Feature Substitution Using Latent Dirichlet Allocation for Text Classification

Norsyela Muhammad Noor Mathivanan[1], Roziah Mohd Janor[2], Shukor Abd Razak[3], Nor Azura Md. Ghani[4]*
College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia[1, 2, 4]
School of Computing and Creative Media, University of Wollongong Malaysia, Shah Alam, Malaysia[1]
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Malaysia[3]

***Abstract*—Text classification plays a pivotal role in natural language processing, enabling applications such as product categorization, sentiment analysis, spam detection, and document organization. Traditional methods, including bag-of-words and TF-IDF, often lead to high-dimensional feature spaces, increasing computational complexity and susceptibility to overfitting. This study introduces a novel Feature Substitution technique using Latent Dirichlet Allocation (FS-LDA), which enhances text representation by replacing non-overlapping high-probability topic words. FS-LDA effectively reduces dimensionality while retaining essential semantic features, optimizing classification accuracy and efficiency. Experimental evaluations on five e-commerce datasets and an SMS spam dataset demonstrated that FS-LDA, combined with Hidden Markov Models (HMMs), achieved up to 95% classification accuracy in binary tasks and significant improvements in macro and weighted F1-scores for multiclass tasks. The innovative approach lies in FS-LDA's ability to seamlessly integrate dimensionality reduction with feature substitution, while its predictive advantage is demonstrated through consistent performance enhancement across diverse datasets. Future work will explore its application to other classification models and domains, such as social media analysis and medical document categorization, to further validate its scalability and robustness.**

Keywords*—Feature extraction; feature selection; Latent Dirichlet Allocation; text classification; Hidden Markov Model; dimensionality reduction

I. Introduction

The exponential growth of online content has transformed digital platforms into key sources for global information acquisition and dissemination. With the rise of unstructured text data from these platforms, there is an increasing need for efficient techniques to analyze and manage large-scale text data, which often surpasses numeric data in volume and complexity [1], [2]. Text mining has emerged as a crucial tool for processing unstructured data, supporting decision-making through tasks like classification, clustering, summarization, association rule mining, and topic detection [2]. Among these tasks, text classification plays a vital role in organizing diverse textual data, including e-commerce products, tweets, news articles, and customer reviews, into structured groups [3]. This process has been widely adopted in various fields, such as product categorization [4], [5], sentiment analysis [6], spam detection [7], news classification [8], and medical document classification [9].

Effective text classification relies on noise-free features that capture the essential semantic meaning of the data [2]. However, large-scale text corpora are often high-dimensional, posing challenges for computational efficiency and model accuracy. Input data preparation, particularly through pre-processing, feature extraction, and feature selection, is essential to ensure the performance of classification models [10]. Pre-processing techniques, such as tokenization, stop-word removal, and stemming, reduce the data's complexity and improve model accuracy by eliminating noise. Feature extraction creates a compact feature space by transforming the original data, while feature selection identifies a subset of relevant features that distinguish different categories [11]. These techniques have a profound impact on model accuracy and efficiency but often struggle with the high dimensionality inherent in text data [12].

Traditional dimensionality reduction methods, such as k-means clustering [13], two-stage feature selection [14], and hybrid approaches combining ReliefF and principal component analysis [15], aim to address these challenges. However, these methods may not fully integrate semantic context into the feature representation, limiting their impact on classification performance. To address these limitations, this study introduces Feature Substitution using Latent Dirichlet Allocation (FS-LDA), a novel technique that combines dimensionality reduction with semantic feature grouping.

FS-LDA leverages the topic modeling capabilities of Latent Dirichlet Allocation (LDA) to group and substitute high-probability topic words into unified representations, reducing dimensionality while preserving meaningful textual features [16]. Unlike feature selection, which eliminates irrelevant features, FS-LDA substitutes related features based on topic modeling, enhancing the representation of the data for classification tasks. A new term called feature substitution is introduced mainly to replace related features according to defined groups from a topic modelling technique. Previous studies have demonstrated the effectiveness of LDA in dimensionality reduction and topic clustering, but its application in feature substitution remains unexplored [16]. By integrating FS-LDA into the pre-processing phase, this study seeks to evaluate its effectiveness in improving classification accuracy and efficiency across various datasets.

The FS-LDA technique offers a significant advantage by reducing feature complexity while maintaining the semantic integrity of the data. This novel approach simplifies input data preparation and enhances the performance of classification

models, as demonstrated through experiments in this study. The findings highlight FS-LDA's potential as a scalable, efficient, and effective method for text classification tasks in real-world applications.

II. Literature Review

The current technological advancement and new research on machine learning over the years contribute tools to deal with a high volume of documents using algorithms that extract information from their original texts. One possible approach to simplify high-volume data is to apply some form of dimensionality reduction. Methods like feature extraction and feature selection offer distinct benefits; feature extraction transforms the original data into a compact feature space, while feature selection retains only the most relevant features, potentially improving model efficiency. Commonly, researchers used n-grams models such as unigram, bigram, and trigram to extract features. Linguistic pattern methods, statistical methods, or a combination of both can enhance the extraction process. Hybridization between a linguistic approach and a statistical method efficiently provides reliable features while improving accuracy, especially in classifying Arabic text [17].

Meanwhile, some researchers preferred to enhance the feature selection technique used in their study to improve classification rates. For instance, previous researchers used collaborative feature-weighted multi-view fuzzy c-means clustering [18] and hybrid binary grey wolf with harris hawks optimizer [19]. The utilization of both techniques accordingly provides a better data pre-processing process. However, these methods often lack a semantic perspective, which is addressed by techniques like Latent Dirichlet Allocation (LDA). Over the years, LDA has been widely used to explore features using a hidden topic analysis [20]. It is known as a classical statistical model for topic mining in natural language processing, and it was proposed by Blei et al. [21]. This model discovers various topics in many documents and builds to model text data subject information. Many domain retrievals involving machine learning models applied the LDA model to help deal with text-related problems [22]. Besides the LDA model, researchers often used another topic modelling approach, Latent Semantic Analysis (LSA) [23]. The model's weaknesses are its dependency on annotated training data and its tendency to overfit. Hence, LDA is often preferred over LSA due to its ability to handle sparse data and its probabilistic nature, which provides a more robust representation of text semantics. This advantage aligns with the study's objective to enhance text classification through semantically enriched feature substitution.

The LDA structure resembles the probabilistic variation of LSA known as Probabilistic Latent Semantic Analysis (PLSA) [24]. While LDA and its predecessor, Probabilistic Latent Semantic Analysis (PLSA), share probabilistic foundations, LDA's use of Dirichlet priors enables better generalization for unseen documents, addressing a critical limitation of PLSA and advancing its utility in text classification tasks [24]. The model learns a distribution over the topic for each document in training, but it is only applicable for training sets with the known topic distribution. The model cannot generate topics from previously unseen documents. Meanwhile, the LDA model learns topic distribution as a random parameter vector and models based on Dirichlet prior. Researchers use symmetric Dirichlet distribution involving a similar value for all parameters in the LDA. The derivation methods commonly acquire the distributions are a variational inference [17] and Gibbs sampling [25].

Previous studies have successfully proved the efficiency and benefits of practicing this model. From the beginning, Blei et al. [21] discovered that LDA slightly decreases text classification performance but improves overall efficiency because of its dimensionality reduction characteristic. Researchers invented an LDA-based model known as Dual Latent Dirichlet Allocation (DLDA) to extract topics for short texts with knowledge obtained from long text data [26]. The improved model utilizes two sets of LDA topics where "target" and "auxiliary" represent short and long texts. The DLDA model performs better than the LDA model, primarily in clustering short text data based on entropy, purity and normalized mutual information as the evaluation criterion.

