



An Unsupervised Machine Learning Model for Identifying Suicidal Posts Using Latent Dirichlet Allocation and Text Feature Extraction

Natthaphong Suthamno* and Jessada Tanthanuch*

Received: May 7, 2025
Revised: August 25, 2025
Accepted: August 28, 2025

* Corresponding Author: Jessada Tanthanuch, E-mail: jessada@g.sut.ac.th

DOI: 10.14416/j.it.2026.v2.005

Abstract

This study aims to enhance the performance of unsupervised machine learning models in identifying suicidal behavior from English-language Reddit posts. Using Python, a model was developed and trained on 232,074 posts with Latent Dirichlet Allocation (LDA) for topic modeling. Text feature extraction techniques incorporating unigram to 4-gram models were applied prior to model input. Four experimental setups tested different feature word sizes: the top 50,000, 100,000, 150,000 most frequent words, and all available features. Results showed that using the top 50,000 features yielded the highest accuracy (85.45%) but the lowest recall (95.55%), while using all features achieved lower accuracy (79.54%) but higher recall (97.70%). The findings suggest that reducing feature quantity enhances model accuracy and focus on suicide-related content, but may compromise recall due to loss of contextual information. Overall, the study concludes that LDA-based models can perform effectively without relying on the full vocabulary of large datasets, though a trade-off between accuracy and recall must be considered.

Keywords: Natural Language Processing, Latent Dirichlet Allocation, Text Feature Extraction, N-gram Model.

1. Introduction

Depression is a prevalent mental disorder characterized by persistent feelings of sadness, a diminished interest or pleasure in activities for long duration of these symptoms. It is estimated that around 3.8% of the global population, including 5% of adults and 5.7% of adults aged 60 and above,

experience depression. Globally, approximately 280 million individuals are affected by depression. Tragically, over 700,000 people lose their lives to suicide each year, making it the fourth leading cause of death among individuals aged 15 to 29 [1]. Diagnosis of this disorder can be based on symptom inquiries by specialized physicians or by observing certain behaviors, such as written records in documents or posts in online social media. If suicidal messages can be detected promptly, patients will be able to receive timely treatment. However, since such messages cover a wide range of topics and exist in large volumes, relying on humans to detect suicidal content can be time-consuming.

With current technology, Natural Language Processing (NLP), a branch of Artificial Intelligence (AI) that focuses on the interaction between computers and humans through natural language, has played a significant role in managing textual data. Its primary objective is to develop algorithms and computational models that enable computers to comprehend, interpret, and generate human language effectively [2]. One of the fundamental aspects of NLP is semantic analysis, which extends beyond mere word-level comprehension to grasp the underlying meaning conveyed by language. One significant application of semantic analysis is the extraction of sentiment from text on social media platforms. Semantic analysis facilitates the expression of thoughts, emotions, daily experiences, and the discussion of issues and challenges on these platforms. A notable challenge within this domain is the identification of suicidal ideation, mortality, and instances of self-harm based on text posted on social media. This is because dealing with the vast amount of text data in the annotation process can be exceedingly time-consuming and requiring significant human

* School of Mathematical Sciences and Geoinformatics, Institute of Science, Suranaree University of Technology.

effort to manually label and categorize data. The timely identification and resolution of such issues can be immensely beneficial in assisting individuals experiencing suicidal thoughts.

A sample of using semantic analysis to assist in analyzing situations related to suicide includes as the follows. In 2015, Burnap et al. combined baseline classifiers from Support Vector Machine (SVM), Decision Tree (DT), and Naïve Bayes (NB) with ensemble techniques, such as the Rotation Forest Algorithm and Maximum Probability voting classification decision methods, to classify English language texts related to suicide from Twitter [3]. They used lexical, structural, emotive, and psychological features extracted from Twitter posts as input for the models. Then, in 2018, Astoveza et al. translated Twitter posts into Filipino language and converted them into vectors using Count vectorizer, along with counting each type of emoji [4]. They selected the best features using the SelectKbest and chi2 method for further classification using Artificial Neural Network (ANN). In 2019, Jain et al. took English language texts related to suicide from Twitter, Reddit, and questionnaire, converted the texts into vectors using Term Frequency-Inverse Document Frequency (TF-IDF), and then input them into models such as Logistics Regression (LR), DT, XGBoost, and SVM [5]. In the same year, Tadesse et al. combined Long Short-Term Memory (LSTM) with Convolutional Neural Network (CNN) and named the model LSTM-CNN to classify suicide-related posts in English language from Reddit, total 7201 posts [6]. Next year, Shah et al. used English language texts from Reddit, consisting of 7098 posts, to extract features using the n-gram model, TF-IDF, and Linguistic Inquiry and Word Count (LIWC), then classified suicide-related texts using models such as NB, SVM, Random Forest (RF), and K-nearest neighbors (KNN) [7]. Two years later, Chadha and Kaushik [8] applied the attention model in CNN with layers of Bidirectional LSTM along with the use of Word embedding techniques such as Glove embedding and Random embedding for classifying suicidal ideation from a dataset of 20,000 posts [8].

