applied sciences

MDPI

Article

GenCo: A Generative Learning Model for Heterogeneous Text Classification Based on Collaborative Partial Classifications

Zie Eya Ekolle * and Ryuji Kohno

Department of Electrical and Computer Engineering, Yokohama National University, Yokohama 240-8501, Japan
* Correspondence: zie-ekolle-cj@ynu.jp

Abstract: The use of generative learning models in natural language processing (NLP) has significantly contributed to the advancement of natural language applications, such as sentimental analysis, topic modeling, text classification, chatbots, and spam filtering. With a large amount of text generated each day from different sources, such as web-pages, blogs, emails, social media, and articles, one of the most common tasks in NLP is the classification of a text corpus. This is important in many institutions for planning, decision-making, and creating archives of their projects. Many algorithms exist to automate text classification tasks but the most intriguing of them is that which also learns these tasks automatically. In this study, we present a new model to infer and learn from data using probabilistic logic and apply it to text classification. This model, called GenCo, is a multi-input single-output (MISO) learning model that uses a collaboration of partial classifications to generate the desired output. It provides a heterogeneity measure to explain its classification results and enables a reduction in the curse of dimensionality in text classification. Experiments with the model were carried out on the Twitter US Airline dataset, the Conference Paper dataset, and the SMS Spam dataset, outperforming baseline models with 98.40%, 89.90%, and 99.26% accuracy, respectively.

Keywords: natural language processing; text classification; probabilistic models; machine learning; generative learning; collaborative learning; explainable AI

Citation: Ekolle, Z.E.; Kohno, R. GenCo: A Generative Learning Model for Heterogeneous Text Classification Based on Collaborative Partial Classifications. *Appl. Sci.* **2023**, *13*, 8211. https://doi.org/10.3390/app13148211

Academic Editors: Julian Szymanski and Ahmed Rafea

Received: 29 May 2023
Revised: 9 July 2023
Accepted: 12 July 2023
Published: 14 July 2023

1. Introduction

There has been an increase in human interactions over recent years due to the rise in globalization [1]. Coupled with the increase in dependency on electronic communication, the amount of electronic data generated has multiplied each day. This electronic data can be in different modalities, such as sound, image, video, or text.

Textual communication has always been one of the predominant methods of communication in human society since the invention of writing by different cultures around the world [2]. Added to the increase in human interactions in recent years, a large amount of textual information is generated daily from different sources, such as web-pages, blogs, emails, social media, and articles.

To understand the content of textual information, many language analysis tasks, such as lexical (or morphological) analysis, syntax analysis (or parsing), semantic analysis, topic modeling, and text classification, can be performed on the text corpus. In this paper, we focus on text classification and provide a machine-learning solution to automate the classification of textual information.

Text classification consists of assigning a sentence or document to an appropriate predefined category [3]. This category can involve topic, sentiment, language, or all. So, text classification tasks may include news classification, emotion classification, sentiment analysis, citation intent classification, spam classification, and so on. This is important in many institutions for creating archives and the organization of large amounts of text needed for effective planning and decision-making on their projects.

In general, depending on the type of category, text classification can be divided into topic classification, sentiment classification (or analysis), language classification, and hybrid classification based on any two or all three of these categories.

The pipeline of a general text classification task in NLP is presented in [4]. The initial step in this pipeline is text preprocessing, where the text corpus is processed for case harmonization, noise removal, tokenization, stemming, lemmatization, normalization, feature extraction, and vectorization. After preprocessing, the vectorized features are then fed into a classification algorithm that outputs a prediction of the category which defines the given text corpus.

During preprocessing, case harmonization involves setting all text in the corpus to the same case, that is, either lowercase or uppercase. Noise removal involves the removal of stop words and special characters from the corpus. Tokenization involves splitting the text into small chunks of words or sentences, called tokens. Stemming involves reducing words to their root or base form. Lemmatization involves breaking a word down to its root meaning to identify similarities.

Furthermore, normalization is the mapping of different, but semantically equivalent, phrases onto a single canonical representative phrase. It can be divided into morphological normalization, such as stemming and lemmatization, lexical normalization, such as spelling correction, syntactic normalization, such as grammatic correction, and semantic normalization, such as synonyms elimination. Since stemming and lemmatization are types of morphological normalization techniques, the normalization step usually involves lexical, syntactic, and/or semantic normalization.

Feature extraction involves the selection of features required for training and classification. Vectorization involves converting selected features from a text corpus into numerical vectors. Different types of vectorization techniques are used in NLP, such as Bag-of-Words, Term Frequency–Inverse Document Frequency (TF-IDF), Word2Vec, and Continuous Bag-of-Words (CBOW) vectorizations.

The importance of text classification has led to the development of many algorithms to automate the process [5]. Such algorithms may use either a deterministic, stochastic, or hybrid approach to infer the category of a text corpus. The advancement in machine learning algorithms as a stochastic logical operation has increased the use of the stochastic approach in text classification.

Machine learning algorithms can broadly be divided into symbolic (mostly rule-based), statistical (mostly data-driven and discriminative), and probabilistic (mostly generative) models. Each of these models can be further divided into supervised, semi-supervised, unsupervised, self-supervised, and reinforcement learning. Furthermore, based on their number of input and output variables, they can be classified as single-input single-output (SISO), multiple-input single-output (MISO), single-input multiple-output (SIMO), and multiple-input multiple-output (MIMO) models. Multiple output (MO) models are also called multitarget models, which include multilabel models for classification tasks. In this paper, we use a MISO-based generative supervised learning model for text classification.

A major challenge in text classification algorithms is the classification of a heterogeneous text corpus (or corpora) [6,7]. Heterogeneous text corpuses (or corpora) are those with latent relationships between their features (i.e., vocabularies), and their classification is challenging. The vocabularies of a text (i.e., a document) in a corpus or a corpus in corpora discussing housing prizes and sports will be explicitly unrelated, but the understanding of any implicit (or hidden) relationship between their vocabularies can help in their classification. The degree of such a heterogeneous relationship between features may vary and will influence the classification of a text corpus (or corpora).

This degree of heterogeneity between features of the same corpus or different corpora is different from the similarity measure between documents, which is estimated using cosine similarity and hamming distance measures, for example. The heterogeneity measure used in this study focuses on capturing the correlation in terms of the probabilistic dissimilarity between features in the same corpus or different corpora, while conventional

similarity measures focus on capturing similarity between documents in the same corpus or different corpora.

Furthermore, the number of vectorized features in most text classification tasks is very large such that conventional text classification models cannot perform very well due to the curse of dimensionality. A common technique to solve this problem is to use an n-gram language model, where $n > 1$. The downside of this technique is that it leads to a sparsed vectorization, which is less useful in generating a reliable classification result, especially when n increases. Also, many preprocessing operations, such as defining the count limits of feature occurrences, are performed to reduce unwanted features.

These are the challenges we seek to solve in this paper, and we do so using a collaborative learning model which takes into account the relationship between the features of a text corpus and their dimensionality, as we shall explain in Section 2.2.

Related Works

In the field of NLP, much research has been undertaken to provide solutions for text classification.

Zhang et al. [8] presented a text classifier using Naive Bayes. They applied their model to spam filtering by using preclassified emails as prior knowledge to train their model. Their model was able to detect if an email was spam or not spam. Also, Shuo [9] proposed a Gaussian Naive Bayes model for text classification after proving, using 20 newsgroups and WebKB datasets, that the Gaussian Naive Bayes model was better than its classical counterpart.

