Article

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

Zie Eya Ekolle [1] and Ryuji Kohno [1]

[1] Department of Electrical and Computer Engineering, Yokohama National University, Yokohama 240-8501, Japan

Abstract: The use of artificial intelligence 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 fi ltering. With a large amount of text generated each day from different sources such as webpages, blogs, emails, social media, and articles, one of the most common tasks in natural language processing is the classification of a text corpus. This is important in many institutions for planning, decision-making, and archives of their projects. Many algorithms exist to automate text classification operations but the most intriguing of them is that which also learns these operations 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 o utput. It provides a heterogeneity measure to explain its classification results and enables the reduction of the curse of dimensionality in text classification. The classification results are compared with those of conventional text classification models, and it shows that our proposed model has a higher classification performance than conventional models.

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

1. Introduction

They have been a tremendous increase in human interactions over the past years due to the rise in globalization. Coupled with the increase in dependency on electronic communication, the amount of electronic data generated grows rapidly 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. Added to the increase in human interaction in recent years, a large amount of textual information is generated each day from different sources such as web pages, blogs, emails, social media, and articles.

To understand the content of textual information, many language analysis operations such as lexical (or morphological) analysis, syntax analysis (or parsing), semantic analysis, topic modeling, and text classification, can be carried out on the text corpus. But the increase in the amount of textual information poses huge challenges in processing them.

In this paper, we focus on the text classification operation and provide a machine-learning solution to automate the classification of vast amounts of textual information.

Text classification consists of assigning a sentence or document to an appropriate predefined category [1]. This category can involve topic, sentiment, language, or all. So, text classification operations may include news classification, emotion classification, sentiment analysis, citation intent classification, spam classification, and so on. This is important in many institutions for archives and organization of large amounts of textual information needed for effective planning and decision-making on their projects.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

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

Figure 1 present the pipeline of a general text classification operation [2] in natural language processing. As shown in Figure 1, the text corpus (or dataset) undergoes a lengthy preprocessing step before the actual classification is done. During preprocessing, the text corpus is processed for case harmonization, noise removal (e.g stop words and special characters removal), tokenization, stemming, lemmatization, normalization (e.g spelling correction), feature extraction, and vectorization. After preprocessing, the vectorized features are then fed into a classification algorithm which outputs the prediction of the category that defines the given text corpus.

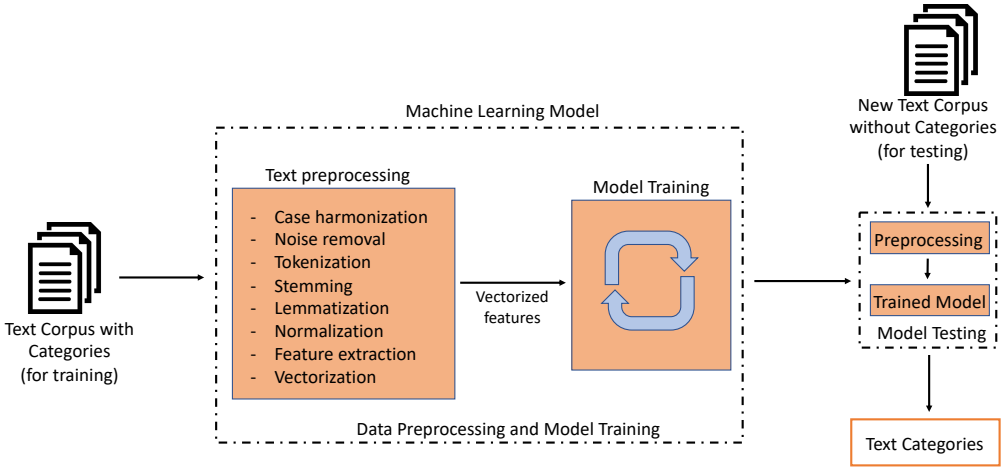

Figure 1. General pipeline for a text classification operation.

The importance of text classification has led to the development of many algorithms to automate the operation [3]. 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 operation has increased the use of the stochastic approach in text classification operations.