Additionally, from reviewing the paper on the use of machine learning for suicide ideation detection on social media

by Alhothali and Abdulsalam [9], it was found that most models used are supervised learning algorithms, including XGBoost, J48, RF, LR, KNN, DT, SVM, NB, CNN, LSTM, and Gradient Boosting (GB). Only one unsupervised learning algorithms, such as K-means, was used, and it was applied only to Russian text. However, this work did not report the accuracy of these models but only provided the precision, recall, and F1-score, making it difficult to conclude the effectiveness of K-means in this study.

Although supervised learning models generally perform more accurate than unsupervised learning models [10], the effectiveness of supervised learning models depends on the labels in the training set. This is the important weakness of supervised learning model because there is no fixed format for splitting data for training model. If the data label is imbalanced or if the training and testing sets are not appropriately divided, it can adversely affect the model's performance.

In this paper, the focus will be on using unsupervised learning algorithm to analyze situations related to suicide. Main part emphasis on the data annotation, which is important for enhancing the accuracy and effectiveness of the model. The dataset employed comprises textual data from individuals identified as being at high risk of suicide, as well as from individuals without such risk, with each instance pre-labeled according to its respective category. To enhance the performance of the unsupervised learning algorithm for this type of dataset, the utilization of a widely adopted topic modeling technique is proposed. One of the most effective approaches involves treating the data as a statistical distribution. In this context, the Dirichlet distribution is particularly relevant, forming the basis for the Latent Dirichlet Allocation (LDA) model. The textual data will undergo feature extraction to transform it into a format suitable for topic modeling using LDA. In the research by Noah Jones et al. LDA was applied as part of a supervised machine learning pipeline following classification with Random Forest, to understand suicide risk levels within the Reddit community using the Computational Linguistics and Clinical Psychology (CLPsych) dataset [11]. The topics

were categorized into different subgroups. However, this research employs LDA directly as an unsupervised machine learning technique to explore and group topics without prior labels. At present, several variants of LDA have been developed, such as Federated LDA (F-LDA), which has been evaluated using two datasets. This approach integrates Regular LDA and Guided LDA with a dual objective, specifically designed for distributed environments [12].

The results produced by the LDA model will then be compared against the predefined labels in the dataset to assess the model's performance and reliability prior to implementing additional models for classifying individuals at risk of suicide.

2. Theoretical Background

For the second section, it will delve into providing comprehensive background information that involves this paper. This section will explain 3 related topics in detail, such as Dirichlet Distribution, Latent Dirichlet Allocation and Gibbs sampling.

2.1 Dirichlet Distribution

The Dirichlet distribution is a multivariate probability distribution that can be used to model a collection of probabilities. This distribution is often used as a prior for the parameters of a multinomial distribution in Bayesian inference. In this case, the Dirichlet distribution provides a prior over the probabilities of the different categories in a multinomial distribution [13]. This means that the Dirichlet distribution can be used to model the distribution of topics in a corpus of text, where each topic is a collection of words.

This distribution is parameterized by a vector of positive real numbers and generalizes the Beta distribution to multiple variables. The probability density function of this distribution has been shown in Equation 1,

$$f(\theta_1, \theta_2, \dots, \theta_k | \alpha_1, \alpha_2, \dots, \alpha_k) = \frac{1}{B(\alpha_1, \alpha_2, \dots, \alpha_k)} \prod_{i=1}^k \theta_i^{\alpha_i-1} \quad (1)$$

$$B(\alpha_1, \alpha_2, \dots, \alpha_k) = \frac{\prod_{i=1}^k \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^k \alpha_i)} \quad (2)$$

where $\alpha = (\alpha_1, \dots, \alpha_k)$ is the concentration parameters of Dirichlet distribution, $\theta = (\theta_1, \dots, \theta_k)$ is the parameter of multinomial distribution with $\theta_1 + \theta_2 + \dots + \theta_k = 1$ and $\theta_i \geq 0$ for all $i = 1, 2, \dots, k$, and the normalizing constant $B(\alpha)$ is the multivariable beta function as shown in Equation 2, where $\Gamma(\cdot)$ is the Gamma function.