Mitra et al. [10] presented a text classifier using the least square support vector machine (LS-SVM) on a corpus of 91,229 words from the University of Denver's Penrose Library catalog. Their proposed LS-SVM is based on a Gaussian radial basis function (GRBF) kernel, which uses the probabilistic coefficients generated by the Latent Semantic Indexing algorithm. Its performance on this corpus was over 99.99%, outperforming the performance of Naive Bayes and K-nearest neighbor.

Guo Qiang [11] proposed a text classification algorithm to improve the performance of the Naive Bayes classifier. It was applied to spam filtering on different text corpora; the results were compared to those for the classical Naive Bayes model and were shown to outperform them. The author actually proposed a new expression for word counts, which solved the problem in Naive Bayes that multiple occurrences of the same word in a document can reduce the probability of other important features which have few occurrences.

Akhter et al. [12] proposed a document classification model for the Urdu language using a single-layer multisize filters convolutional neural network (SMFCNN). They compared this model with sixteen machine learning baseline models on three imbalanced datasets. Their method achieved a higher accuracy than the selected baseline models, with accuracy values of 95.4%, 91.8%, and 93.3% on medium, large, and small size datasets, respectively.

Li et al. [13] proposed a text classification model based on the Bidirectional Encoder Representations from Transformers (BERT) model and feature fusion. A comparison with the state-of-the-art model showed that the accuracy of the proposed model outperformed those of state-of-the-art models. The model can improve the accuracy of tag prediction for text data with sequence features and obvious local features.

Du et al. [14] proposed an attention-based recurrent neural network for text classification. The network was trained on two news classification datasets published by NLPCC2014 and Reuters, respectively. The classification results showed that the model outperformed baseline models by achieving F-values of 88.5% and 51.8% on the two datasets.

Conventional models focus on the classification of a text corpus without considering the heterogeneity between the features, which may reduce the explainability of their results. Also, most text classification uses a large number of features, and working with a large number of features may cause conventional approaches to be less efficient because of the curse of dimensionality.

In this study, we propose a generative model based on collaborative partial classifications as a solution to the problem of a heterogeneous text corpus and the curse of

dimensionality in text classification. We performed experiments to evaluate the performance of our model on different heterogeneous text datasets and compared the outcome with results from other studies. We propose a method to explain the classification results of our proposed model.

The rest of the paper is organized as follows: Section 2 presents the conventional and proposed approaches to text classification and the performance measures for their evaluation. Section 3 focuses on the experimental results and discussions of the proposed approach and its comparison with other models. This work is concluded in Section 4.

2. Materials and Methods

In this section, the conventional and proposed models related to text classification in this study are presented.

2.1. Conventional Approach

The conventional approach to classifying text makes use of all the extracted features of the text corpus to predict the given category. This can be represented mathematically as

$$\hat{y} \triangleq f(X) \tag{1}$$

where X is a vector of the text features, $\hat{y}$ is the predicted value of the text category, and $f(.)$ is a function defining the prediction process.

Different conventional models are used to implement this prediction process. These include Naive Bayes (NB), Support Vector Machine (SVM), Deep Neural Networks (DNN), Bidirectional Encoder Representations from Transformers (BERT), and Recurrent Neural Networks (RNN). Figure 1 represents the conventional text classification using NB.

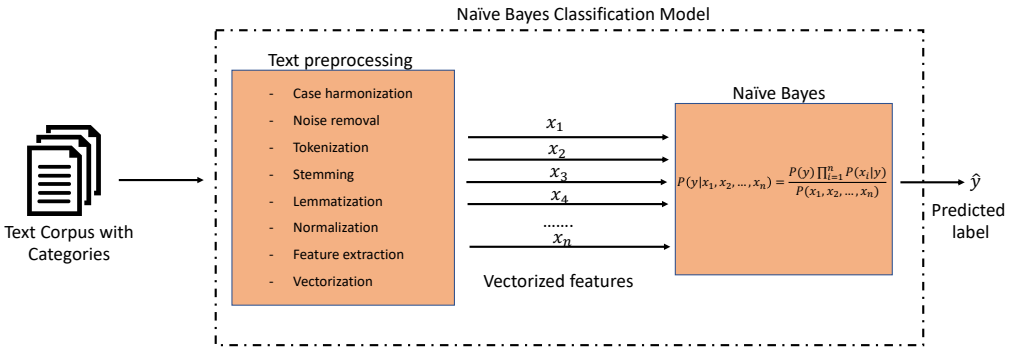

Figure 1. Text classification using Naive Bayes.

Considering a news corpus with a feature vector $X = (x_1, x_2, \ldots, x_n)$ and category set $y = \{y_1, y_2, \ldots, y_m\}$, where the elements of y are mutually exclusive. Using a conventional probabilistic learning model such as NB, the text classification can be expressed based on the Bayes rule as follows:

$$\hat{y} = P(y|x_1, x_2, \ldots, x_n) = \frac{P(y)P(x_1, x_2, \ldots, x_n|y)}{P(x_1, x_2, \ldots, x_n)} \tag{2}$$

where n is the number of features in X, $P(y)$ is the prior distribution of the category y, $\hat{y}$ is the posterior distribution of the category y, $P(X|y)$ is the likelihood of the category y given all text features in X (i.e., probability of X given y), and $P(x_1, x_2, \ldots, x_n) = P(X)$ is the marginal distribution of the text features (also called evidence).

The final predicted category (or class) $\hat{y}_k$ of y is defined by the class y_k with the maximum probability over all categories.

$$\hat{y}_k = \underset{k\in\{1,2,\ldots,m\}}{\arg\max} \left(\frac{P(y_k)P(x_1, x_2, \ldots, x_n|y_k)}{P(x_1, x_2, \ldots, x_n)} \right) \tag{3}$$

where m is the number of categories in y, $y = \{y_1, y_2, \ldots, y_m\}$, and $P(y_k)$ is the probability of a given category y_k of y.

Given that $P(x_1, x_2, \ldots, x_n)$ is independent on y, then

$$\hat{y}_k \propto \underset{k\in\{1,2,\ldots,m\}}{\arg\max} \left(P(y_k)P(x_1, x_2, \ldots, x_n|y_k) \right) \tag{4}$$

In this way, $P(x_1, x_2, \ldots, x_n)$ is considered as a normalizing factor that depends only on X, and, thus, will be a constant if the values of all the features of X are known.

The learning process based on this Bayesian inference model is then defined as an update operation that aims to maximize the predicted distribution $\hat{y}$ over the cumulative instances of X and y.

$$\max(\hat{y}) = \underset{j\in\{1,2,\ldots,l\}}{\arg\max} \left(\frac{P(y^{(j)})P(X^{(j)}|y^{(j)})}{P(X^{(j)})} \right) \tag{5}$$

where j is the numbering of the cumulative instances (also considered here as the learning or update time), and l is the total instances of X and y, i.e., the data size.

As l increases, the probability improves [15], but increase in l will also increase the computational complexity of the learning and inference process. Thus, this model is preferable with a small data size. Nevertheless, unlike data-hungry models, such as deep neural networks that require a large data size to perform well, this Bayesian model does perform well with small data sizes.