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 further be divided into supervised, semi-supervised, unsupervised, and self-supervised 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 outputs (SIMO), and multiple input multiple outputs (MIMO) models. Multiple output (MO) models are also called multitarget models which include multilabel models for classification tasks. In this paper, we focus on using a MISO-based generative supervised learning algorithm for text classification.

A major challenge in text classification algorithms is the classification of heterogeneous text corpus (or corpora) [4,5]. Heterogeneous text corpus (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., 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).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

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 for example. The Heterogeneity measure 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 operations 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 count limits of feature occurrences, are done 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.

1.1. Related works

In the field of Natural language processing, much research has been done to provide solutions for text classification.

Zhang et al.[6] presented a text classifier using Naive Bayes. They applied their model to spam filtering by using pre-classified emails as prior knowledge to train their model. Their model was able to detect if an email is spam or not spam. Also, Shuo [7] proposed a Gaussian Naive Bayes model for text classification.

Mitra et al.[8] presented a text classifier using the least square support vector machine on a corpus of 91,229 words from the University of Denver's Penrose Library catalog. Its performance on this corpus is over 99.99% and the results were compared to that of Naive Bayes and K-nearest neighbor.

Guo Qiang[9] proposed a text classification algorithm to improve the performance of the Naive Bayes classifier. It was applied to spam filtering on different text corpora, and the results were compared to that of other models and were shown to outperform them.

Akhter et al. [10] proposed a document classification model using single-layer multisize filters convolutional neural network (SMFCNN). They further compare this model with sixteen machine learning baseline models on three imbalanced datasets. Their method achieved a higher accuracy than the selected baseline models.

Li et al. [11] proposed a text classification model based on bidirectional encoder representations from transformers (BERT) model and features union. A comparison with the state-of-the-art model shows that the accuracy of the proposed model outperforms those of state-of-the-art models.

Du et al. [12] proposed an attention-based recurrent neural network for text classification. The network is trained on two news classification datasets published by NLPCC2014 and Reuters, respectively. The classification result shows that the model outperforms baseline models by achieving an F-values of 88.5% and 51.8% on the two datasets.

Conventional models focus on the classification of text corpus without considering the heterogeneity between the features, which may reduce the explainability of their results. Furthermore, most text classification makes use of 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. We provide a solution that performs well on high-dimensional text corpus and whose result can clearly be explained.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

1.2. Contribution

Our contributions to this study are:

- We proposed a generative model based on collaborative partial classifications as a solution to the problem of heterogeneous text corpus and the curse of dimensionality in text classification operation.
- We did experiments to prove the performance of our model on different heterogeneous text datasets and compare this with results from other studies.
- We proposed a method to explain the classification results of our proposed model.

1.3. Organization

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, we present the conventional and proposed models related to text classification in this study.

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

$$\hat{y} \triangleq f(X) \tag{2.1}$$

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 operation.

Different conventional models are used to achieve this operation. This includes Naive Bayes (NB), support vector machine (SVM), deep neural networks (DNN), bidirectional encoder representations from transformers (BERT), and recurrent neural networks (RNN). Figure 2 represent the conventional text classification operation using Niave Bayes.

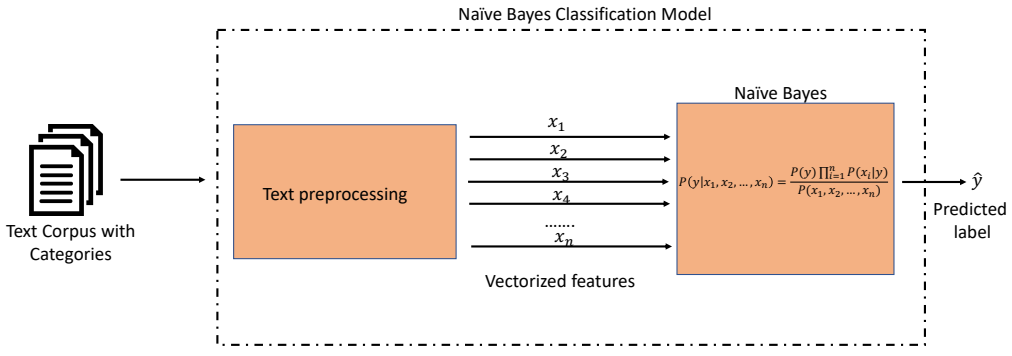

Figure 2. Text classification operation using Naive Bayes.