Although Dirichlet distribution can be challenging to visualize directly due to its multidimensional nature, it is often used to represent the probabilities of a categorical variable with K possible outcomes using a K -simplex—a generalization of a triangle embedded in a higher dimension [14]. the K -simplex represents the space of all possible probability distributions over K categories. While visualizing a K -simplex for $K > 3$ can be challenging, understanding the meaning of the concentration parameters is still useful [13].

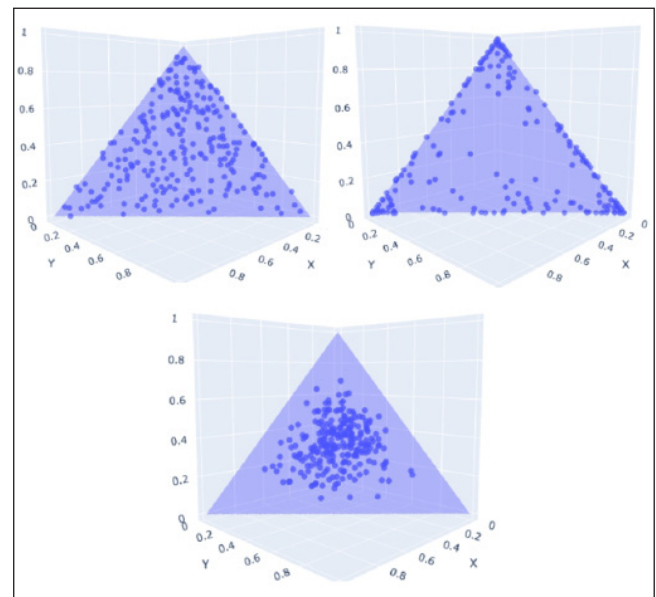


Figure 1. The Visualizations of Dirichlet Distribution on simplex in difference concentration parameters (upper left) $\alpha_i = 1$ (upper right) $\alpha_i < 1$ (lower middle) $\alpha_i > 1$ for all $i = 1, 2, \dots, k$.

The concentration parameters are hyperparameters that control the shape of the Dirichlet distribution. When $\alpha_i = 1$, the Dirichlet distribution is equivalent to a uniform distribution over the $(k - 1)$ -simplex, meaning it is uniform over all points in the hyperplane as shown in Figure 1 (left) for $k = 3$. If $\alpha_i < 1$, the distribution will be concentrated near the corners of the simplex,

as depicted in Figure 1. (middle) for $k = 3$. This implies a preference for sparse distributions, where most of the values within a single sample will be close to 0, and majority of the probability mass will be concentrated in a few values. For values of $\alpha_i > 1$, the distribution tends toward the centre of the simplex, as illustrated in Figure 1. (right) for $k = 3$. In this case, all the values within a sample will be similar to each other. As α_i increases, the distribution becomes more tightly concentrated around the centre of the simplex.

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA), first introduced by Blei, Ng and Jordan in 2003, is a statistical model that is used to discover topics in a corpus of text. This model represents topics by word probabilities, the words with highest probabilities is usually useful for specifying topic [15].

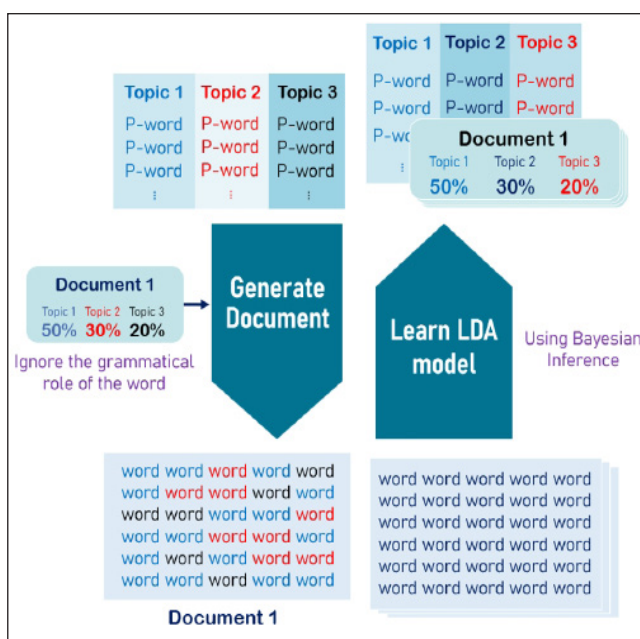


Figure 2. Brief Concept Model of LDA.