From this Bayesian inference and learning, NB is defined using the assumption of mutual independence between the features of X conditioned on the category y.

$$P(x_i|X, y) = P(x_i|y) \tag{6}$$

Thus, NB inference and learning models can be obtained from the Bayesian model as follows,

$$\hat{y}_k = \underset{k\in\{1,2,\ldots,m\}}{\arg\max} \left(\frac{P(y_k)\prod_{i=1}^{n} P(x_i|y_k)}{P(x_1, x_2, \ldots, x_n)} \right) \tag{7}$$

$$\max(\hat{y}) = \underset{j\in\{1,2,\ldots,l\}}{\arg\max} \left(\frac{P(y^{(j)})\prod_{i=1}^{n} P(x_i^{(j)}|y^{(j)})}{P(X^{(j)})} \right) \tag{8}$$

As an example, consider the sentences from a news corpus:

1. The weather is worse today due to climate change.
2. The increase in economic crises is due to the pandemic.
3. World leaders are determined to end world crises.
4. Major decisions to end climate change were made by world leaders at the climate summit.
5. During the pandemic, economic activities were shut down, making world leaders struggle with the world economy.
6. No world economy survives the pandemic.
7. World climate change summit discusses how to tackle world climate change crises during a pandemic crisis.
8. Most world leaders don't have a large economy to tackle the pandemic and climate change crises.
9. Without a sustainable economy, it may take longer to survive the pandemic shock.

10. World economic crises and pandemics are headaches to world leaders.

After preprocessing the news corpus, consider that the extracted vectorized features and labels are those shown in Table 1. The task is to classify each sentence into Business (B) or Geography (G) news based on the extracted features.

Table 1. Bag-of-Words (i.e., 1-gram word) vectorization of a news corpus.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y
1	0	0	0	1	1	0	0	G
2	1	1	1	0	0	0	0	B
3	0	1	1	0	0	2	1	B
4	0	0	0	2	1	1	1	G
5	2	0	1	0	0	2	1	B
6	1	0	1	0	0	1	0	B
7	0	2	1	2	2	2	0	G
8	1	1	1	1	1	1	1	G
9	1	0	1	0	0	0	0	B
10	1	1	1	0	0	1	1	B

x_1 denotes economy, x_2 denotes crises, x_3 denotes pandemic, x_4 denotes climate, x_5 denotes change, x_6 denotes world, x_7 denotes leader, y denotes class.

Using NB, we first compute the prior of each class, then calculate the likelihoods, and finally estimate the posterior by multiplying the prior with the likelihood since the marginal is fixed. A Laplacian smoothing is used as a normalizer to avoid the probability of zero. A log probability is used to avoid computational underflow (i.e., floating point underflow) in the case of many features.

Computing the prior and the likelihood is given as follows

$$\text{Class priors: } P(c) = \frac{n_c}{n} \tag{9}$$

$$\text{Likelihoods: } P(w|c) = \frac{n_{w,c} + \alpha}{n_c + \alpha|V|}\ ,\ \ \alpha = 1 \tag{10}$$

where c is a class, n_c is the frequency (number of occurrences) of a class, n is the total occurrences of all the classes, w is a feature, $n_{w,c}$ is the frequency (number of occurrences) of a feature given a class c, α is a smoothing parameter which is equal to 1 for Laplacian smoothing, and V is the number of vectorized features.

There are many variants of NB but the two most popular variants used for text classification are Multinomial NB [16] and Bernoulli NB [17]. For example, using a Bernoulli NB model on Table 1 will require Table 1 be transformed to a binary Bag-of-Words vectorization, as shown in Table 2; then, Equations (9) and (10) will be applied to Table 2 for prior and likelihood estimations, respectively.

Table 2. Binary Bag-of-Words (i.e., 1-gram word) vectorization of a news corpus.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y
1	0	0	0	1	1	0	0	G
2	1	1	1	0	0	0	0	B
3	0	1	1	0	0	1	1	B
4	0	0	0	1	1	1	1	G
5	1	0	1	0	0	1	1	B
6	1	0	1	0	0	1	0	B
7	0	1	1	1	1	1	0	G
8	1	1	1	1	1	1	1	G
9	1	0	1	0	0	0	0	B
10	1	1	1	0	0	1	1	B

x_1 denotes economy, x_2 denotes crises, x_3 denotes pandemic, x_4 denotes climate, x_5 denotes change, x_6 denotes world, x_7 denotes leader, y denotes class.

Using Equation (9), the prior estimations with respect to Table 2 will be

$$P(G) = 4/10, P(B) = 6/10$$

Using Equation (10), the likelihood estimations will be

$$P(x_1|G) = \frac{1+1}{4+7} = \frac{2}{11}, P(x_1|B) = \frac{5+1}{6+7} = \frac{6}{13}$$

$$P(x_2|G) = \frac{2+1}{4+7} = \frac{3}{11}, P(x_2|B) = \frac{3+1}{6+7} = \frac{4}{13}$$

$$P(x_3|G) = \frac{2+1}{4+7} = \frac{3}{11}, P(x_3|B) = \frac{6+1}{6+7} = \frac{7}{13}$$

$$P(x_4|G) = \frac{6+1}{4+7} = \frac{7}{11}, P(x_4|B) = \frac{0+1}{6+7} = \frac{1}{13}$$

$$P(x_5|G) = \frac{4+1}{4+7} = \frac{5}{11}, P(x_5|B) = \frac{0+1}{6+7} = \frac{1}{13}$$

$$P(x_6|G) = \frac{3+1}{4+7} = \frac{4}{11}, P(x_6|B) = \frac{4+1}{6+7} = \frac{5}{13}$$

$$P(x_7|G) = \frac{2+1}{4+7} = \frac{3}{11}, P(x_7|B) = \frac{3+1}{6+7} = \frac{4}{13}$$

For the same text classification problem, the Bernoulli model will be,

$$P(G|x_1, x_2, x_7) \propto \frac{2}{11} \times \frac{3}{11} \times \frac{7}{13} \times \frac{1}{13} \times \frac{1}{13} \times \frac{5}{13} \times \frac{3}{11} \times \frac{4}{10} = 0.00017$$

$$P(B|x_1, x_2, x_7) \propto \frac{6}{13} \times \frac{4}{13} \times \frac{3}{11} \times \frac{7}{11} \times \frac{5}{11} \times \frac{4}{11} \times \frac{4}{13} \times \frac{6}{10} = 0.00075$$

Since the probability for the statement to be Business news is higher than that to be Geography news, the sentence is classified as Business news.

As shown in the prediction and learning processes of conventional Bayesian learning models, such as NB, no consideration is made to capture the relationship between the features. This is considered to be computationally expensive, especially when estimating the marginal $P(X)$ for large data size and number of features. This leads to approximate solutions, such as (4), that ignore such complexities.

As previously explained, the elimination of such a relationship will affect the classification performance of the model, and eliminate the possibility to manage the heterogeneity between the features of the text corpus. In the next section, we propose an inference and learning model that takes into account such a relationship and applies it to classify heterogeneous text corpora in Section 3.