A previous study can merge the document's representation based on the LDA by applying labels to enhance text classifier performances [27]. The modified LDA works as a semi-supervised learning model where the model includes partial expert knowledge at word and document levels. There is accuracy rate improvisation as more documents are labelled. The modified LDA is feasible for real-world applications with many unlabeled data with few labelled data for training purposes. On the other hand, Cheng et al. [28] combine the idea of using the LDA and word co-occurrence patterns in the corpus to detect topics for a document. It addresses co-occurrence, such as bi-term individually as a semantic unit representing a single topic for recognizing the words most likely to be together. The LDA with word co-occurrence patterns combination improves the topic selection consistency for each document.

A study also merged the LDA with clustering through a Self-Aggregation based topic model (SATM) [29]. The proposed model helps detect relevant topics in short text data. A Multi-CoTraining (MCT) system implementation through LDA combination with Term Frequency-Inverse Document Frequency (TF-IDF) and Doc2Vec provides various feature sets for document classification [30]. The proposed model is robust when dealing with parameter changes. The performance of MCT is superior compared to other benchmark methods. Instead of using Doc2Vec, another study presents the combination of Word2Vec as the word embedding technique with the LDA [31]. The experimental result shows the proposed model outperforms the basic LDA. It can solve problems created by a Bag-of-Word (BOW) model related to high dimensionality and sparsity data.

An automatic text mining framework based on the LDA is proposed in the financial sector to analyze texts as financial disclosures from firms [32]. The topic model aims to find a firm's strengths and weaknesses through business units, activities, and processes depending on its risk. The proposed framework helps to improve the existing business management tools regardless of any business level. The LDA is also an

alternative representation model for BOW because it reduces the feature numbers for text classification [33]. The WEKA package has included the framework to provide a feasible option for other researchers to select features from their data sets.

LDA is used as a feature selection technique in Celard et al. [33] to create a new text representation model utilizing the probability of a document belonging to each topic. However, the probability is not yet used to substitute existing features extracted from classic representation models such as unigram and bigram models. The utilization of LDA topics in the feature selection process can greatly reduce the input data dimensionality while improving the classification model performance. Hence, this study's main objective is to assess the LDA model's efficiency in text classification as a feature substitution technique.

III. Methodology

This section briefly describes the proposed framework used in this study. The detailed description of the feature substitution technique provides a better understanding of the proposed technique for data preparation related to features. This study used HMM as the text classification model.

A. Proposed Framework

The study involves several steps before classifying the data, as shown in Fig. 1. The typical steps are data extraction, data pre-processing, feature extraction, and feature selection. These are the necessary steps in data preparation related to text classification. After data extraction, three pre-processing involve tokenization, stop word removal, and stemming [34]. Data pre-processing is vital to ensure the data is standardized and in proper form. The standardized way is achieved after applying the three pre-processing steps, where each observation is tokenized into words at first. Then, stop words are removed from the word list, and the remaining words are stemmed to ensure the words follow the root word forms.

Feature extraction and selection are essential to ensure the data are well transformed into significant and functional features before performing the classification process [35]. The choice of features may affect the classification model accuracy. Thus, the study compares two feature representation models, i.e., unigram and bigram, to observe their effect on classification performance. The feature selection used in the study was the filter method known as correlation-based feature selection (CFS). The study also compares the classification model before and after applying the proposed feature substitution technique. Then, the chosen features are used as inputs to perform the classification model. All the input data preparation steps were computed using R-Programming software. The classification step is done using Python programming software.

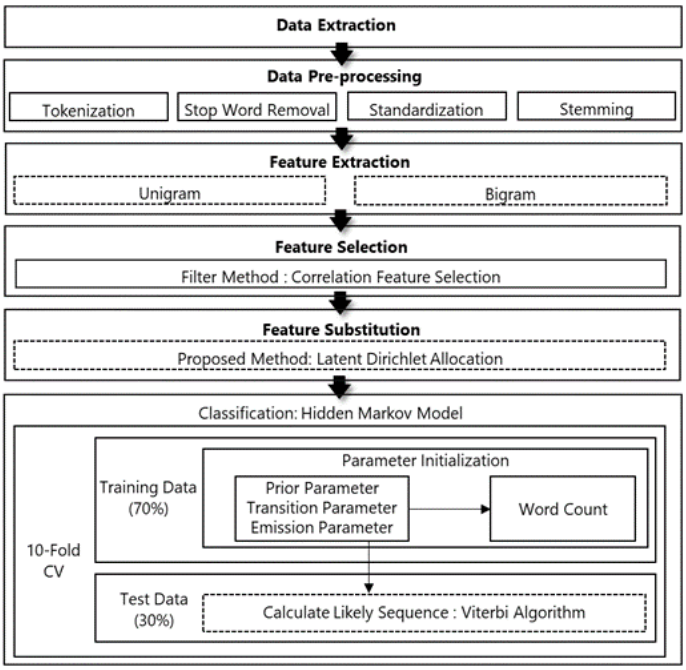

Fig. 1. Proposed study framework.

B. Feature Substitution using Latent Dirichlet Allocation

All the steps involved are standard procedures in text classification before training the classifier, except for applying the LDA model to perform the feature substitution. It is a generative probabilistic model of a document collection [15]. LDA searches for these latent semantic topics in the corpus [35], and it considers each document as a topic collection where each topic is a keyword collection. The topics are a collection of dominant keywords. These topics express an approach to quantitatively describe the document and describe the document content [36]. The critical factors in obtaining adequate keyword segregations are the text processing quality, topic diversity in the text, algorithm selection, and algorithm tuning.

The LDA algorithm input is basic units of discrete data, i.e., words in the text documents. The output of the LDA algorithm is a set of topics. For instance, each document's category belongs to an extensive collection of words, and documents can be observed by checking the words' occurrence in the documents. However, this method is costly and inefficient. Instead of checking every word in the document, another layer is initiated with a set of topics. The collection of words is mapped to the topics, and the topics are mapped to the documents. Hence, this action will reduce costs while increasing efficiency.

The study used LDA to group and substitute the features before applying the classification model. This model involves a generative process assuming that documents consist of a mixture of topics. Then, words from the typical vocabulary of each selected topic are drawn from each document. In the study, the document in topic modelling is represented by observation. Accordingly, LDA assumes that observations are described as a bag of words in a unigram or bigram model with different topics in different proportions. The pseudocode for the

proposed feature substitution technique using the LDA is shown in Algorithm 1.

Algorithm 1: Feature Substitution Technique using LDA

Initialize
O: Observations in the dataset
O_0: The first observation
T: Topics in the observation
W: Word in the observation
P: Percentage of the highest probabilities in a topic

Compute
Assign each W in O_0 a topic T

While (observation remain) do
  For each W in O_0 do
  Assume the assigned topic T is wrong.
  Assume the assigned topic T for other W in O_0 is correct.
    Update and analyze
      Calculate the probabilities to assign a topic T based on:
      Number of topics in the document.
      Number of times the same topic is assigned to the word across all document.
  End
  End
Repeat the process for all O
Remove overlap words with percentage P in all T
  For each T
  Assign a new topic name to W
  End

The calculation involves in LDA is to obtain the probability of words belonging to a topic where the procedure starts with randomly assigning each word in the observation O to one of T topics. Then, the required probabilities of each word, W can be computed after assuming the randomly assigned topic for that particular word is wrong. The computation of the first probability involves the proportion of words in observation O that are assigned to the topic T. This action is to observe how many words belong to the topic T for a given observation O excluding the current word W. If many words from observation O belongs to topic T it is more probable word W belongs to topic T.