Considering a news corpus with feature vector $X = (x_1, x_2, .., x_n)$ and category set $y = \{y_1, y_2, ..., y_m\}$, where the elements of y are mutually exclusive. Using a conventional probabilistic

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

learning model such as Naive Bayes, the text classification operation can be expressed based on the Bayes rule as follows:

$$\hat{y} = P(y|x_1, x_2, ..., x_n) = \frac{P(y)P(x_1, x_2, ..., x_n|y)}{P(x_1, x_2, ..., x_n)} \tag{2.2}$$

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 $P(x_1, x_2, ..., x_n) = P(X)$ 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.

$$\hat{y}_k = \underset{k \in \{1,2,...,m\}}{\arg\max} \left(\frac{P(y_k)P(x_1, x_2, ..., x_n|y_k)}{P(x_1, x_2, ..., x_n)} \right) \tag{2.3}$$

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

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

$$\hat{y}_k \propto \underset{k \in \{1,2,...,m\}}{\arg\max} \left(P(y_k)P(x_1, x_2, ..., x_n|y_k) \right) \tag{2.4}$$

In this way, $P(x_1, x_2, ..., x_n)$ 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.

$$\max(\hat{y}) = \underset{j \in \{1,2,...,l\}}{\arg\max} \left(\frac{P(y^{(j)})P(X^{(j)}|y^{(j)})}{P(X^{(j)})} \right) \tag{2.5}$$

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 gets better[13], but an 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, Naive Bayes is defined using the assumption of mutual independence between the features of X conditioned on the category y.

$$P(x_i|X, y) = P(x_i|y) \tag{2.6}$$

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

$$\hat{y}_k = \underset{k \in \{1,2,...,m\}}{\arg\max} \left(\frac{P(y_k) \prod_{i=1}^{n} P(x_i|y_k)}{P(x_1, x_2, ..., x_n)} \right) \tag{2.7}$$

$$\max(\hat{y}) = \underset{j \in \{1,2,...,l\}}{\arg\max} \left(\frac{P(y^{(j)}) \prod_{i=1}^{n} P(x_i^{(j)}|y^{(j)})}{P(X^{(j)})} \right) \tag{2.8}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

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

1. The weather gets worst 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 this year.
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. Must world leaders don't have a large economy to tackle both 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 pre-processing 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 word (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 Naive Bayes, we first compute the prior of each class, then compute the likelihoods, and finally estimate the posterior by multiplying the prior with the likelihood since the marginal is fixed. A Laplacian smoothing is performed as a normalizer to avoid the occurrence of zero probability. To avoid computational underflow (i.e., floating point underflow) in the case of many features, a log probability is used.

Computing the prior and the likelihood is given as follows

$$\text{Class priors: } P(c) = \frac{n_c}{n} \tag{2.9}$$

$$\text{Likelihoods: } P(w|c) = \frac{n_{w,c} + \alpha}{n_c + \alpha|V|}, \quad \alpha = 1 \tag{2.10}$$

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.

They are many variants of Naive Bayes but the two most popular variants used for text classification are Multinomial Naive Bayes[14] and Bernoulli Naive Bayes[15].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

For example, using a Bernoulli Naive Bayes model on Table 1 will require Table 1 be transformed to a binary bag of words vectorization as shown in Table 2, then equations (2.9) and (2.10) will be applied to Table 2 for prior and likelihood estimations, respectively.

Table 2. Binary Bag of word (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 (2.9), the prior estimations with respect to Table 2 will be
$P(G) = 4/10$,$P(B) = 6/10$
Using equation (2.10), 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 Naive Bayes, no consideration is taken 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 lead to approximate solutions such as (2.4), that ignore such complexities.

As earlier 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 relationship and apply it to classify heterogeneous text corpora in Section 3.