2.2. Proposed Model

Unlike the conventional approach that makes use of all the features of the text corpus to infer and learn the given category, our proposed approach provides the possibility to segment the input features into multiple groups of input features, forming different corpora, on which separate learning and inference are performed. This model is defined mathematically using probabilistic logic as follows:

Axiom 1.

$$P(X_i|X, y) = P(X_i|y) \tag{11}$$

Proposition 1.

$$\hat{y} = P(y|X) = \frac{1}{P(y)^{n-1}} \prod_{i=0}^{n} P(y|X_i) \left(\frac{P(X_{i+1}| \bigcap_{\mu=0}^{i} X_\mu)}{P(X_{i+1})} \right)^{-1} \tag{12}$$

where y is the given set of categories, X is a vector of feature vectors $X = (X_0, X_1, \ldots, X_n)$ and $X_i = (x_1, x_2, \ldots)$, $\hat{y}$ is the posterior distribution (i.e., class posterior) of y given X, $P(y)$ is the prior distribution (i.e., class prior) of y based on X, $P(y|X_i)$ is the partial posterior distribution (i.e., partial class posterior) of y given X_i, $P(X_{i+1})$ is the prior distribution (i.e., observation prior) of X_{i+1} based on $\bigcap_{\mu=0}^{i} X_\mu$, $P(X_i| \bigcap_{\mu=0}^{i} X_\mu)$ is the posterior distribution (i.e., observation posterior) of X_i given $\bigcap_{\mu=0}^{i} X_\mu$, and n is the number of features vectors.

It is worth noting that the posterior probability distributions can be interpreted as likelihood functions, i.e., the posterior of y given X is similar to the likelihood of X given y, and so on. Thus, the term posterior is used interchangeably with the term likelihood in this manuscript. The proof of Proposition 1 is given in Appendix A.

The inference and learning based on this proposed model can then be expressed as

$$\begin{aligned}
\hat{y}_k &= \underset{k \in \{1,2,\ldots,m\}}{\arg\max}\ P(y_k|X) \\
&= H(X) \underset{k \in \{1,2,\ldots,m\}}{\arg\max} \left(\frac{1}{P(y_k)^{n-1}} \prod_{i=0}^{n} P(y_k|X_i) \right) && (13) \\
&\Rightarrow \hat{y}_k \propto \underset{k \in \{1,2,\ldots,m\}}{\arg\max} \left(\frac{1}{P(y_k)^{n-1}} \prod_{i=0}^{n} P(y_k|X_i) \right) && (14) \\
\max(\hat{y}) &= \underset{j \in \{1,2,\ldots,l\}}{\arg\max} P(y_k^{(j)}|X^{(j)}) && (15)
\end{aligned}$$

where $H(X) = \prod_{i=0}^{n} \left(\frac{P(X_{i+1}| \bigcap_{\mu=0}^{i} X_\mu)}{P(X_{i+1})} \right)^{-1}$, $H(X) \in [0, \infty]$

During inference and learning, estimating the priors and likelihoods (i.e., posteriors) based on this model for both the class and observation is given as

$$\text{Class prior: } P(c) = \frac{n_c}{n} \tag{16}$$

$$\text{Observation prior: } P(w) = \frac{n_w}{n} \tag{17}$$

$$\text{Class Likelihood: } P(c|w) = \frac{n_{c,w} + \alpha}{n_w + \alpha|V|}, \ \alpha = 1 \tag{18}$$

$$\text{Observation Likelihood: } P(w_i|w_j) = \frac{n_{w_i,w_j} + \alpha}{n_{w_j} + \alpha|V|}, \ \alpha = 1 \tag{19}$$

where n is the total number of instances in the text vectorization, c is a class, n_c is the frequency (number of) occurrences of a class, w is a feature vector, n_w is the frequency of occurrences of a feature vector w, $n_{c,w}$ is the frequency of occurrence of a class c given the occurrence of a feature vector w, n_{w_i,w_j} is the frequency of occurrence of a feature vector w_i given the occurrence of another feature vector w_j, α is a smoothing parameter, which is equal to 1 for Laplacian smoothing, and V is the number of vectorized features.

Using this proposed model, for any given classification task, the first step is to segment the features into parts, forming separate feature vectors, then inference and learning are applied on each of the feature vectors using the class likelihood (or partial class posterior) $P(c|w)$. The partial class posterior can also be expressed as an inference problem on each segmented feature where our model or a Bayesian model can be applied to generate its results. The results from each partial class posterior are then integrated (or aggregated

in the case of a logarithmic scale) to represent the classification result on the whole text corpus.

Segmenting a feature vector into sub-vectors while maintaining the relationship between the features can be a daunting task. One way to approach this is to organize the segmented feature vectors in a sequence, then to apply Proposition 1, taking into account the heterogeneity between the features in the sequence. This heterogeneity between the features is given by the value of $H(X)$ and is independent of y.

The classification process is illustrated in Figure 2 and described in Algorithm 1.

As an example, using the vectorized features defined in Table 2, and applying Algorithm 1 with one feature per segment, the following priors and likelihoods can be estimated.

Algorithm 1 GenCo learning and classification of heterogeneous text corpora

Require: X (binary vectorized input), k (number of class), m(number of observation instances).

Ensure: $max(P(y|X))$

$n \leftarrow n$ ▷ number of feature segments

for $j = 1$ to m **do** ▷ learning of class posterior

$P(y^{(j)}) \leftarrow P(y^{(j)})$ ▷ estimating class priors

for $i = 0$ to n **do** ▷ estimating partial posteriors

$P(X_i^{(j)}) \leftarrow P(X_i^{(j)})$ ▷ estimating observation priors

$a_i \leftarrow P(y^{(j)}|X_i^{(j)})$ ▷ partial class posterior

$b_i \leftarrow H(X_i^{(j)})/P(y^{(j)})$ ▷ partial heterogeneity

end for

$P(y^{(j)}|X^{(j)}) \leftarrow \prod_{i=0}^{n} a_i^{(j)} b_i^{(j)}$ ▷ class posterior update

end for

$\hat{y}_k \leftarrow \underset{k \in \{1,2,3,\ldots\}}{\arg\max}\ (P(y_k|X))$ ▷ final class posterior

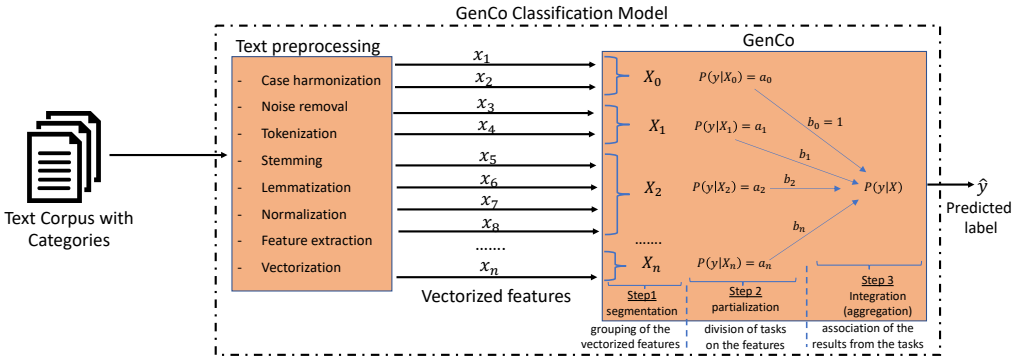

Figure 2. Text classification using GenCo.