The second probability involves the proportion of assignments to topic T out of all documents derived from the word W. This action is to observe how many observations are in topic T because of the word W. LDA represents documents as a collection of topics. A topic is also a collection of words. If a word has a high likelihood of appearing in a topic, all observations containing W will also be more strongly correlated with T. Similarly, if W is not very likely to be in T, documents including W will have a very low likelihood of being in T, because the rest of the words in O will belong to a different topic, giving O a higher probability for other topics. Even if W is added to T, it will not bring many of these observations to T. The probability that a W in observation O belongs to topic T is stated in Eq. (1).

$$P(T \mid W, O) = \frac{m \text{ of word } W \text{ in topic } T + \beta}{\text{total tokens in } T + \beta} \quad (1)$$

m represents the words in O that belong to T, adjusted by the hyperparameter α. The parameter α controls how topics are distributed in a document: a smaller α focuses the document on fewer topics, while a larger α mixes more topics evenly. Similarly, β manages the distribution of words within topics. A smaller β emphasizes a few dominant words, making topics more distinct, while a larger β spreads probabilities across many words, resulting in broader topics. Although each topic technically includes all words in the vocabulary, the most probable words define the topic, making it both meaningful and flexible.

After evaluating each word's probability belonging to different topics based on the LDA model, the subsequent action is to substitute the non-overlap words with high probability in each topic. According to the LDA model analysis, these words become homogeneous by assigning the same name to represent the group of words that most probably belong to the topic. For example, Fig. 2 shows that Observation 1 is only about Topic 1.

In contrast, Observation 2 is a mix of Topic 1 and 2 because one of the words, "banana", has a higher probability value in Topic 1 than in Topic 2. Specifically, each topic is represented as a probability distribution over a controlled vocabulary. Usually, all the words appear in the observation collection. In the example, Topic 1 has words such as "fresh" (3.41%), "drink" (2.35%), and "juice" (1.99%). Meanwhile, Topic 2 has words such as "biscuit" (2.73%), "mix" (2.21%), and "apple" (1.86%). These words are the three highest probabilities in each topic. Given this information, Topic 1 can be labeled as "drink" and Topic 2 as "food". Consequently, Observation 1 is purely about "drink", while Observation 2 is a mix of the "drink" and the "food" topics. The only observable variable is words from the observations, whereas all other variables, such as the topic distributions for each document and the word distributions for each topic, are hidden. Hence, LDA aims to infer these hidden distributions, given the observed words per observation.

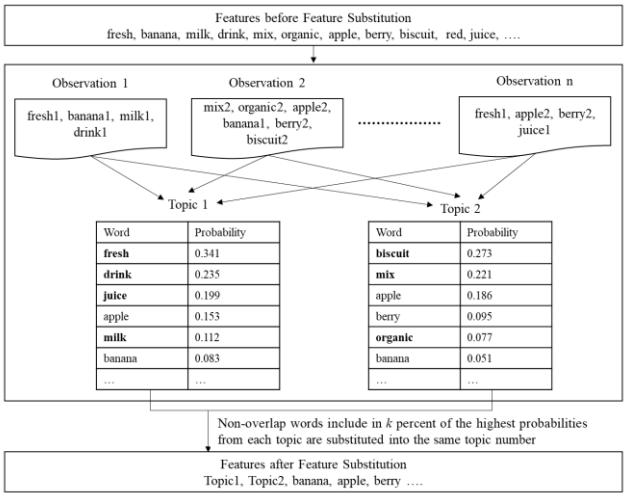

Fig. 2. Proposed feature substitution technique for input data preparation in text classification.

After applying LDA, each topic is represented by words with specific probabilities of belonging to that topic. The feature substitution technique replaces high-probability, non-overlapping words from each topic with a single constant term, such as "Topic1" or "Topic2," ensuring that the selected words uniquely represent their topic. The study tested this substitution at different levels (10%, 20%, 30%, 40%, and 50% of the top words per topic). Fig. 3 illustrates how this technique represents data before applying the classification model, along with examples from a sample dataset.

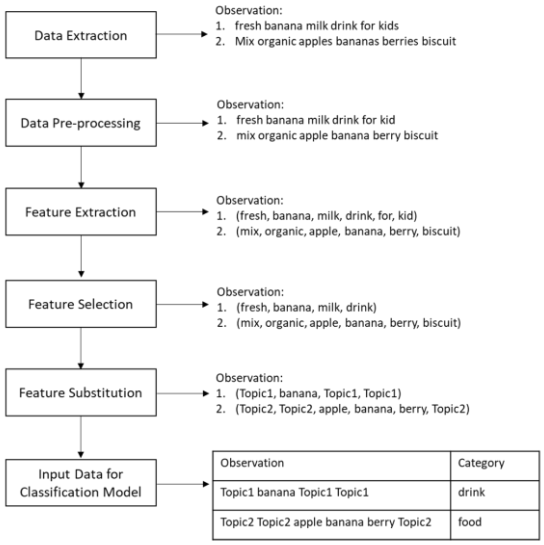

Fig. 3. Sample text representation with proposed feature substitution technique before applying classification model.

C. Classification

A Hidden Markov Model (HMM) is often applied to text classification as a supervised learning task. The application of HMM can be seen through various study areas related to text and language processing applications, e.g., text classification [37], text discretization [38], and information extraction [35]. The input data used for the supervised learning model is a corpus of words labeled with the correct category. Table I shows the components that specify an HMM.

TABLE I. HIDDEN MARKOV MODEL COMPONENTS

Symbol	Component	Description
Q	$q_1 q_2 \dots q_N$	A set of N states
A	$a_{11} \dots a_{ij} \dots a_{NN}$	A transition probability matrix A, each a_{ij} represents the moving probability from state i to state j
o	$o_1 o_2 \dots o_T$	A sequence of T observations, each one is drawn from a vocabulary $V = v_1, v_2, \dots, v_v$
B	$b_i(o_t)$	A sequence of observation likelihoods, also called emission probabilities, each expressing the probability of an observation o_t being generated from a state i
π	$\pi_1 \pi_2 \dots \pi_N$	An initial probability distribution over states. π_i is the probability that the Markov chain will start in the state i

HMM's decoding problem is finding the optimal state sequence given the observation sequence and the trained HMM. The Viterbi algorithm is commonly applied to find the most likely hidden state sequence based on every word sequence input. There are given the observation sequence for test data $\{o_t\}_{t=1}^{N}$ and trained HMM with parameters $\lambda = (\pi, A, B)$ to find the most likely sequence. The formula is presented in Eq. (2).

$$\underset{\{q_t\}_{t=1}^{N}}{argmax}\ p(\{q_t\}_{t=1}^{N} | \{o_t\}_{t=1}^{N}) \quad (2)$$

The optimal hidden state sequence is produced for each word sequence of the test data using the Viterbi decoding algorithm. The prediction of the text data is based on the majority role, i.e., a product will be labeled as the drink category if the optimal hidden state sequence has more drink features than food features. Otherwise, the product is marked under the food category.

IV. DATASETS

This study utilizes two different text data to evaluate classification models' performance. The first data involves five e-commerce product data, which these datasets are crawled from an e-commerce website. Department of Statistics Malaysia (DOSM) has collected product information from one of the primary online store websites through the STATSBDA project known as Price Intelligence (PI) using its prototype web scraper. Another dataset is retrieved from the UCI repository. Table II presents a summary of all the datasets used in the study.