2.2. Proposed Model

Unlike the conventional approach which 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 features into multiple groups of input features, forming different corpora, on which separate learning and inference are done independently. This model is defined mathematically using a probabilistic logic as follows:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

Axiom 2.1.

$$P(X_i|X,y) = P(X_i|y) \tag{2.11}$$

Proposition 2.1.

$$\hat{y} = P(y|X) = \frac{1}{P(y)^{n-1}} \prod_{i=0}^{n} P(y|X_i) \left(\frac{P(X_{i+1}|\bigcap_{\mu=0}^{i} X_\mu)}{P(X_{i+1})} \right)^{-1} \tag{2.12}$$

where y is the given set of categories, X is a vector of feature vectors $X = (X_0, X_1, ..., X_n)$ and $X_i = (x_1, x_2, ...)$, $\hat{y}$ 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 2.1 is given in Appendix (A).

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

$$\begin{aligned}
\hat{y}_k &= \underset{k\in\{1,2,...,m\}}{\arg\max}\ P(y_k|X) \\
&= H(X) \underset{k\in\{1,2,...,m\}}{\arg\max} \left(\frac{1}{P(y_k)^{n-1}} \prod_{i=0}^{n} P(y_k|X_i) \right) && (2.13) \\
&\Rightarrow \hat{y}_k \propto \underset{k\in\{1,2,...,m\}}{\arg\max} \left(\frac{1}{P(y_k)^{n-1}} \prod_{i=0}^{n} P(y_k|X_i) \right) && (2.14) \\
\max(\hat{y}) &= \underset{j\in\{1,2,...,l\}}{\arg\max} P(y_k^{(j)}|X^{(j)}) && (2.15)
\end{aligned}$$

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

$$\text{Class prior: } P(c) = \frac{n_c}{n} \tag{2.16}$$

$$\text{Observation prior: } P(w) = \frac{n_w}{n} \tag{2.17}$$

$$\text{Class Likelihood: } P(c|w) = \frac{n_{c,w} + \alpha}{n_w + \alpha|V|}, \quad \alpha = 1 \tag{2.18}$$

$$\text{Observation Likelihood: } P(w_i|w_j) = \frac{n_{w_i,w_j} + \alpha}{n_{w_j} + \alpha|V|}, \quad \alpha = 1 \tag{2.19}$$

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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

Using this proposed model, for any given classification task, the first step is to segment the features into parts, forming separate feature vectors, then apply inference and learning 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 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 apply Proposition 2.1, 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 it is independent of y.

The classification process is illustrated in Figure 3 and described in Algorithm 1.

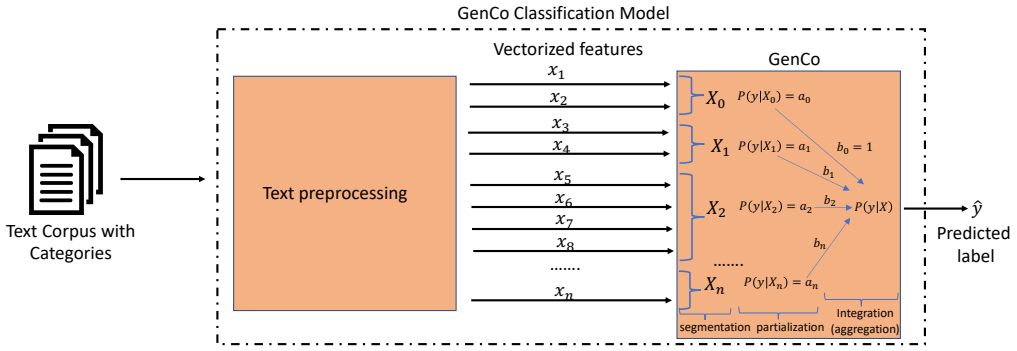

Figure 3. Text classification operation using GenCo.

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

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.