As a generative model, LDA can also generate the documents based on latent variables or topics resembling those in the corpus. To generate and interpret data efficiently, all text in dataset must according to the assumption of LDA model. It assumes that: 1) the order of words in a document can be ignored. 2) the distribution of topics in a document and words in each topic follow the Dirichlet distribution. 3) each word in a document is generated by first selecting a topic (according to the document's

topic distribution) and then drawing a word from the word distribution associated with that topic as illustrate in Figure 2.

- 4) the documents are generated independently of each other.
- 5) the word is generated independently of the other words in the document, conditioned on that topic.

In practice, it is not possible to know the document's topic distribution and the topic's words distribution. Estimating these two distributions require the Bayesian inference method to approximate the distribution characteristics of words or observed variables, as shown in Figure 2. A popular method for such estimation is Gibbs sampling, which will be discussed in section 2.3.

In general, the graphical representation used to present the allocation model of LDA is a probabilistic graphical model. This graphical model depicts the dependencies among various random variables. Each box or plate represents a repetition or iteration. The graphical representation of the LDA allocation model typically follows the structure shown in Figure 3., where D is the number of documents, N_i is the number of words in document i , α is the Dirichlet parameter of document-topic distribution, β is the Dirichlet parameter of topic-word distribution, θ_d is the parameter of topic distribution in document d , φ_k is the parameter of word distribution in topic k , $Z_{n,d}$ is the word-topic assignment at word n in document d , and $W_{n,d}$ is the observed word.

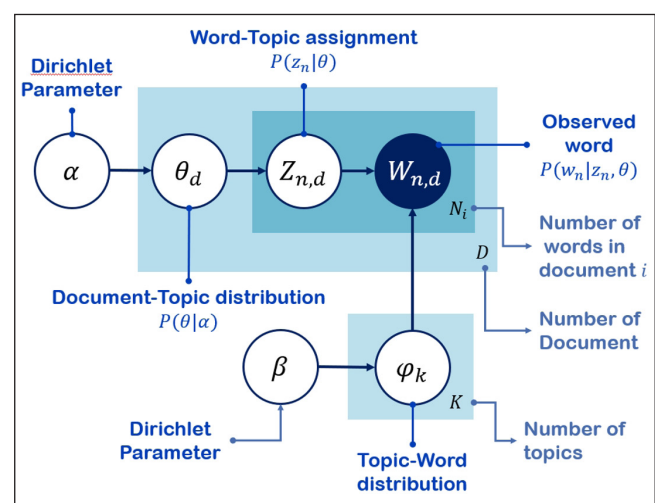


Figure 3. Probabilistic Graphical Model of LDA.

LDA finds application in various text analysis scenarios, including document classification, document clustering, text summarization, and question answering. Key advantages of LDA include its effectiveness in text analysis, its relative simplicity in understanding and usage, and the availability of software packages facilitating model training.

2.3 Gibbs Sampling

Gibbs sampling is an iterative algorithm within the Markov Chain Monte Carlo (MCMC) framework used to sample from a probability distribution. The MCMC algorithm aims to construct a Markov chain that has target posterior distribution as its stationary distribution [16]. This sampling is based on sampling from joint conditional distributions of the desired posterior $P(x_1, x_2, \dots, x_n | y)$ where $y = (y_1, y_2, \dots, y_n)$ is observed variable.

In practice, sampling from a high-dimensional joint conditional distribution can be challenging. Therefore, it is often more feasible to sample from a one-dimensional joint conditional distribution $P(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, y)$. A Gibbs sampling would perform according to Algorithm 1. After a burn-in period, samples $x^{(k)}$ are effectively samples from desired posterior distribution.

In LDA, Gibbs sampling serves as a prominent algorithmic approach for inferring the latent thematic structure within a corpus of documents. This iterative procedure refines our understanding of latent topics by following a formalized sequence of steps:

```

Randomly initial  $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$ 
for  $k = 1, 2, 3, \dots$  do
    for  $i = 1, 2, 3, \dots, n$  do
         $x_i^{(k)} \sim P(x_i | x_1^{(k)}, \dots, x_{i-1}^{(k)}, x_{i+1}^{(k-1)}, \dots, x_n^{(k-1)}, y)$ 
     $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$ 

```

Algorithm 1 Gibbs Sampling.