Using Equation (16), the class priors will be

$$P(G) = 4/10, P(B) = 6/10$$

Using Equation (17), the observation priors will be

$$P(x_1 = yes) = 6/10, P(x_2 = yes) = 5/10,$$
$$P(x_3 = yes) = 8/10, P(x_4 = yes) = 4/10,$$

$$P(x_5 = yes) = 4/10, P(x_6 = yes) = 7/10,$$

$$P(x_7 = yes) = 5/10$$

Using Equation (18), the class likelihoods will be

$$P(G|x_1) = \frac{0+1}{6+7} = \frac{1}{13}, P(G|\neg x_1) = \frac{3+1}{4+7} = \frac{5}{11},$$

$$P(G|x_2) = \frac{2+1}{5+7} = \frac{3}{12}, P(G|\neg x_2) = \frac{2+1}{5+7} = \frac{3}{12},$$

$$P(G|x_3) = \frac{2+1}{8+7} = \frac{3}{15}, P(G|\neg x_3) = \frac{2+1}{2+7} = \frac{3}{9},$$

$$P(G|x_4) = \frac{4+1}{4+7} = \frac{5}{11}, P(G|\neg x_4) = \frac{0+1}{6+7} = \frac{1}{13},$$

$$P(G|x_5) = \frac{4+1}{4+7} = \frac{5}{11}, P(G|\neg x_5) = \frac{0+1}{6+7} = \frac{1}{13},$$

$$P(G|x_6) = \frac{3+1}{7+7} = \frac{4}{14}, P(G|\neg x_6) = \frac{1+1}{3+7} = \frac{2}{10},$$

$$P(G|x_7) = \frac{2+1}{5+7} = \frac{3}{12}, P(G|\neg x_7) = \frac{2+1}{5+7} = \frac{3}{12},$$

Using Equation (19) and ordering the features to be conditionally dependent from x_1 to x_7 with a Markov property of a level 1 assumption, the observation likelihoods will be

$$P(x_2|x_1) = \frac{3+1}{6+7} = \frac{4}{13}, P(x_2|\neg x_1) = \frac{1+1}{4+7} = \frac{2}{11}$$

$$P(x_3|x_2) = \frac{5+1}{5+7} = \frac{6}{13}, P(x_3|\neg x_2) = \frac{0+1}{5+7} = \frac{1}{13}$$

$$P(x_4|x_3) = \frac{1+1}{8+7} = \frac{2}{15}, P(x_4|\neg x_3) = \frac{2+1}{2+7} = \frac{3}{9}$$

$$P(x_5|x_4) = \frac{4+1}{4+7} = \frac{5}{11}, P(x_5|\neg x_4) = \frac{0+1}{6+7} = \frac{1}{13}$$

$$P(x_6|x_5) = \frac{3+1}{4+7} = \frac{4}{11}, P(x_6|\neg x_5) = \frac{4+1}{6+7} = \frac{5}{13}$$

$$P(x_7|x_6) = \frac{5+1}{7+7} = \frac{6}{14}, P(x_7|\neg x_6) = \frac{0+1}{3+7} = \frac{1}{10}$$

This implies that, using the Bernoulli version of our proposed model, the classification of a corpus given that it has the words economy, crises and leaders, will be,

$$P(G|x_1, x_2, x_7) \propto \left(\frac{4}{10}\right)^{-6} \times \frac{1}{13} \times \frac{3}{12} \times \frac{3}{9} \times \frac{1}{13} \times \frac{1}{13} \times \frac{2}{10} \times \frac{3}{12} = 0.012$$

$$P(B|x_1, x_2, x_7) \propto \left(\frac{6}{10}\right)^{-6} \times \frac{7}{13} \times \frac{4}{12} \times \frac{1}{9} \times \frac{7}{13} \times \frac{7}{13} \times \frac{3}{10} \times \frac{4}{12} = 0.046$$

Similar to the classification results using conventional models, our proposed model also classifies the statement as Business news but with larger floating point value than

the conventional model. Furthermore, the heterogeneity between the features at the last prediction instance can be estimated using the heterogeneity function $H(X)$ as follows:

$$H(X) = \prod_{i=1}^{7} \left(\frac{P(x_{i+1} | \bigcap_{\mu=1}^{i} x_\mu)}{P(x_{i+1})} \right)^{-1} = \frac{\frac{4}{13}}{\frac{5}{10}} \times \frac{\frac{1}{13}}{\frac{8}{10}} \times \frac{\frac{1}{9}}{\frac{6}{10}} \times \frac{\frac{7}{13}}{\frac{6}{10}} \times \frac{\frac{3}{13}}{\frac{3}{10}} \times \frac{\frac{1}{10}}{\frac{5}{10}} = 0.001513$$

We consider the reciprocal of this value as a form of mutuality between the features, which measures their similarity (i.e., homogeneity), and can be used to indirectly measure their heterogeneity in this model. Thus, the mutuality $M(X)$ between the features will be,

$$M(X) = \frac{1}{H(X)}, \quad M(X) \in [0, \infty] \tag{20}$$

Therefore, if $H(X) = 0.001513$, then $M(X) = 660.983143$. This implies that there is more probabilistic similarity than dissimilarity between the features. In other words, the features are probabilistically more joined together than disjoint in their occurrence.

In this study, $M(X)$ measures the correlation (or association) between the features in terms of the probabilistic similarity (i.e., homogeneity) of their dependency on one another. For any two features X_i and X_j, where each one is conditioned on the other, if $M(X_i, X_j) = 1$, then $P(X_i|X_j) = P(X_i)$ and $P(X_j|X_i) = P(X_j)$, which implies X_i and X_j are probabilistically identical, but non-similar in their dependence to each other. If $M(X) > 1$, then $P(X_i|X_j) > P(X_i)$ and $P(X_j|X_i) > P(X_j)$, which implies X_i and X_j have a direct (i.e., increase in value when conditioned) probabilistic similarity in their dependency to one another. If $M(X) < 1$, then $P(X_i|X_j) < P(X_i)$ and $P(X_j|X_i) < P(X_j)$, which implies X_i and X_j have an indirect (i.e., decrease in value when conditioned) probabilistic similarity in their dependency to one another. Using M(X) in the logarithmic scale will lead to $M(X) = 0$ (region of no mutuality), $M(X) > 0$ (region of increasing mutuality), and $M(X) < 0$ (region of decreasing mutuality), respectively. These also apply to the dissimilarity measure $H(X)$.

$M(X)$ can be compared with conventional similarity measures in NLP, such as the cosine similarity and hamming distance measures. However, unlike conventional similarity measures which are separated from the classification model, our proposed similarity measure $M(X)$ and dissimilarity measure $H(X)$ form part of our classification model; hence, they can be used to explicitly explain the classification results.

One advantage of this proposed model in text classification rests on the fact that it enables the breakdown of a high computational classification problem into smaller less computational classification problems. This may be better in terms of conventional models, whose computational complexity increases with the number of features due to the computation of the marginal distribution. The Markov property can also be applied to reduce such complexity in both the conventional and our proposed models.

Also, the use of a heterogeneity function or homogeneity function in the model, rather than a marginal function as in conventional probabilistic models, enables clear visibility of the influence of the relationship between the features to the learning and prediction processes of the model. We shall present a mutuality matrix in Section 3 to capture the probabilistic variation in the homogeneous relationship between the features during learning and show how this variation influences the learning and prediction results of the model.

Furthermore, this model can be expressed as an algebraic series as follows:

$$f(y|X) = a_0 b_0 \times a_1 b_1 \times a_2 b_2 \times a_3 b_3 \times \ldots \times a_n b_n \tag{21}$$

where $a_0 = P(y|X_0)$, $a_1 = P(y|X_1)$, $\ldots$, $a_n = P(y|X_n)$, $b_0 = 1$, $b_1 = \frac{H(X_1)}{P(y)}$, $b_2 = \frac{H(X_2)}{P(y)}$, $\ldots$, $b_n = \frac{H(X_n)}{P(y)}$, $H(X_1) = \frac{P(X_1)}{P(X_1|X_0)}$, $H(X_2) = \frac{P(X_2)}{P(X_2|X_0,x_1)}$, $\ldots$, $H(X_n) = \frac{P(X_n)}{P(X_n|\bigcap_{\mu=0}^{n-1} X_\mu}$), and $X = (X_0, X_1, X_2, \ldots, X_n)$.