TABLE II. SUMMARY OF DATASETS

Data Name	Data Description	Class Number	Class Name (Instance Number per Class)	Instance Number
ECD01	E-Commerce pets products	2	food (265) and care & accessories (45)	310
ECD02	E-Commerce non-food products	2	cooking & dining (407) and party accessories (80)	487
ECD03	E-Commerce frozen food products	5	frozen food (291), yoghurt (162), ice cream (147), cheese (85), and juices (87)	772
ECD04	E-Commerce household products	6	laundry (370), air freshener (297), household kitchen cleaner (181), sundries (158), light bulbs (100), and toilet cleaner (100)	1206
ECD05	E-Commerce grocery products	14	cooking ingredients (677), chocolates & sweets (594), biscuits & cakes (491), snacks (440), sauces & dressings (364), canned food (331), pasta & instant noodles (294), baking (269), jam (220), cereals (208), dry condiments (206), sugar & flour (176), rice (138), and cooking oil (130)	4538
SPAM	SMS spam collection data set	2	ham (4827) and spam (747)	5574

A. Dataset's Characteristics

Each dataset's characteristics can be seen through its data distribution. Text length, word count, and class distribution can describe the data. The detailed characteristics are shown in Fig. 4 for each data set correspondingly. In class distribution for e-

commerce product datasets, ECD01 and ECD02 fall under binary classification problems. However, these two datasets have different text characteristics, as shown in Fig. 4. There are two dominant features in the ECD01 dataset, i.e., “food” and “cat”. Other features seem to have not much different frequent existences in the dataset compared to these two features. Meanwhile, there are six dominant features in ECD02, whereas other features are far less number of occurrences in the dataset. The variation of dominant features may affect a classification model, especially when using HMM because the parameter estimation is based on feature occurrences.

Three e-commerce product datasets, i.e., ECD03, ECD04, and ECD05, belong to multiclass classification problems. Usually, datasets with a higher number of classes tend to have a much lower classification model performance because of increased data complexity. ECD03 has a higher number of dominant features than the other two datasets. When a dataset has less prevalent features, such as features in ECD04, there is a tendency that is performing the proposed feature selection technique may not significantly reduce the number of features while improving the model performance. The reason is that the proposed model recognizes a group of features to be combined as one topic where the features must not belong to any pre-defined classes. The relatively similar number of occurrences for each feature in the dataset may emphasize that the features may have equal weight pertaining to any hidden topic created to reduce the features. Hence, there is an assumption that any dataset with a high number of dominant features may be beneficial for using the proposed feature substitution technique.

The text length plots represent the product description distributions for ECD01, ECD02, and ECD04, are appear to have an approximately normal distribution. Meanwhile, ECD03 and ECD05 have shorter product description lengths as their distribution is right-skewed. Typically, the term frequency distribution is based on the number of times a feature appears in a dataset divided by the total number of features in that dataset. Both axes are plotted on logarithmic scales in the term frequency distribution plot because the frequency of the most frequent features is much higher than the frequency of the long tail of infrequent features that a figure of this size without a logarithmic transformation would look like the letter L.

The frequency distribution plots for all e-commerce product datasets illustrated the frequency curve decreases very steeply from the extremely high values corresponding to the most frequent features. They become progressively flattered until they reach an extensive level corresponding to the ranks assigned the tail of words occurring once. The same skewed shape is not specific to the datasets used in this study. Still, it often emerges in natural language texts, independently of tokenization or type mapping method, size, language, and textual typology [39]. The only difference is that the variation of inflected forms can be seen from the frequency distribution plots. Even though the overall pattern is the same, the number of very low-frequency forms in the three datasets, i.e. ECD01, ECD02, and ECD03, is lower than in the other two datasets.

The ordinary skewed structure of word frequency distributions was first comprehensively studied by Zipf [40]. The utilization of various datasets leads to frequency’s nonlinearly decreasing rank function. Theoretically, the high ranks fall more sharply than the low ranks. Fitting a straight line to the log-log curves is commonly rational and practicable. Fig. 5 visualizes the frequency distribution plot for each dataset according to Zipf’s law. The plots are generally not perfectly fitted, especially at the edges. The curve's right edges represent features among the highest ranks with the lowest frequencies. The inconsistent patterns are because the increasingly more comprehensive horizontal lines, in accord with the rare words, are assigned different ranks but have the same frequency. The results may happen due to fitting a model consisting of many words with very near-continuous frequencies to an empirical curve, originally a discrete step function for high ranks.

Meanwhile, the left plot’s curved edges represent features among low ranks with high frequencies. Each plot portrayed a different degree of downward curves.

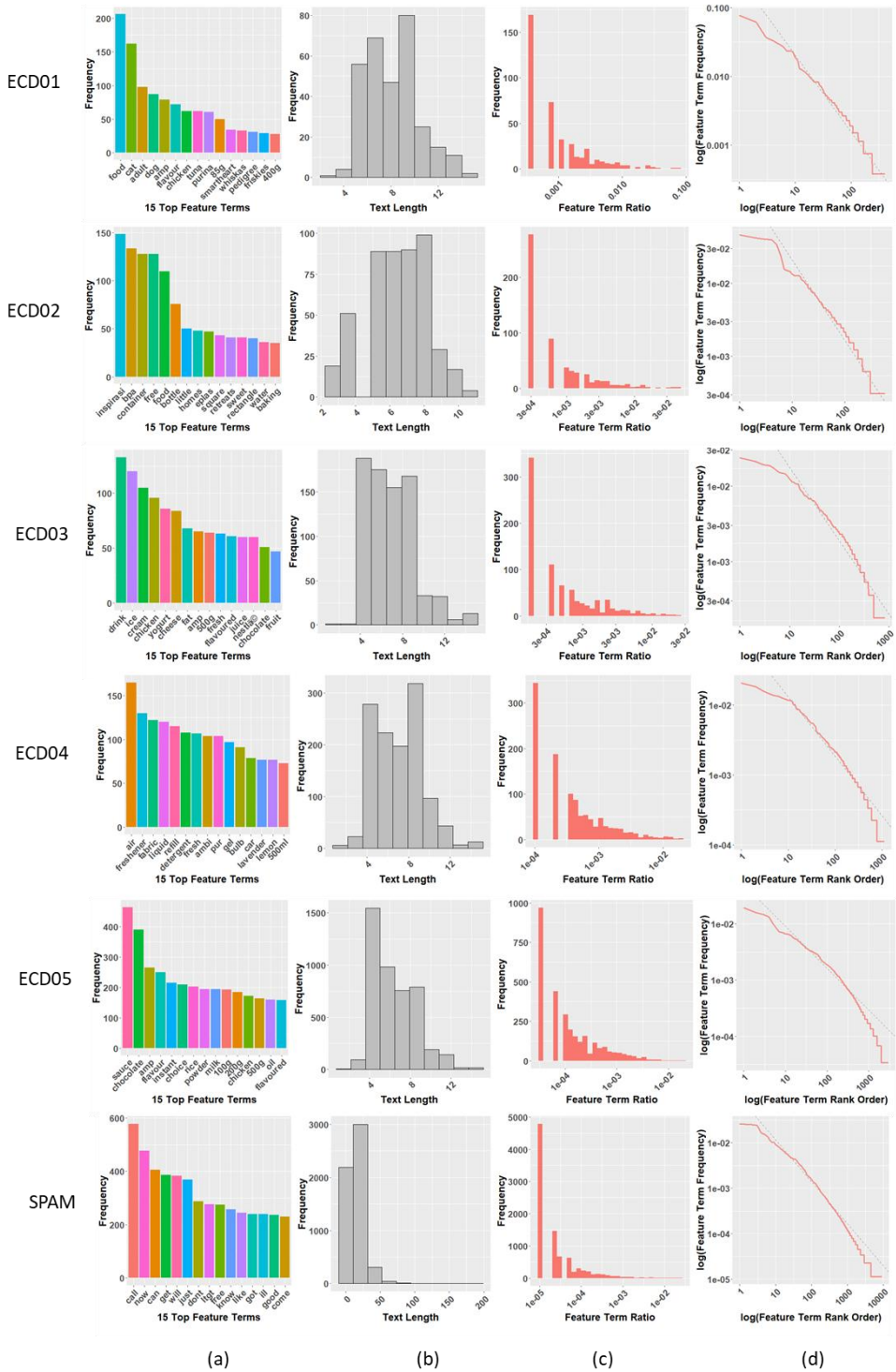

Fig. 4. Text characteristics and feature term distribution according to (a) 15 Top features, (b) Text length plot, (c) Term frequency distribution plot, and (d) Zipf's law distribution plot.