Using equation (2.16), the class priors will be
$P(G) = 4/10$,$P(B) = 6/10$

Using equation (2.17), 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 (2.18), the class likelihoods will be
$P(G|x_1) = \frac{0+1}{6+7} = \frac{1}{13}$, $P(G|\neg x_1) = \frac{3+1}{4+7} = \frac{5}{11}$,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

$P(G|x_2) = \frac{2+1}{5+7} = \frac{3}{12}, P(G|\neg x_2) = \frac{2+1}{5+7} = \frac{3}{12},$
$P(G|x_3) = \frac{2+1}{8+7} = \frac{3}{15}, P(G|\neg x_3) = \frac{2+1}{2+7} = \frac{3}{9},$
$P(G|x_4) = \frac{4+1}{4+7} = \frac{5}{11}, P(G|\neg x_4) = \frac{0+1}{6+7} = \frac{1}{13},$
$P(G|x_5) = \frac{4+1}{4+7} = \frac{5}{11}, P(G|\neg x_5) = \frac{0+1}{6+7} = \frac{1}{13},$
$P(G|x_6) = \frac{3+1}{7+7} = \frac{4}{14}, P(G|\neg x_6) = \frac{1+1}{3+7} = \frac{2}{10},$
$P(G|x_7) = \frac{2+1}{5+7} = \frac{3}{12}, P(G|\neg x_7) = \frac{2+1}{5+7} = \frac{3}{12},$

Using equation (2.19) and ordering the features to be conditionally dependent from x_1 to x_7 with a Markov property of 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 and with more certainty 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,

$$M(X) = \frac{1}{H(X)}, \quad M(X) \in [0, \infty] \tag{2.20}$$

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 joint 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 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 natural language processing such as the cosine similarity measure and hamming distance. But unlike conventional similarity measures which are separated from the classification model, our proposed similarity measure $M(X)$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

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 operation rest 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 conventional and our proposed models.

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

$$f(y|X) = a_0b_0 \times a_1b_1 \times a_2b_2 \times a_3b_3 \times ... \times a_nb_n \tag{2.21}$$

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 regularize of $H(X)$.

This representation transforms the model into a network of partial actions as shown in Figure 3. Such a representation is handy for mathematical analysis and the network can be expanded to multiple segmentation and partialization layers, but this will increase its design and computational complexities. This network is different from conventional Bayesian networks (i.e., belief networks) [16],[17] and Markov networks (i.e., Markov random field) [17], 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 the 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. More on prediction performance is discussed in [18].

3. Experimental Results and Discussions

The aim of the experiment is to demonstrate the performance of our proposed text classification model and compare the performance results with conventional models. For validating the performance

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

of our model, we carried out simulation experiments on different datasets and compare 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 include the Twitter US Airline Sentiment dataset [19] for sentimental analysis, Conference paper dataset [20] for topic classification, and the SMS spam dataset [21] 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 [19]	14640	100	10	3
2) Conference paper dataset [20]	2507	100	10	5
3) SMS spam dataset [21]	5574	50	5	2

3.1.2. Dataset pre-processing

The general data preprocessing steps as shown in Figure 1 were used for all the datasets as part of the text classification pipeline. These data preprocessing steps were done using Python NLTK [22] library, and consist of case harmonization, noise removal (specifically stop words and punctuations), tokenization, lemmatization, normalization, and feature (vocabulary) extraction. 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 2.1 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., dimension of the vocabulary) of the dataset concerned.
- Number of partialization layers: A single layer of partialization is used, and the number of partial class posterior in the layer is equal to the number of feature segments.

3.2. Results Discussions

After running the experiments, the classification results 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 are presented. Lastly, we compared 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 4 together with the mutuality matrix of the 10 feature segments.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