The first term a_0 is considered as a bias partial class posterior in the model, and b is a combination of the heterogeneity (or mutuality) and regularization values. In general, $P(y)$ is considered to act as a regularizer to each heterogeneous (or mutual) relationship $H(X_i)$, while $H(X_i)$ acts as a normalizer of the partial class posterior a_i. In this way, $H(X)$ acts as a normalizer of the merged (integrated or aggregated) class posterior $f(y|X)$, while $(P(y))^{1-n}$ acts as a regularizer of $H(X)$.

This representation transforms the model into a network of partial actions, as shown in Figure 2. Such a representation is useful for mathematical analysis and the network can be expanded to multiple segmentation and partialization layers; however, this will increase its design and computational complexities. This network is different from conventional Bayesian networks (i.e., belief networks) [18,19] and Markov networks (i.e., Markov random field) [19], which focus on using the marginal distribution of the input features and ignore the heterogeneity between the input features.

To avoid computational overload, the conditional expressions between the features can be reduced to fewer dependent features through the application of the Markov property. Also, each term in the series can be normalized using logarithmic normalization to avoid computational underflow.

2.3. Performance Measure

The performance of this model can be evaluated based on its prediction, learning, and complexity. In this study, we focus on prediction performance. We also present a heterogeneous (and mutuality) matrix of the features to show how the heterogeneity between the given features affects the inference and learning of the model. The prediction performance is evaluated using the confusion matrix, from which the accuracy, precision, recall, and F-score can be calculated. Further discussion on prediction performance is provided in [20].

3. Experimental Results and Discussions

The aim of the experiment was to demonstrate the performance of our proposed text classification model and to compare the performance results with conventional models. For validating the performance of our model, we carried out simulation experiments on different datasets and compared the results with other models.

3.1. Experimental Setup

Two important aspects of the experiments are the model definition and the datasets.

3.1.1. Dataset and Feature Presentation

The datasets (i.e., text corpora) used for the experiment included the Twitter US Airline Sentiment dataset [21] for sentimental analysis, the Conference Paper dataset [22] for topic classification, and the SMS Spam dataset [23] for spam classification, as shown in Table 3.

Table 3. Statistics of the datasets.

Datasets	Documents	Vocabularies	Vocabulary Segments	Categories
(1) Twitter US Airline dataset [21]	14,640	100	10	3
(2) Conference Paper dataset [22]	2507	100	10	5
(3) SMS Spam dataset [23]	5574	50	5	2

3.1.2. Dataset Pre-Processing

The general data preprocessing steps, as shown in Figure 2, were used for all the datasets as part of the text classification pipeline. These data preprocessing steps were performed using the Python NLTK [24] library and were explained in Section 1 of this current paper. The vectorized vocabulary for each dataset was generated using Python

sklearn CountVectorizer. A lower and upper bound frequency of the vocabularies was set to reduce non-semantic vocabularies and computational complexity.

Each vectorized dataset was later split into 70% training and 30% test instances. Label encoding was used to encode the labels.

3.1.3. Model Definition

The model used during this experiment is based on Proposition 1 and is defined by the following hyperparameters:

- Smoothing parameter: The smoothing parameter $\alpha \in (0, 1]$, which is fixed to $\alpha = 1$, corresponding to a Laplacian smoothing.
- Number of segmentation: The number of segments used depends on the number of features (i.e., dimensions of the vocabulary) of the dataset concerned.
- Number of partialization layers: A single layer of partialization is used, and the number of partial class posteriors in the layer is equal to the number of feature segments.

3.2. Results Discussions

The classification results for the experiments are presented in this section, together with the homogeneity measure between the features of each dataset using a mutuality matrix. Also, the confusion matrix and classification report based on each dataset is presented. Lastly, we provide a comparison of the classification results of our model for each dataset with models from different studies that used the same dataset.

3.2.1. Twitter US Airline Dataset Results

The confusion matrix of our model based on the Twitter US Airline dataset is presented in Figure 3, together with the mutuality matrix of the 10 feature segments.

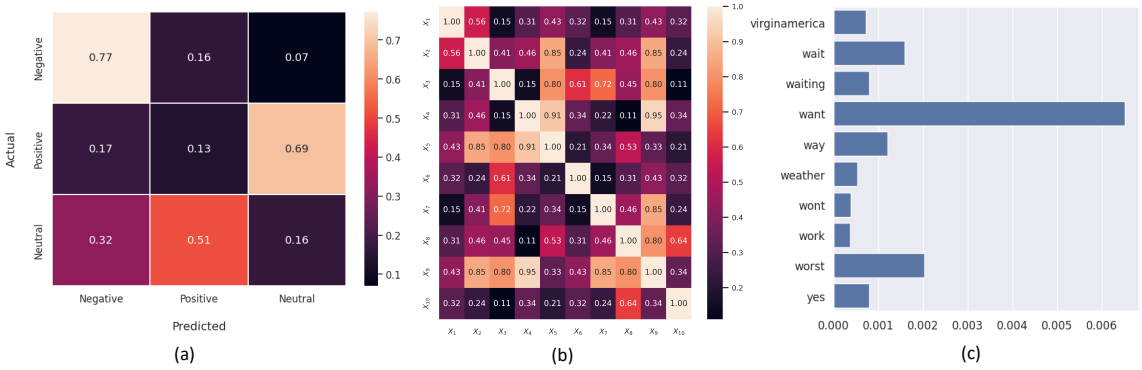

Figure 3. Results with the Twitter US Airline dataset. (**a**) Confusion matrix. (**b**) Mutuality matrix for the 10 feature segments. (**c**) Combined mutuality of each feature in the segment with the highest mutuality.

The results presented in Figure 3a show that the proposed model classifies negative labels better than both positive and neutral labels. To explain this behavior of the model, we generate the mutuality matrix for the 10 feature segments, as presented in Figure 3b.

The mutuality between every two feature segments is defined using (20), applying a level 1 Markov property assumption and Axiom 1. This results in the diagonal values next to the leading diagonal values in Figure 3b. The mutuality matrix at this level gives information about the interrelationship between the feature segments. As shown in Figure 3b, most of these relationships are decreasing mutual relationships because, for every two feature segments, X_i and X_j, $M(X_i, X_j) < 1$.

We further examined the mutuality of the features in each segment and found that the segment X_{10} with the highest combined mutuality contained features with semantically negative sentiments and whose combined mutuality was amongst the highest in

all the segments, such as the words "worst" and "wait" in Figure 3c. This implies that mutuality (and heterogeneity) between the features or feature segments plays an essential role in the classification process of this model; hence, they can be used to explain its classification results.

The type of semantic distinctions (classification) of features with respect to a class label during label classification is considered in this study to represent the semantic intelligence of the model on the labels, i.e., the ability to understand the meanings of the labels. This implies that training the predictive (causal) intelligence of this model will imply training its semantic intelligence and vice versa. However, one should not expect the semantic logic of the model on the text to always be similar to human semantic logic applied to the same text, since the model may use a different semantic logic from humans, although it is formally trained for human semantic awareness.