- (1) **Temporary removal of word assignments:** Initially, a word is selected from a document, and its current topic assignment is temporarily disregarded.
- (2) **Reassignment of topics:** Through a probabilistic assessment considering contextual information such as the surrounding words, the document's overall topic, and the general topic

distribution across the corpus, the likelihood of the word pertaining to each latent topic is computed. Subsequently, a new topic assignment is randomly assigned a new topic according to these probabilities.

- (3) **Iteration:** Steps 1 and 2 are systematically applied to all words across all documents within the corpus, thereby iteratively refining the assignments of topics to words.

Over many iterations, the topic assignments for individual words tend to converge, revealing the underlying thematic structure within the documents. The actual algorithm involves the computation of two conditional probabilities: Document-topic distribution, which assesses the likelihood of a given topic appearing in a document based on prior assignments, and Topic-Word distribution, which evaluates the probability of a word occurring given a particular topic. By iteratively updating topic assignments through Gibbs sampling, LDA gradually discovers the underlying topics that offer the most coherent explanation for the observed word occurrences within each document.

3. Research Methodology

The dataset used in this study is "Suicide and Depression Detection (Version 14)", which was downloaded from Kaggle. This dataset was updated by Nikhileswar Komati in 2021 and is licensed under CC BY-SA 4.0. It was collected from posts on the Reddit platform and consists of 232,074 rows and a label column. The dataset, categorized under the "SuicideWatch" and "r/teenagers" subreddits, was compiled from December 16, 2008 (creation) to January 2, 2021. All posts collected from the "SuicideWatch" are labeled as suicide, while posts collected from the "r/teenagers" subreddit are labeled as non-suicide. Samples from the dataset are shown in Figure 4.

	text	class
0	Ex Wife Threatening SuicideRecently I left my ...	suicide
1	Am I weird I don't get affected by compliments...	non-suicide
2	Finally 2020 is almost over... So I can never ...	non-suicide
3	i need helpjust help me im crying so hard	suicide
4	I'm so lostHello, my name is Adam (16) and I've...	suicide

Figure 4. Samples from the "Suicide and Depression Detection (Version 14)" dataset.

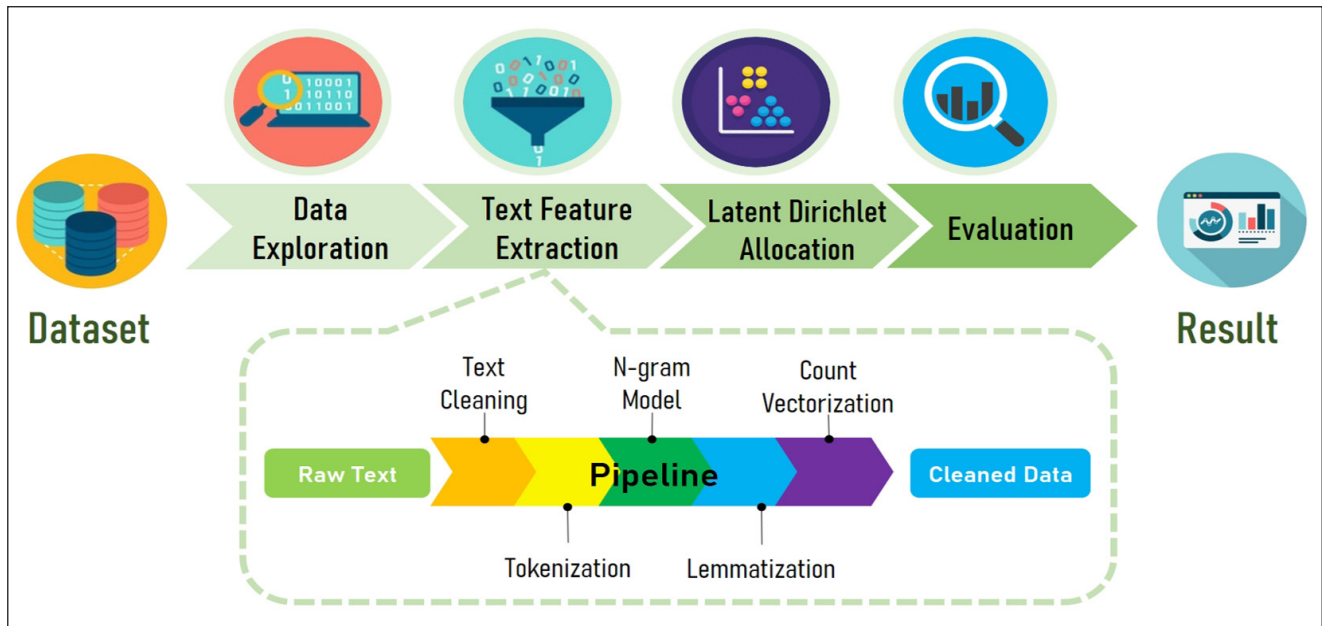


Figure 5. Workflow of this research.