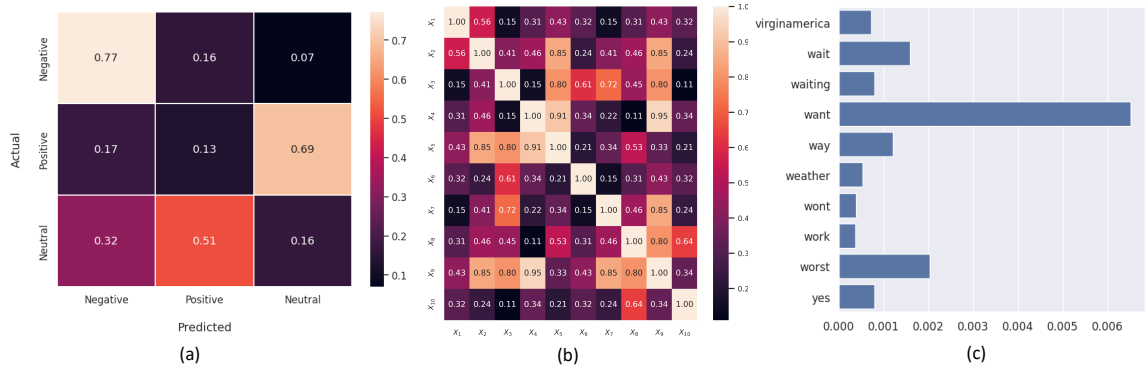

Figure 4. Results with Twitter US airline dataset. (a) Confusion matrix. (b) Mutuality matrix for the 10 feature segments. (c) Combined mutuality of each feature in the segment with highest mutuality.

The results in Figure 4(a) shows that the proposed model classifies negative labels better than both positive and neutral labels. To explain such behavior of the model, we generate the mutuality matrix for the 10 feature segments as presented in Figure 4(b).

The mutuality between every two feature segments is defined using (2.20), and applying a level one Markov property assumption and Axiom 2.1. This results in the diagonal values next to the leading diagonal values in Figure 4(b). The mutuality matrix at this level gives information about the interrelationship between the feature segments. As shown in Figure 4(b), 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 look into the mutuality of the features in each segment and found out that the segment X_{10} with the highest combined mutuality contains features with semantically negative sentiments and whose combined mutuality is amongst the highest in all segments such as the words "worst" and "wait" in Figure 4(c). This implies that mutuality (and heterogeneity) between the features or feature segments plays an essential role in the classification process of this model, hence can be used to explain its classification results.

Such type of semantic distinctions (classification) of features with respect to a class label during label classification is considered in this study as a semantic intelligence of the model, i.e., the ability to understand meanings. This implies that training the predictive (causal) intelligence of this model will imply training its semantic intelligence, and vice-versa. But one should not expect the semantic logic of the model on the text to always be similar to human semantic logic on the same text, since the model may use different semantic logic from humans, except 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 5 together with the mutuality matrix of the 10 feature segments.

The results in Figure 5(a) shows that the model classifies WWW label better than the other labels. Using the mutuality matrix defined for the 10 feature segments as presented in Figure 5(b), we also get information about the interrelationship between the feature segments, where segment X_4 has the highest combined mutuality.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

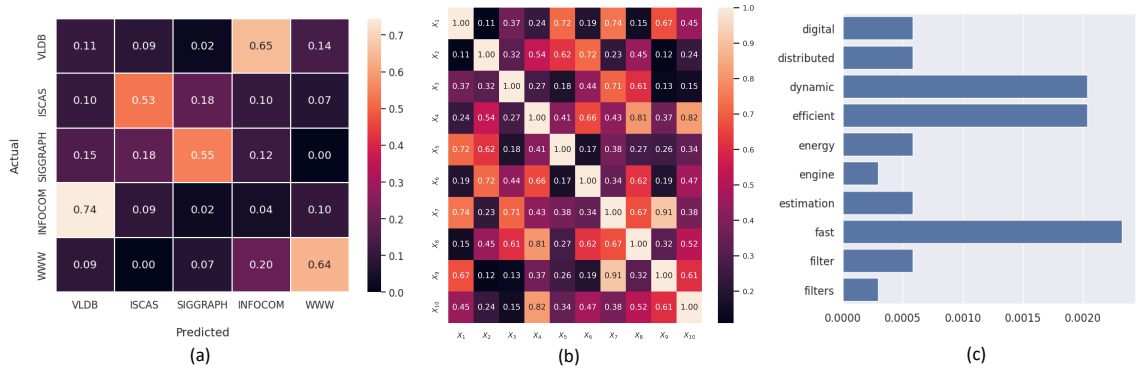

Figure 5. Results with Conference paper dataset. (a) Confusion matrix. (b) Mutuality matrix for the 10 feature segments. (c) Combined mutuality of each feature in the segment with lowest mutuality.