3.2.2. Conference Paper Dataset Results

The confusion matrix of our model based on the Conference Paper dataset is presented in Figure 4, together with the mutuality matrix of the 10 feature segments.

The results presented in Figure 4a show that the model classifies the WWW label better than the other labels. Using the mutuality matrix defined for the 10 feature segments as presented in Figure 4b, we also obtain information about the interrelationships between the feature segments, where segment X_4 has the highest combined mutuality.

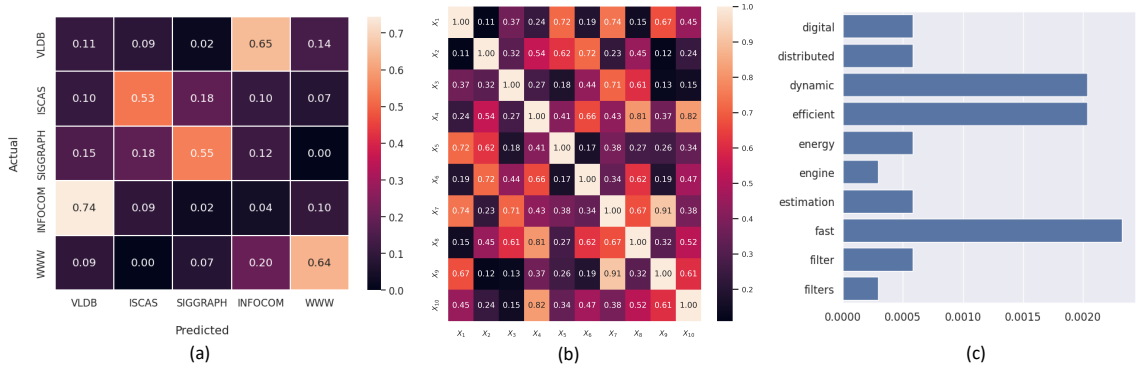

Figure 4. Results with the Conference Paper dataset. (**a**) Confusion matrix. (**b**) Mutuality matrix for the 10 feature segments. (**c**) Combined mutuality of each feature in the segment with the lowest mutuality.

Looking further into the mutuality of the features in each segment, we also found that the segment X_4 with the highest combined mutuality contained features which were semantically related to the WWW label and whose combined mutuality was amongst the highest in all segments, such as the words "dynamic", "efficient" and "fast", as shown in Figure 4c. Nevertheless, the low normalized true positive (TP) value for the WWW label implies some features in the label were not semantically classified by the model under the WWW label. The incorrect semantic classification of features with respect to the labels using mutual value allocation may account for the low TP of the other labels.

3.2.3. SMS Spam Dataset Results

The confusion matrix of our model based on the SMS Spam dataset is presented in Figure 5, together with the mutuality matrix of the five feature segments.

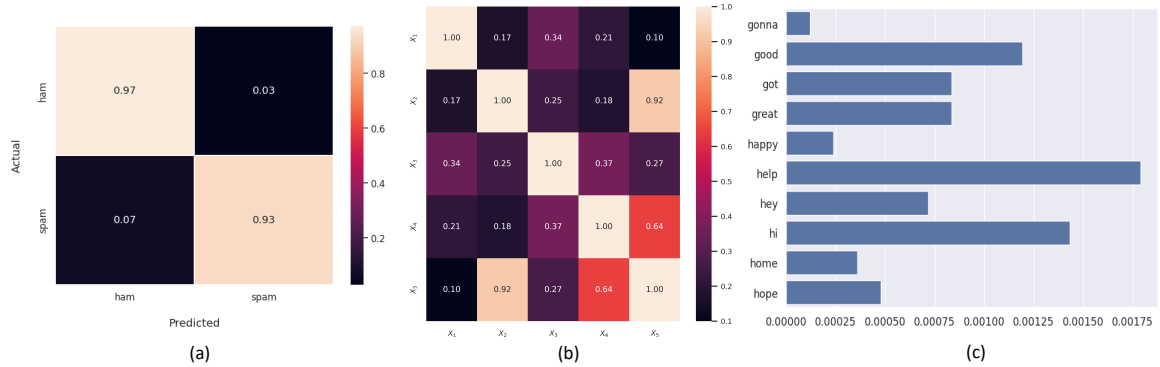

Figure 5. Results with SMS Spam dataset. (**a**) Confusion matrix. (**b**) Mutuality matrix for the 5 feature segments. (**c**) Combined mutuality of each feature in the segment with the lowest mutuality.

The results in Figure 5a show that the model classifies the "ham" label better than the "spam" label. Using the mutuality matrix defined for the five feature segments as presented in Figure 5b, segment X_2 has the highest combined mutuality; the combined mutuality of each of its features is presented in Figure 5c. The features with high combined mutuality include words such as "help" and "hi", which are common words in spam SMS, while words such as "good", "got" and "hi" are common words used in ham SMS. The high mutual value allocation on both ham and spam words explains the high true positive (TP) results for both ham and spam labels in Figure 5a.

3.2.4. Performance and Comparison with Models from Other Studies

The performance of the proposed model was compared with models from other studies, considered as baseline models. The results are presented in Table 4.

Table 4. Performance and comparison.

Datasets	Models	Accuracy (%)	BIC
Twitter US Airline dataset	RoBERTa-GRU [25]	91.52	2455.53
	ULMFit-SVM [26]	99.78	1352.18
	ABCDM [27]	92.75	1178.23
	GenCo (our work)	**98.40**	**959.18**
Conference Paper dataset	Linear SVM [28]	74.63	1041.10
	GenCo (our work)	**89.90**	**782.90**
SMS Spam dataset	Discrete HMM [29]	95.90	833.61
	Hybrid CNN-LSTM [30]	98.37	2103.36
	GenCo (our work)	**99.26**	**431.31**

As shown in Table 4, the proposed model, GenCo, resulted in performances of 98.40%, 89.9%, and 99.26%, better than other models on the Twitter US Airline dataset, Conference Paper dataset, and SMS Spam dataset, respectively. However, on the Twitter US Airline dataset, the ULMFit-SVM model had an accuracy of 99.7%, better than the GenCo model, with an accuracy of 98.4% on the same dataset. The low performance of the proposed model on some datasets can be explained by reference to the mutuality matrices from the different datasets. From these matrices, it can be inferred that the model performs better on datasets on which it can easily maximize the combined mutual value of the features or feature segments but performs less well otherwise.

Furthermore, the models were statistically compared by calculating their Bayesian information criterion (BIC) [31] on the different datasets. GenCo had the lowest BIC values of 959.18, 782.90, and 431.31 for the Twitter US Airline dataset, Conference Paper dataset, and SMS Spam dataset, respectively.

4. Conclusions

In this study, we presented a probabilistic generative model for text classification based on collaborative partial classifications. The model considers both the dimension and the heterogeneity of the features in the text corpus. A mathematical representation was provided for the model along with that of a conventional model. Using this mathematical representation, the model was implemented and tested on three different datasets, and the classification results were presented for each dataset.

For each classification result, the confusion matrix, mutuality matrix, and combined mutuality values for 10 words were presented. Using these mutuality values, the results of the confusion matrix were explained, where features or feature segments with high combined mutual value enhanced the true positive values of a particular class label in the confusion matrix, hence indicating a type of semantic intelligence. The accuracy of the model was evaluated and was observed to outperform that of conventional models on most of the datasets.