However, the curve falling under the fitted lines depicted features with high frequencies tend to be lower than predicted by their rank relative to Zipf's law. Natural language text distributions typically have similar overall patterns of a few very high-frequency types and long tails of infrequent words. The difference can be spotted through detailed observation of specific inconsistent parts in a frequency distribution plot. For example, ECD01 and ECD02 may imply the same frequency distribution plot, but according to Zipf's law, the distribution varies, especially in explaining the features among low ranks with high frequencies. Hence, each e-commerce product dataset implied typical text distributions, yet they may encounter different classification performance results.

On the other hand, there is a noticeable difference between text characteristics for the SPAM dataset and e-commerce product datasets. The former dataset showed a right-skewed text length distribution because messages have longer texts than e-commerce product descriptions. Meanwhile, the frequency distribution plot illustrated that the frequency curve decreased more steeply and quickly flattered than frequency distribution plots for e-commerce product datasets. This pattern implied that many features in the dataset might not be frequently used. Some of the features only occurred once when processing text from messages.

In addition, Zipf's law distribution plot for the SPAM dataset closely follows the fitted line. The model predicts a very rapid decrease in frequency among the most frequent words, which becomes slower as the rank grows, leaving very long tails of words with similar low frequencies. Contrary, e-commerce product descriptions tend to utilize similar features across different categories and, at the same time, use particular features to describe products in a category. Hence, the text distributions for e-commerce products differed from the SPAM dataset. The study utilized both datasets to show the effectiveness of the proposed model.

B. Feature Reduction

Each dataset had been through all the pre-processing data procedures. Two feature extraction techniques, i.e., unigram and bigram, are used to extract the features. Then, the features from each set are selected using a correlation-based feature selection (CFS). It is a well-known filter method widely used in previous studies [10]. The features were also chosen using feature substitution by Latent Dirichlet Allocation (FS-LDA) with 10%, 20%, 30%, 40%, and 50% of each class's highest probability features. Table III shows the number of features used as the input data for HMM using different feature extraction techniques and feature substitution involvement in the model.

TABLE III. NUMBER OF FEATURES FOR EACH DATASET

Feature Extraction	Data	CFS	FS-LDA				
			10%	20%	30%	40%	50%
Unigram	ECD01	304	250	194	202	220	224
	ECD02	461	383	301	275	325	345
	ECD03	656	497	508	524	529	561
	ECD04	941	919	914	901	887	860
	ECD05	2630	2054	2189	2310	2393	2443
	SPAM	5903	4788	3641	3210	3847	4171
Bigram	ECD01	734	702	639	638	653	656
	ECD02	1072	1027	962	921	961	943
	ECD03	1934	1864	1832	1781	1738	1788
	ECD04	2789	2781	2779	2772	2756	2659
	ECD05	10852	10554	10407	10333	10341	10246
	SPAM	30349	29299	27531	26624	27102	26981

Fig. 5 and Fig. 6 show the number of features for some datasets does not decrease with the percentage increment of features from the highest probability in each class. The selected features to be substituted differ for each percentage where the overlap features are not replaced. The feature substitution by 10% shows features decreasing regardless of any datasets used in the study. Then, the increment of 20% shows irregular patterns in unigram representation models.

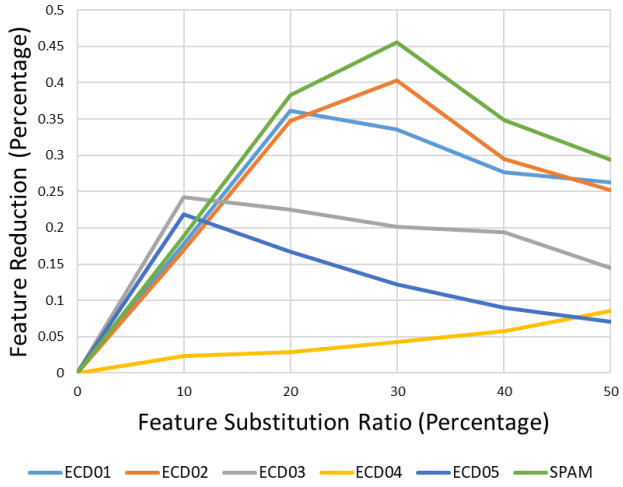

Fig. 5. Unigram feature reduction percentage for each dataset.

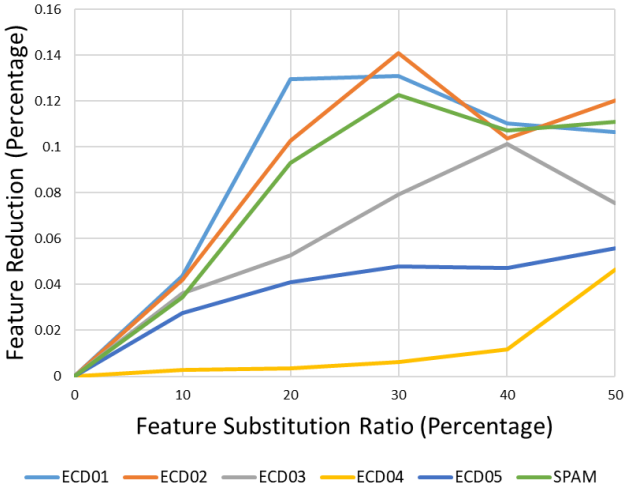

Fig. 6. Bigram feature reduction percentage for each dataset.

The feature number for ECD03 and ECD05 is greater than the feature number reduced by the 10% FS-LDA for each dataset. The irregular pattern for bigram models is only noticeable when the feature substitution by 30% is applied to the datasets. All datasets using the bigram model for feature extraction showed a lower performance increase than the unigram model. When using the bigram model, features extracted from a dataset become more specific, and each feature's representativeness differs from the unigram model. The same feature occurrences decrease drastically with the increase of unique features through the Bigram model. The inclusion of various features with minimal occurrences leads to poor LDA estimation on features belonging to particularly one hidden topic. Hence, the feature reduction percentage becomes smaller than expected while not being able to increase the model performance efficiently.

Nonetheless, using FS-LDA in preparation for classifying data using HMM did not jeopardize the model performance.

Regarding data reduction consistency, 10% of each topic's highest probabilities of non-overlap words seems like a good percentage to be used in general. However, the feature sets' performance was analyzed to prove that the proposed model is useful for reducing data dimensionality while improving a classifier's accuracy.

V. Experimental Results and Discussion

This section presents the results of the experiments conducted and discusses the findings in the context of the proposed framework. The analysis evaluates the effectiveness of the feature substitution technique using FS-LDA in reducing dimensionality and its impact on text classification performance.