Looking further into the mutuality of the features in each segment, we also found out that the segment X_4 with the highest combined mutuality contained features which are semantically related to the WWW label and whose combined mutuality is amongst the highest in all segments such as the words "dynamic", "efficient" and "fast" in Figure 5(c). 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 wrong semantic classification of features with respect to the labels using mutual value allocation may be used to explain 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 6 together with the mutuality matrix of the 5 feature segments.

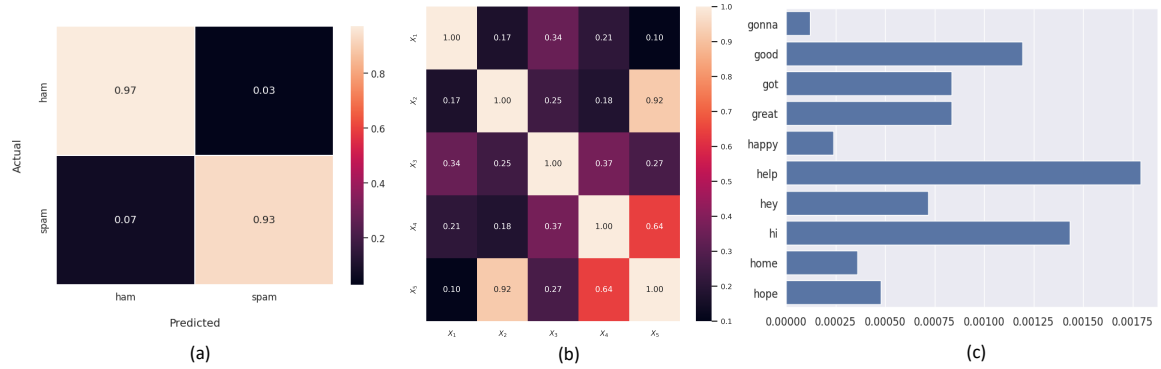

Figure 6. Results with SMS spam dataset. (a) Confusion matrix. (b) Mutuality matrix for the 5 feature segments. (c) Combined mutuality of each feature in the segment with lowest mutuality.

The results in Figure 6(a) shows that the model classifies the "ham" label better than the spam label. Using the mutuality matrix defined for the 5 feature segments as presented in Figure 6(b), segment X_2 has the highest combined mutuality, and the combined mutuality of each of its features are presented in Figure 6(c). The features with high combined mutuality include words such as "help" and "hi" are common words in the spam SMS, while words such as "good", "got" and "hi" are common words use in the 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 6(a).

3.2.4. Performance and Comparison with models from other studies

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2023 doi:10.20944/preprints202306.0078.v1

Table 4. Performance and comparison.

Datasets	Models	Accuracy (%)
Twitter US airline dataset	RoBERTa-GRU [23]	91.52
	ULMFit-SVM [24]	99.78
	ABCDM [25]	92.75
	GenCo (our work)	**98.4**
Conference paper dataset	Linear SVM [26]	74.63
	GenCo (our work)	**89.9**
SMS spam dataset	Discrete HMM [27]	95.90
	Hybrid CNN-LSTM [28]	98.37
	GenCo (our work)	**99.26**

As shown in Table 4, the proposed model, GenCo, performed better than other models on the Twitter US airline dataset, Conference dataset, and SMS spam dataset. On the Twitter US airline dataset, the ULMFit-SVM model has 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 using its mutuality matrix from the different datasets. From these matrices, one can consider that the model performs better on datasets on which it can easily maximize the combined mutual value of the features or feature segments and less performant if otherwise.

4. Conclusion

In this study, we presented a probabilistic generative model for text classification based on collaborative partial classifications. The model considers both the dimension and 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 turn to enhance the true positive value of a particular class label in the confusion matrix, hence a type of semantic intelligence. Last, the accuracy of the model was evaluated and outperformed those of conventional models on most of the datasets.

