 2019.

[2] E. Stamatatos, “AUTHORSHIP ATTRIBUTION BASED ON FEATURE SET SUBSPACING ENSEMBLES,” Int. J. Artif. Intell. Tools, vol. 15, no. 05, pp. 823–838, Oct. 2006, doi: 10.1142/S0218213006002965.

[3] Y. Zhang, R. Jin, and Z.-H. Zhou, “Understanding bag-of-words model: a statistical framework,” Int. J. Mach. Learn. Cybern., vol. 1, no. 1–4, pp. 43–52, Dec. 2010, doi: 10.1007/s13042-010-0001-0.

[4] W. A. Qader, M. M. Ameen, and B. I. Ahmed, “An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges,” in 2019 International Engineering Conference (IEC), Erbil, Iraq, Jun. 2019, pp. 200–204. doi: 10.1109/IEC47844.2019.8950616.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” p. 9.

[6] X. Rong, “word2vec Parameter Learning Explained,” p. 22.

[7] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques,” p. 20.

[8] D. K. Srivastava and L. Bhambhu, “DATA CLASSIFICATION USING SUPPORT VECTOR MACHINE,” p. 8, 2005.

[9] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, “Multilayer Perceptron and Neural Networks,” vol. 8, no. 7, p. 11, 2009.

[10] K. Taunk, S. De, S. Verma, and A. Swetapadma, “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification,” in 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, May 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.

[11] I. Rish, “An empirical study of the naive Bayes classifier,” p. 7.

[12] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian Network Classifiers,” Mach. Learn., vol. 29, no. 2/3, pp. 131–163, 1997, doi: 10.1023/A:1007465528199.

[13] D. Grossman and P. Domingos, “Learning Bayesian network classifiers by maximizing conditional likelihood,” in Twenty-first international conference on Machine learning - ICML ’04, Banff, Alberta, Canada, 2004, p. 46. doi: 10.1145/1015330.1015339.

[14] C. Bustamante, L. Garrido, and R. Soto, “Comparing Fuzzy Naive Bayes and Gaussian Naive Bayes for Decision Making in RoboCup 3D,” in MICAI 2006: Advances in Artificial Intelligence, vol. 4293, A. Gelbukh and C. A. Reyes-Garcia, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 237–247. doi: 10.1007/11925231_23.