A. Feature Reduction

This study utilized two performance measurements, namely macro F1-score and weighted F1-score. The micro F1-score is not used in the study because all classification decisions in the dataset are considered without class discrimination when using this approach. Contrary, the macro F1-score is computed for each class within the dataset. Its average score calculation is based on the overall classes. In this way, class distributions in the training set are disregarded, and equal weight is assigned to each class. The formulas are presented in Eq. (3) - Eq. (7). *S* is the set of classes or states, *TP* is the number of true positives, *TN* is the number of true negatives, *FP* is the number of false positives, and *FN* is the number of false negatives.

Meanwhile, the weighted F1-score is represented because this approach considers class imbalance [1]. Hence, the study observed the difference when the average score calculation for macro F1-score is based on each class's weight. The formula for the weighted F1-score is presented in Eq. (8).

$$accuracy_s = \frac{TP_s+TN_s}{TP_s+FP_s+TN_s+FN_s} \quad (3)$$

$$precision_s = \frac{TP_s}{TP_s+FPs} \quad (4)$$

$$recall_s = \frac{TP_s}{TP_s+FN_s} \quad (5)$$

$$f_s = 2 \cdot \frac{precision_s \times recall_s}{(precision_s+recall_s)} \quad (6)$$

$$macro\ F1-Score = \frac{\sum_{s\in S} f_s}{size\ (dataset)} \quad (7)$$

$$weighted\ F1-Score = \frac{\sum_{s\in S} f_s \times size(s)}{size\ (dataset)} \quad (8)$$

B. Results for E-Commerce Product Data

The proposed model's effectiveness (FS-LDA) was observed based on its performance in classifying five e-commerce product data. The data involved binary and multiclass classification using HMM. Table IV presents the macro F1-scores for these datasets. According to the results, the unigram model application for extracting the features enhanced the HMM performance compared to the bigram model regardless of the feature substitution existence. The macro F1-score for HMM with correlation-feature selection (CFS) seemed to increase when substituting 10% and 20% of the ECD01 and ECD04 data features with the unigram model. Meanwhile, the feature substitution worked the best when using 10% FS-LDA for ECD02, ECD03, and ECD05 data with the unigram model.

TABLE IV. Macro F1-Scores for E-Commerce Datasets using HMM

Feature Extraction	Data	CFS	FS-LDA				
			10%	20%	30%	40%	50%
Unigram	ECD01	0.6346	**0.7866**	0.7366	0.6947	0.6913	0.6531
	ECD02	0.8227	**0.8695**	0.8679	0.8632	0.8643	0.8437
	ECD03	0.6685	**0.7402**	0.7212	0.6733	0.6690	0.6469
	ECD04	0.6431	0.6454	**0.6470**	0.6449	0.6279	0.5994
	ECD05	0.5421	**0.5935**	0.5503	0.5259	0.5225	0.5097
Bigram	ECD01	0.4236	**0.4650**	0.4630	0.4306	0.4489	0.4560
	ECD02	0.4940	**0.5038**	0.4932	0.4928	0.4940	0.4940
	ECD03	0.2900	**0.3663**	0.3368	0.3573	0.3100	0.3412
	ECD04	0.2892	0.2902	**0.2913**	0.2909	0.2903	0.2985
	ECD05	0.2748	0.2856	0.2942	**0.3022**	0.2979	0.3053

On the other hand, Table V shows the weighted F1-scores for e-commerce product data. Like the macro F1-scores results, HMM with the unigram model was preferable rather than the bigram model to extract features for classifying these data. The HMM model for each data was similar to results obtained using macro F1-scores. However, the only difference is that the weighted F1-scores produced higher scores than macro F1-scores. A macro F1-score is most useful if there are many classes in the data and the researchers are interested in the average F1-score for each class.

Meanwhile, weighted F1-scores are influenced by the proportion for each class in the dataset. The score works well for observing the dataset's classification performance for unequal classes. Even though this score provides an alternative score for imbalanced dataset performance, a large weighted F1-score might be slightly misleading for a highly imbalanced dataset because the majority class overly influences it.

For example, the macro F1-score for ECD02 using CFS and 10% FS-LDA of the bigram HMM model was 0.5038 compared to the weighted F1-score value of 0.7805. The proportion of classes in Table II for ECD02 indicated that the dataset consists of 83.57% product descriptions for the cooking and dining category and only 16.43% product descriptions for party accessories. Hence, a noticeable difference in these two F1-scores was due to a highly imbalanced dataset. The inclusion of both scores was to observe the impact of the imbalanced dataset towards F1-scores as most of the datasets in the study are imbalanced datasets. However, both scores are equally acceptable according to the final goals of the study. The proposed feature substitution technique improves HMM performance according to both macro and weighted F1-scores.

TABLE V. WEIGHTED F1-SCORES FOR E-COMMERCE DATASETS USING HMM

Feature Extraction	Data	CFS	FS-LDA				
			10%	20%	30%	40%	50%
Unigram	ECD01	0.7459	**0.8764**	0.8431	0.8016	0.8027	0.7729
	ECD02	0.9104	**0.9327**	0.9325	0.9294	0.9306	0.9210
	ECD03	0.6936	**0.7729**	0.7486	0.6927	0.6890	0.6664
	ECD04	0.6815	0.6817	**0.6853**	0.6807	0.6618	0.6425
	ECD05	0.5365	**0.5892**	0.5385	0.5109	0.5122	0.5049
Bigram	ECD01	0.4868	**0.5427**	0.4963	0.5411	0.5216	0.5312
	ECD02	0.7767	**0.7805**	0.7759	0.7756	0.7767	0.7767
	ECD03	0.2626	**0.3474**	0.3046	0.3313	0.2626	0.3186
	ECD04	0.3603	0.3621	**0.3626**	0.3623	0.3615	0.3748
	ECD05	0.2563	0.2684	0.2765	**0.2839**	0.2814	0.2874

The percentage of feature substitution that worked best for each dataset differed due to their text characteristics and distributions. The results encountered two situations: the HMM model performance suddenly dropped at a certain percentage of FS-LDA, or the model performance did not show any promising result throughout the FS-LDA. For example, for situation one, the model performance for ECD03 highly increased when using 10% FS-LDA, but the performance started dropping when using 20% FS-LDA. This situation occurred because substituting 20% from the highest probabilities from each topic disturbed the overall performance by decreasing the critical features used in the dataset to set the boundary of recognizing different categories.

Meanwhile, the second situation can be described through model performance for ECD04. The model did not show promising performance improvement regardless of any percentage of FS-LDA due to a highly similar number of features existing in the dataset, as shown in Fig. 5, compared to other e-commerce datasets. However, the model performances for all datasets kept increasing when using 10% FS-LDA compared to standard HMM. Hence, the study found that FS-LDA with 10% feature substitution for non-overlap words of the highest probabilities from each topic can reduce the data dimensionality while increasing the HMM performance. The higher feature substitution percentage may harm model performance. Previously, LDA was proven to enhance a supervised learning model [41]. This study supported the literature on discovering LDA potentiality for data reduction in text classification.

Despite showing macro or weighted F1-scores, Table VI presents the straight-forward model performance evaluation using model accuracy between several text classifiers, including HMM, HMM with 10% FS-LDA, Naïve Bayes, and Support Vector Machine. HMM with 10% FS-LDA outperformed most model performances for e-commerce datasets except for ECD01 when using the unigram model. Concurrently, the proposed model performed the best for ECD04 and ECD05 when using the Bigram model.

Support Vector Machine and Naïve Bayes outperformed the proposed model performance for ECD01. These two classifiers are known for their excellent performances in solving binary classification problems without interfering with uncommon feature distributions such as ECD01. When dealing with data such as ECD01, the proposed model seemed to improve the performance of standard HMM. However, combining improvisation from the feature substitution technique presented in the study with enhancing theory in developing a better HMM model may outperform the other two classifiers.