In this study, we developed code in Python version 3.9 to implement the proposed methodology and compiled it on a laptop with an Intel Core i7 processor and 32GB of RAM. The workflow of this research was divided into 4 main steps. The simplified form of proposed methodology as illustrated in Figure 5.

3.1 Data Exploration

The first step of the workflow is to verify the dataset before inputting it to the next step. In this step, missing values are removed and the two classes in the dataset are balanced. After checking for missing values and class imbalance, it was found that the dataset contains no missing values and the amount of data in the suicide and non-suicide classes are exactly equal (116,037).

```

import pandas as pd
import numpy as np
import re
from collections import Counter
from tqdm import tqdm
import nltk
from nltk.corpus import stopwords
import spacy
import gensim
from gensim.utils import simple_preprocess
from gensim.models.phrases import ENGLISH_CONNECTOR_WORDS
from sklearn.feature_extraction.text import CountVectorizer

```

Figure 6. Python libraries in text feature extraction.

3.2 Text Feature Extraction

For text feature extraction in English, it begins with importing the relevant Python libraries as shown in Figure 6. [10]. This process is divided into 5 basic steps as illustrated in Figure 5. for converting text data into the format that can be inputted to the LDA model, which are outlined as follows:

- (1) **Text Cleaning:** This step involves converting uppercase letters to lowercase and removing non-meaningful text from the data.
- (2) **Tokenization:** Tokenization refers to the process of dividing the text into individual words or tokens.
- (3) **N-gram Models:** N-gram models are language probability models that capture the sequence of N consecutive words in the text [10]. This model helps in the analysis to identify phrases, as general word tokenization only splits into individual words or unigrams. In some cases, this may change the meaning of the words, as demonstrate in Figure 7. In this study, an n-gram model was constructed by considering n ranging from 1 to 4 in order to obtain a diverse set of n-grams.

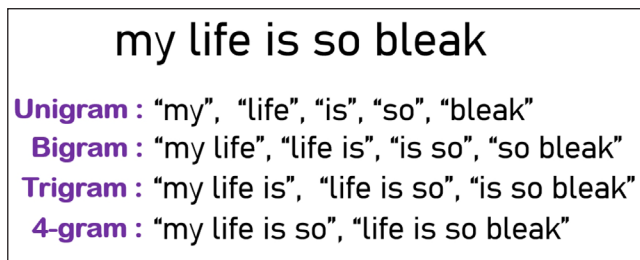


Figure 7. The demonstration of n-gram model.

- (4) **Lemmatization:** Lemmatization groups words derived from the same root into a common form to reduce the number of word variations.
- (5) **Word Vectorization:** Word vectorization is the process of converting textual data into vector representations, enabling mathematical operations and analysis on the data [17]. In this study, the vectorizer that used is called count vectorizer.

After these 5 steps, the experiment was divided into 4 sets such as using the top 50,000 most frequently words, using the top 100,000 most frequently words, using the top 150,000 most frequently words and using all words (194,916 words).

3.3 Hyperparameter setting in LDA

In this step, each cleaned data from the previous step had been inputted to LDA model and set hyperparameters each experimental sets in the same way. We assumed that this dataset must contain topics about suicide and another topic unrelated to suicide. Moreover, each document must have a dominant topic and the words in each topic must also be clear. This means that the proportion of topics in each document and the proportion of words in each topic must be differ significantly. Therefore, the Dirichlet parameter must be less than 1 in each component to ensure that the sampled multinomial parameters are distinct.

Based on the assumptions for hyperparameters settings, the number of topics is set to be 2 and each component values of Dirichlet parameters (α and β) is equal to 0.5, which is derived from 1 divided by the number of topics as shown in Figure 8.