Appendix A. Proof

Proof of Proposition (2.1)

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 (A.1) and (A.2),

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

Applying Axiom (2.1),

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

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. Korde, V. Text Classification and Classifiers:A Survey. *International Journal of Artificial Intelligence and Applications* **2012**, *3*, 85–99. doi:10.5121/ijaia.2012.3208.
2. Dogra, V.; Verma, S.; ., K.; Chatterjee, P.; Shafi, J.; Choi, J.; Ijaz, M.F. A Complete Process of Text Classification System Using State-of-the-Art NLP Models. *Computational Intelligence and Neuroscience* **2022**, *2022*, 1–26. doi:10.1155/2022/1883698.
3. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D.; Id, L.; Barnes. Text Classification Algorithms: A Survey. *Information (Switzerland)* **2019**, *10*. doi:10.3390/info10040150.
4. Malvestuto, F.; Zuffada, C., The classification problem with semantically heterogeneous data; 2006; pp. 157–176. doi:10.1007/BFb0027512.
5. Staš, J.; Juhár, J.; Hladek, D. Classification of heterogeneous text data for robust domain-specific language modeling. *EURASIP Journal on Audio Speech and Music Processing* **2014**, *2014*. doi:10.1186/1687-4722-2014-14.
6. Zhang, H.; Li, D. Naïve Bayes Text Classifier. 2007, pp. 708 – 708. doi:10.1109/GrC.2007.40.
7. Xu, S. Bayesian Naïve Bayes classifiers to text classification. *Journal of Information Science* **2018**, *44*, 48 – 59.
8. Mitra, V.; Wang, C.J.; Banerjee, S. Text classification: A least square support vector machine approach. *Applied Soft Computing* **2007**, *7*, 908–914. doi:10.1016/j.asoc.2006.04.002.
9. Qiang, G. An Effective Algorithm for Improving the Performance of Naive Bayes for Text Classification. *2010 Second International Conference on Computer Research and Development* **2010**, pp. 699–701.
10. 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. doi:10.1109/ACCESS.2020.2976744.
11. Li, W.; Gao, S.; Zhou, H.; Huang, Z.; wei Zhang, K.; Li, W. The Automatic Text Classification Method Based on BERT and Feature Union. *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)* **2019**, pp. 774–777.
12. Du, C.; Huang, L. Text Classification Research with Attention-based Recurrent Neural Networks. *International Journal of Computers Communications & Control* **2018**, *13*, 50. doi:10.15837/ijccc.2018.1.3142.
13. Wilbur, W.J. Boosting naïve Bayesian learning on a large subset of MEDLINE. *Proceedings. AMIA Symposium* **2000**, pp. 918–22.
14. Xu, S.; Li, Y.; Wang, Z. Bayesian Multinomial Naive Bayes Classifier to Text Classification. MUE/FutureTech, 2017.
15. Singh, M.; Bhatt, M.W.; Bedi, H.S.; Mishra, U. Performance of bernoulli's naive bayes classifier in the detection of fake news. *Materials Today: Proceedings* **2020**.
16. Daly, R.; Shen, Q.; Aitken, S. Learning Bayesian networks: Approaches and issues. *Knowledge Eng. Review* **2011**, *26*, 99–157. doi:10.1017/S0269888910000251.
17. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press, 2012.
18. Vihinen, M. How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. *BMC genomics* **2012**, *13 Suppl 4*, S2. doi:10.1186/1471-2164-13-S4-S2.
19. Twitter US Airline Sentiment dataset.
20. Research papers dataset.
21. Almeida, T.A.; Hidalgo, J.M.G.; Yamakami, A. Contributions to the Study of SMS Spam Filtering: New Collection and Results. Association for Computing Machinery, 2011, p. 259–262. doi:10.1145/2034691.2034742.

22. Bird, S.; Edward, L.; Ewan, K. *Natural Language Processing with Python*; O'Reilly Media Inc, 2009.
23. Tan, K.L.; Lee, C.P.; Lim, K.M. RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis. *Applied Sciences* **2023**, *13*. doi:10.3390/app13063915.
24. 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. *Applied System Innovation* **2022**, *5*. doi:10.3390/asi5010013.
25. 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.
26. Li, S. Machine Learning SpaCy. https://github.com/susanli2016/Machine-Learning-with-Python/blob/master/machine%20learning%20spaCy.ipynb, 2018.
27. Xia, T.; Chen, X. A Discrete Hidden Markov Model for SMS Spam Detection. *Applied Sciences* **2020**, *10*. doi:10.3390/app10145011.
28. 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*. doi:10.3390/fi12090156.