TABLE VI. ACCURACY RATE COMPARISON BETWEEN HMM, HMM (10% FS-LDA), NAÏVE BAYES AND SUPPORT VECTOR MACHINE

Feature Extraction	Data	HMM	HMM (10% FS-LDA)	Naïve Bayes	Support Vector Machine
Unigram	ECD01	0.7009	0.8604	0.9434	**0.9774**
	ECD02	0.9181	**0.9367**	0.8343	0.8518
	ECD03	0.6449	**0.7422**	0.3487	0.3529
	ECD04	0.6805	**0.6808**	0.3489	0.3856
	ECD05	0.5156	**0.5523**	0.2196	0.2340
Bigram	ECD01	0.4377	0.4462	0.9519	**0.9811**
	ECD02	0.8410	**0.8428**	0.8072	0.8392
	ECD03	0.2563	0.2688	0.3221	**0.3312**
	ECD04	0.4019	**0.4034**	0.3297	0.3684
	ECD05	0.2065	**0.2152**	0.1965	0.2113

C. Results for Spam Data

The proposed technique presented in this paper can be applied to other kinds of text data. The study utilized a well-known benchmark data, SMS spam data collection, to evaluate its performance in a different text data application. Based on the experimental results in Table VII, the F1-scores for unigram models were better than bigram models for all HMM models. Even though there were improvements when applying all percentages of FS-LDA for both feature extraction models, the HMM model performance started to drop when using 50% and 40% FS-LDA for unigram and bigram models, respectively. Like results from e-commerce product datasets, some of the HMM model performance improvement can be up to 40% FS-LDA. However, the only similarity between the results was that the model performance increases when applying 10% FS-LDA regardless of any datasets. Hence, it can be concluded that there are stable improvements in HMM performances using 10% FS-LDA in classifying the e-commerce product and SMS spam datasets.

TABLE VII. PERFORMANCE RESULTS FOR SPAM DATASET

Metrics	Feature Extraction	CFS	FS-LDA				
			10%	20%	30%	40%	50%
Macro F1-score	Unigram	0.687	0.745	0.777	**0.787**	0.742	0.710
	Bigram	0.668	0.674	0.704	0.888	0.695	0.685
Weighted F1-score	Unigram	0.873	0.893	0.905	**0.909**	0.892	0.881
	Bigram	0.870	0.872	0.883	0.719	0.880	0.876
Accuracy	Unigram	0.894	0.906	0.914	**0.917**	0.904	0.898
	Bigram	0.898	0.899	0.906	0.909	0.904	0.900

The HMM model performed the best by securing an accuracy of 90.56% to classify spam and ham SMS considering 10% FS-LDA as the optimum HMM model across different datasets. Both model precision and recall increased when applying the proposed technique. This improvement leads to finer F1-scores for the HMM. The result implied the effectiveness of FS-LDA not only for e-commerce product classification but also for spam detection. The model accuracy was superior compared to the LDA result obtained by Nagwani and Shara [42]. The proposed model outperformed the Naïve Bayes model.

However, when the proposed model is compared with J48 and multi-layer perceptron classifiers, it seems not to be better, as shown in Renuka et al. [43]. Although, there is a slight difference between the accuracy of these models and the proposed model. The HMM model is a reliable and good classifier for classifying text datasets, especially when applying the FS-LDA technique. An HMM model itself may need some modification to achieve better performance. Yet, this feature substitution technique using the LDA model proposed in this study is relatively helpful, simple, and easy to implement. Hence, it is beneficial for commercial uses related to text classification.

VI. Conclusion

This study introduces FS-LDA, a novel technique integrating LDA into the preprocessing phase of text data classification. The results highlight the effectiveness of FS-LDA when applied with HMMs, demonstrating superior performance compared to using feature selection alone. By substituting non-overlapping words in high-probability topic groups identified by LDA, FS-LDA significantly reduces data dimensionality while enhancing the accuracy and efficiency of classification models.

The study also highlights the advantage of using a unigram model over a bigram model for feature extraction. Unigrams simplify the feature space while retaining important semantic information, making them more effective for accurate classification. This aligns with findings that simpler models often perform better in text classification by focusing on key features efficiently.

Overall, the integration of FS-LDA with HMMs and the adoption of unigram-based feature extraction represent robust strategies for improving the practical utility of text classification systems, paving the way for enhanced performance in various applications such as e-commerce product classification, spam detection, sentiment analysis, and document categorization. However, the fixed substitution percentage of FS-LDA could be tested on more datasets or through simulations to confirm its reliability. While this study focused on HMMs, trying FS-LDA with other machine learning models could offer more insights.

Acknowledgment

This research was financially supported by Universiti Teknologi MARA and the Institute of Postgraduate Studies, UiTM. It forms part of a study under the Grant Scheme (FRGS/1/2018/STG06/UITM/01/1). The authors would like to express their deepest gratitude to the Department of Statistics Malaysia for their knowledge and data support.

References

[1] D. D. Le Nguyen, Y. C. Huang, and Y. C. Chang, “Discriminative features fusion with bert for social sentiment analysis,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12144 LNAI, pp. 30–35, 2020, doi: 10.1007/978-3-030-55789-8_3.

[2] B. S. Kumar and V. Ravi, “LDA based feature selection for document clustering,” ACM Int. Conf. Proceeding Ser., pp. 125–130, 2017, doi: 10.1145/3140107. 3140129.

[3] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep Learning Based Text Classification: A Comprehensive Review,” arXiv, vol. 1, no. 1, pp. 1–43, 2020, [Online]. Available: http://arxiv.org/abs/2004.03705

[4] M. Gupta, R. Kumar, C. Ved, and S. Taneja, “Hybrid deep learning approach for product categorization in e-commerce,” AIP Conf. Proc., vol. 3072, no. 1, 2024, doi: 10.1063/5.0198666.

[5] D. Pakpahan, V. Siallagan, and S. Siregar, “Classification of E-Commerce Product Descriptions with The Tf-Idf and Svm Methods,” Sinkron, vol. 8, no. 4, pp. 2130–2137, 2023, doi: 10.33395/sinkron.v8i4 .12779.

[6] M. T. Alrefaie, N. E. Morsy, and N. Samir, “Exploring Tokenization Strategies and Vocabulary Sizes for Enhanced Arabic Language Models,” Mar. 2024, Accessed: May 02, 2024. [Online]. Available: http://arxiv.org/abs/2403.11130

[7] M. Adnan, M. O. Imam, M. F. Javed, and I. Murtza, “Improving spam email classification accuracy using ensemble techniques: a stacking approach,” Int. J. Inf. Secur., vol. 23, no. 1, pp. 505–517, 2024, doi: 10.1007/s10207-023-00756-1.

[8] J. Singh, D. Pandey, and A. K. Singh, “Event detection from real-time twitter streaming data using community detection algorithm,” Multimed. Tools Appl., vol. 83, no. 8, 2024, doi: 10.1007/s11042-023-16263-3.

[9] A. M. Ali et al., “Explainable Machine Learning Approach for Hepatitis C Diagnosis Using SFS Feature Selection,” Machines, vol. 11, no. 3, 2023, doi: 10.3390/machines11030391.

[10] N. M. N. Mathivanan, N. A. M. Ghani, and R. M. Janor, “Improving Classification Accuracy Using Clustering Technique,” Bull. Electr. Eng. Informatics, vol. 7, no. 3, pp. 465–470, 2018, doi: 10.11591/eei.v7i3.1272.