This model can be deployed in many applications, such as large-scale heterogeneous email spam filtering, multilingual fake-news detection, part-of-speech (PoS) tagging, search engines, and large language modeling. We look forward to implementing it for these different applications.

Author Contributions: Conceptualization, formal analysis, writing—original draft preparation, Z.E.E.; methodology, supervision, R.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Proof of Proposition 1. Consider the joint probability distribution $P(X_1, X_2, X_3, y)$.

$$P(X_1, X_2, X_3, y) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)P(y|X_1, X_2, X_3) \tag{A1}$$

$$P(X_1, X_2, X_3, y) = P(y)P(X_1|y)P(X_2|X_1, y)P(X_3|X_2, X_1, y) \tag{A2}$$

Equating (A1) and (A2),

$$P(y|X_1, X_2, X_3) = P(y)P(X_1|y)P(X_2|X_1, y)P(X_3|X_2, X_1, y)\frac{1}{P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)} \tag{A3}$$

Applying Axiom 1,

$$P(y|X_1, X_2, X_3) = P(y)P(X_1|y)P(X_2|y)P(X_3|y)\frac{1}{P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)} \tag{A4}$$

Applying Bayes rule to $P(X_1|y)$, $P(X_2|y)$, and $P(X_3|y)$

$$P(y|X_1, X_2, X_3) = P(y|X_1)P(y|X_2)P(y|X_3)\left[\frac{P(X_2|X_1)P(X_3|X_1, X_2)}{P(X_2)P(X_3)}\right]^{-1}\frac{1}{P(y)^2} \tag{A5}$$

Therefore, for $P(y|X = X_0, X_1, X_2, X_3, \ldots, X_n)$

$$P(y|X) = \frac{1}{P(y)^{n-1}} \prod_{i=0}^{n} P(y|X_i) \left(\frac{P(X_{i+1}| \bigcap_{\mu=0}^{i} X_\mu)}{P(X_{i+1})} \right)^{-1} \tag{A6}$$

□

References

1. Nancy, R.; Gianluca, G.; Rick, W.; Marilynn, B.; Enrique, F.; Margaret, F. Globalization and human cooperation. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 4138–4142. [CrossRef]
2. Goody, J. *The Logic of Writing and the Organization of Society*; Cambridge University Press: Cambridge, UK, 1986. [CrossRef]
3. Korde, V. Text Classification and Classifiers: A Survey. *Int. J. Artif. Intell. Appl.* **2012**, *3*, 85–99. [CrossRef]
4. Dogra, V.; Verma, S.; Kavita; Chatterjee, P.; Shafi, J.; Choi, J.; Ijaz, M.F. A Complete Process of Text Classification System Using State-of-the-Art NLP Models. *Comput. Intell. Neurosci.* **2022**, *2022*, 1883698. [CrossRef] [PubMed]
5. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D.; Barnes, L. Text Classification Algorithms: A Survey. *Information* **2019**, *10*, 150. [CrossRef]
6. Malvestuto, F.; Zuffada, C. *The Classification Problem with Semantically Heterogeneous Data*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 157–176. [CrossRef]
7. Staš, J.; Juhár, J.; Hladek, D. Classification of heterogeneous text data for robust domain-specific language modeling. *EURASIP J. Audio Speech Music. Process.* **2014**, *2014*, 14. [CrossRef]
8. Zhang, H.; Li, D. Naïve Bayes Text Classifier. In Proceedings of the 2007 IEEE International Conference on Granular Computing (GRC 2007), Fremont, CA, USA, 2–4 November 2007; p. 708. [CrossRef]
9. Xu, S. Bayesian Naïve Bayes classifiers to text classification. *J. Inf. Sci.* **2018**, *44*, 48–59. [CrossRef]
10. Mitra, V.; Wang, C.J.; Banerjee, S. Text classification: A least square support vector machine approach. *Appl. Soft Comput.* **2007**, *7*, 908–914. [CrossRef]
11. Qiang, G. An Effective Algorithm for Improving the Performance of Naive Bayes for Text Classification. In Proceedings of the 2010 Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, 7–10 May 2010; pp. 699–701.
12. Akhter, M.P.; Jiangbin, Z.; Naqvi, I.R.; Abdelmajeed, M.; Mehmood, A.; Sadiq, M.T. Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network. *IEEE Access* **2020**, *8*, 42689–42707. . [CrossRef]
13. Li, W.; Gao, S.; Zhou, H.; Huang, Z.; Zhang, K.; Li, W. The Automatic Text Classification Method Based on BERT and Feature Union. In Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 4–6 December 2019; pp. 774–777.
14. Du, C.; Huang, L. Text Classification Research with Attention-based Recurrent Neural Networks. *Int. J. Comput. Commun. Control* **2018**, *13*, 50. [CrossRef]
15. Wilbur, W.J. Boosting naïve Bayesian learning on a large subset of MEDLINE. In *Proceedings of the AMIA Symposium*; American Medical Informatics Association: Bethesda, MD, USA, 2000; pp. 918–922.
16. Xu, S.; Li, Y.; Wang, Z. Bayesian Multinomial Naive Bayes Classifier to Text Classification. In *Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2017*; Springer: Singapore, 2017.
17. Manning, C.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; pp. 253–286. [CrossRef]
18. Daly, R.; Shen, Q.; Aitken, S. Learning Bayesian networks: Approaches and issues. *Knowl. Eng. Rev.* **2011**, *26*, 99–157. [CrossRef]
19. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012.
20. Vihinen, M. How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. *BMC Genom.* **2012**, *13* (Suppl. S4), S2. [CrossRef] [PubMed]
21. Figure, E. Twitter US Airline Sentiment Dataset. 2019. Available online: https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment (accessed on 25 February 2023).
22. Harun, R. Research Papers Dataset. 2018. Available online: https://www.kaggle.com/datasets/harunshimanto/research-paper (accessed on 25 February 2023).
23. Almeida, T.A.; Hidalgo, J.M.G.; Yamakami, A. Contributions to the Study of SMS Spam Filtering: New Collection and Results. In Proceedings of the 11th ACM Symposium on Document Engineering, Mountain View, CA, USA, 19–23 September 2011; pp. 259–262. [CrossRef]
24. Bird, S.; Edward, L.; Ewan, K. *Natural Language Processing with Python*; O'Reilly Media Inc.: Sebastopol, CA, USA, 2009.
25. Tan, K.L.; Lee, C.P.; Lim, K.M. RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis. *Appl. Sci.* **2023**, *13*, 3915. [CrossRef]
26. AlBadani, B.; Shi, R.; Dong, J. A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM. *Appl. Syst. Innov.* **2022**, *5*, 13. [CrossRef]

27. Basiri, M.E.; Nemati, S.; Abdar, M.; Cambria, E.; Acharrya, U.R. ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *115*, 279–294. [CrossRef]
28. Li, S. Machine Learning SpaCy. 2018. Available online: https://github.com/susanli2016/Machine-Learning-with-Python/blob/master/machine%20learning%20spaCy.ipynb (accessed on 24 February 2023).
29. Xia, T.; Chen, X. A Discrete Hidden Markov Model for SMS Spam Detection. *Appl. Sci.* **2020**, *10*, 5011. [CrossRef]
30. Ghourabi, A.; Mahmood, M.A.; Alzubi, Q.M. A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages. *Future Internet* **2020**, *12*, 156. [CrossRef]
31. Schwarz, G. Estimating the Dimension of a Model. *Ann. Stat.* **1978**, *6*, 461–464. [CrossRef]