3.4 Evaluation

After the data is fed into the model and successfully trained, document-topic and topic-word distributions are

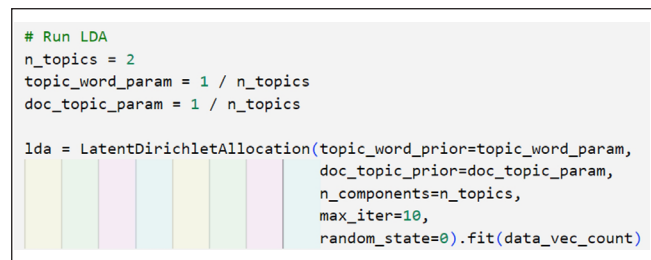


Figure 8. Coding block for hyperparameter setting in LDA.

used to categorize the types of text. The process begins with examining the topic-word distributions in each topic by investigating the first 50 words and the first 10 n-gram ($n \geq 2$) words most probability from each topic. This examination aims to identify which topics are related to suicidal tendencies and which are not.

Then, document-topic distribution will be used for labeling the document in the data. This step will assign a category to each document based on the topic that has the highest probability.

For suicide classification models, misclassifying suicide as non-suicide, as known as Type II error, is a critical concern. If this error occurs, some patients may not receive the treatment in time, which could result in death. In this case, focusing solely on accuracy to assess the performance of LDA models may not be sufficient. So, the evaluation of the model must consider recall, as it indicates the number of Type II error. The final step of the workflow compares the label from LDA model with the original label in dataset and evaluates model by considering not only accuracy and recall in each experimental set, but also computational time to assess the performance from reducing the number of words.

4. Experimental Results

The results of 4 sets will be shown in this section. In the beginning, topic-word distributions of each set were analyzed before using document-topic distribution to label all documents or texts as follow in the previous section.

4.1 Using the most 50,000 frequently words

From Table 1, the first topic has 9 suicidal words that have represented in bold words, while other topic does not any words that involve to suicide. This means the second topic must be non-suicide and the first must be suicide.



Labels obtained from analyzing Table 1 are compared with labels in the datasets. The accuracy of the first experimental set was 85.45%, with a recall value of 95.55% and a computational time of 8 minutes.

4.2 Using the most 100,000 frequently words

In Table 2, the second topic has 8 words that related to

suicide, while others topic does not. This means the first topic must be non-suicide and the second must be suicide.

After comparing between label from Table 2 and label in datasets, the accuracy and recall of the second experimental sets was 82.49% and 97.31%, respectively. Additionally, this model requires 9 minutes and 30 seconds for processing.

Table 1. The result of Experiment 1 (Using the top 50,000 most frequently words).

	Topic	
	1	2
First 50 words	feel, want, go, get, know, life, think, time, even, year, make, try, people, thing, really, friend, never, filler, live, much, well, help, day, tell, kill , end, take, die , bad , good, work, way, love, see, fuck, anymore, family, say, care, always, talk, keep, leave, still, find, long, need, hate, suicide , job	get, say, go, make, girl, guy, people, friend, day, know, school, talk, post, good, see, ask, think, tell, look, really, time, start, want, call, need, also, fuck, play, come, use, thing, give, class, first, take, help, find, today, even, reddit, year, like, new, game, still, stuff, shit, watch, back, mom
First 10 n-gram ($n \geq 2$) words	panic attack , mental illness , mental health, loved one, multiple time, suicidal ideation , commit suicide , sleeping pill, self harm , story short	ur mom ur mom, cecil cecil cecil cecil, th grade, sus sus sus sus, video game, social media, cheese cheese cheese cheese, gtpoplt gtpoplt gtpoplt gtpoplt, brazil come, discord server

Note: Words in **bold** in Tables 1 through 4 represent suicidal terms.

Table 2. The result of Experiment 2 (Using the top 100,000 most frequently words).

	Topic	
	1	2
First 50 words	filler, say, get, guy, go, girl, make, people, post, day, ask, look, good, see, play, school, friend, talk, know, think, need, also, fuck, use, call, really, tell, class, game, start, give, come, time, today, reddit, first, want, teacher, watch, like, find, comment, man, stuff, new, thing, question, help, take	feel, want, go, get, know, life, think, time, even, year, make, friend, people, really, try, thing, never, tell, day, live, much, help, well, take, say, bad , kill , end, good, die , see, work, way, talk, fuck, love, anymore, always, family, care, keep, still, leave, find, need, start, back, long, school, hate
First 10 n-gram ($n \geq 2$) words	ur mom ur mom, cecil cecil cecil cecil, sus sus sus sus, video game, cheese cheese cheese cheese, th grade, social media, gtpoplt gtpoplt gtpoplt gtpoplt, brazil come, discord server	panic attack , mental illness , mental health, loved one, multiple time, th grade, video game, suicidal ideation commit suicide , story short

Table 3. The result of Experiment 3 (Using the top 150,000 most frequently words).