[11] N. M. N. Mathivanan, N. A. M. Ghani, and R. M. Janor, “Performance analysis of supervised learning models for product title classification,” IAES Int. J. Artif. Intell., vol. 8, no. 3, pp. 299–306, 2019, doi: 10.11591/ijai.v8.i3.pp299-306.

[12] H. Liu and H. Motoda, Computational Methods of Feature Selection, vol. 198, no. 1. 2007. doi: 10.1201/9781584888796.

[13] N. M. N. Mathivanan, N. A. M. Ghani, and R. M. Janor, “A comparative study on dimensionality reduction between principal component analysis and k-means clustering,” Indones. J. Electr. Eng. Comput. Sci., vol. 16, no. 2, pp. 752–758, 2019, doi: 10.11591/ijeecs. v16.i2.pp752-758.

[14] J. Meng, H. Lin, and Y. Yu, “A two-stage feature selection method for text categorization,” Comput. Math. with Appl., vol. 62, no. 7, pp. 2793–2800, 2011, doi: 10.1016/j.camwa.2011.07.045.

[15] D. Jain and V. Singh, “An Efficient Hybrid Feature Selection model for Dimensionality Reduction,” Procedia Comput. Sci., vol. 132, no. Iccids, pp. 333–341, 2018, doi: 10.1016/j.procs.2018.05.188.

[16] M. Shao and L. Qin, “Text Similarity Computing Based on LDA Topic Model and Word Co-occurrence,” no. Sekeie, pp. 199–203, 2014, doi: 10.2991/sekeie-14.2014.47.

[17] N. Omar and Q. Al-Tashi, “Arabic nested noun compound extraction based on linguistic features and statistical measures,” GEMA Online J. Lang. Stud., vol. 18, no. 2, 2018, doi: 10.17576/gema-2018-1802-07.

[18] M. S. Yang and K. P. Sinaga, “Collaborative feature-weighted multi-view fuzzy c-means clustering,” Pattern Recognit., vol. 119, 2021, doi: 10.1016/j.patcog.2021.108064.

[19] R. Al-Wajih, S. J. Abdulkadir, N. Aziz, Q. Al-Tashi, and N. Talpur, “Hybrid binary grey Wolf with Harris hawks optimizer for feature selection,” IEEE Access, vol. 9, 2021, doi:

10.1109/ACCESS.2021.3060096.

[20] A. Christy, A. Praveena, and J. Shabu, "A hybrid model for topic modeling using latent dirichlet allocation and feature selection method," J. Comput. Theor. Nanosci., vol. 16, no. 8, 2019, doi: 10.1166/jctn.2019.8234.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," J. Mach. Learn. Res., vol. 1, no. 4–5, 2003.

[22] J. S. Su, B. F. Zhang, and X. Xu, "Advances in machine learning based text categorization," Ruan Jian Xue Bao/Journal Softw., vol. 17, no. 9, 2006, doi: 10.1360/jos171848.

[23] Q. Wang, R. Peng, J. Wang, Y. Xie, and Y. Zhou, "Research on Text Classification Method of LDA- SVM Based on PSO optimization," in Proceedings - 2019 Chinese Automation Congress, CAC 2019, 2019. doi: 10.1109/CAC48633.2019.8996952.

[24] Y. Lu, Q. Mei, and C. X. Zhai, "Investigating task performance of probabilistic topic models: An empirical study of PLSA and LDA," Inf. Retr. Boston., vol. 14, no. 2, 2011, doi: 10.1007/s10791-010-9141-9.

[25] T. L. Griffiths and M. Steyvers, "Finding scientific topics," Proc. Natl. Acad. Sci. U. S. A., vol. 101, no. SUPPL. 1, 2004, doi: 10.1073/pnas.0307752101.

[26] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang, "Transferring topical knowledge from auxiliary long texts for short text clustering," in International Conference on Information and Knowledge Management, Proceedings, 2011. doi: 10.1145/20635 76.2063689.

[27] D. Wang, M. Thint, and A. Al-Rubaie, "Semi-supervised latent Dirichlet allocation and its application for document classification," in Proceedings of the 2012 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, WI-IAT 2012, 2012. doi: 10.1109/WI-IAT.2012.211.

[28] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM: Topic modeling over short texts," IEEE Trans. Knowl. Data Eng., vol. 26, no. 12, 2014, doi: 10.1109/TKDE.2014.2313872.

[29] X. Quan, C. Kit, Y. Ge, and S. J. Pan, "Short and sparse text topic modeling via self-aggregation," in IJCAI International Joint Conference on Artificial Intelligence, 2015.

[30] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec," Inf. Sci. (Ny)., vol. 477, 2019, doi: 10.1016/j.ins.2018.10.006.

[31] W. Zhou, H. Wang, H. Sun, and T. Sun, "A method of short text representation based on the feature probability embedded vector," Sensors (Switzerland), vol. 19, no. 17, 2019, doi: 10.3390/s19173728.

[32] N. Pröllochs and S. Feuerriegel, "Business analytics for strategic management: Identifying and assessing corporate challenges via topic modeling," Inf. Manag., vol. 57, no. 1, 2020, doi: 10.1016/j.im.2018.05.003.

[33] P. Celard, A. S. Vieira, E. L. Iglesias, and L. Borrajo, "LDA filter: A Latent Dirichlet Allocation preprocess method for Weka," PLoS One, vol. 15, no. 11 November, pp. 1–14, 2020, doi: 10.1371/journal.pone .0241701.

[34] G. R. Venkataraman et al., "FasTag: Automatic text classification of unstructured medical narratives," PLoS One, vol. 15, no. 6 June, pp. 1–18, 2020, doi: 10.1371/journal.pone.0234647.

[35] D. Freigtag and A. K. McCallum, "Information Extraction with HMMs and Shrinkage," in Proceesings of Workshop on Machine Learnig for Information Extraction, 1999.

[36] D. R. H. Miller, T. Leek, and R. M. Schwartz, "A hidden Markov model information retrieval system," Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, SIGIR 1999, pp. 214–221, 1999, doi: 10.1145/312624.312680.

[37] N. M. N. Mathivanan, N. A. M. Ghani, and R. M. Janor, "Text classification of E-commerce product via Hidden Markov model," Adv. Technol. Ind. through Intell. Softw. Methodol. Tools Tech. Proc. 18th SoMeT_19, vol. 318, pp. 310–318, 2019, doi: 10.3233/FAIA190 058.

[38] M. S. Khorsheed, "Diacritizing Arabic text using a single hidden markov model," IEEE Access, vol. 6, pp. 36522–36529, 2018, doi: 10.1109/ACCESS.2018.285 2619.

[39] W. De Gruyter, Corpus Linguistics: An International Handbook, Volume 2, vol. 2, no. 1. 2008.

[40] P. S. Florence and G. K. Zipf, "Human Behaviour and the Principle of Least Effort.," Econ. J., vol. 60, no. 240, 1950, doi: 10.2307/2226729.

[41] M. Zrigui, R. Ayadi, M. Mars, and M. Maraoui, "Based on Latent Dirichlet Allocation," J. Comput. Inf. Syst., vol. 20, no. 2, pp. 125–140, 2012.

[42] N. K. Nagwani and A. Sharaff, "SMS spam filtering and thread identification using bi-level text classification and clustering techniques," J. Inf. Sci., vol. 43, no. 1, pp. 75–87, 2017, doi: 10.1177/01655515 15616310.

[43] D. Karthika Renuka, T. Hamsapriya, M. Raja Chakkaravarthi, and P. Lakshmi Surya, "Spam classification based on supervised learning using machine learning techniques," in Proceedings of 2011 International Conference on Process Automation, Control and Computing, PACC 2011, 2011. doi: 10.1109/PACC.2011.5979035.