	Topic	
	1	2
First 50 words	filler, say, get, guy, girl, make, go, post, people, day, ask, look, good, play, see, school, talk, know, friend, think, need, fuck, also, use, game, call, class, give, really, reddit, start, come, teacher, today, tell, watch, comment, first, like, man, stuff, new, find, time, question, want, mean, boy, help, thank	feel, want, go, get, know, life, think, time, even, make, year, friend, people, really, try, thing, never, tell, day, live, much, help, well, say, take, bad , good, kill , end, see, die , work, talk, way, fuck, love, anymore, always, family, still, care, keep, leave, start, find, need, school, back, long, hate
First 10 n-gram ($n \geq 2$) words	ur mom ur mom, cecil cecil cecil cecil, sus sus sus sus, video game, cheese cheese cheese cheese, th grade, gtpoplt gtpoplt gtpoplt gtpoplt, brazil come, social media, discord server	panic attack , mental illness , mental health, th grade, loved one, video game, multiple time, suicidal ideation , story short, commit suicide

Table 4. The result of Experiment 4 (Using all words).

	Topic	
	1	2
First 50 words	filler, say, get, guy, girl, make, post, people, go, ask, day, look, play, see, good, talk, game, school, use, need, also, fuck, know, think, friend, class, call, reddit, give, comment, today, teacher, watch, come, first, like, man, start, question, really, new, stuff, tell, find, boy, thank, mean, name, send, crush	want, feel, go, get, know, life, think, time, even, make, year, friend, people, really, try, thing, never, tell, day, much, live, help, say, well, take, good, bad, kill , end, see, talk, work, die , way, fuck, love, anymore, always, still, family, care, keep, leave, school, start, find, need, back, long, last
First 10 <i>n</i> -gram ($n \geq 2$) words	ur mom ur mom, cecil cecil cecil cecil, sus sus sus sus, cheese cheese cheese cheese, video game, gtpoplt gtpoplt gtpoplt gtpoplt, brazil come, discord server, social media, th grade	panic attack, mental illness , th grade, mental health, video game, loved one, multiple time, story short, suicidal ideation, commit suicide

4.3 Using the most 150,000 frequently words

From Table 3, these 2 topics have a resemblance to the previous experimental set. Therefore, the first topic will be non-suicide topic and the second will be suicide topic. However, the accuracy, recall and computational time of the third experimental sets do not match those of the second test set, which was 81.01%, 97.53% and 10 minutes and 30 seconds, respectively.

4.4 Using all words

From Table 4, the second topic has 6 suicidal words, while other topic does not any words that related to suicide. This means the second topic must be suicide and the first must be non-suicide. The accuracy of the fourth experimental sets was 79.54%, with a recall value of 97.70% and the computing time was 11 minutes and 2 seconds.

5. Discussion

The results from each experiment are summarized in Table 5. The findings show that reducing the number of words input into the LDA model increases the proportion of suicide-related terms in both the top 50 highest-probability unigrams and the top 10 highest-probability *n*-grams. Additionally, text classification accuracy improves. This suggests that limiting the number of terms before inputting helps eliminate irrelevant words, thereby enhancing model efficiency. In the case of LDA, certain feature words may lie ambiguously between two states. Consequently, reducing the number of terms can shift these ambiguous instances toward the correct state, thereby improving overall accuracy. However, this process

slightly decreases the true positive rate, resulting in a marginal reduction in recall. In comparison, the gain in accuracy is considerably greater than the small loss in recall.

Moreover, reducing the number of featured words decreases computational time by lowering the complexity of the model. However, the recall value decreases as the word count is reduced, indicating an increase in misclassifications of suicide-related texts as non-suicide. This may result from the loss of crucial contextual information due to the word limitation.

Table 5. Accuracy and Recall of each experimental set.

Featured Words	Accuracy	Recall	Time Taken
50,000 words	85.45 %	95.55 %	8 minutes
100,000 words	82.49 %	97.31 %	9 minutes 30 seconds
150,000 words	81.01 %	97.53 %	10 minutes 30 seconds
All words	79.54 %	97.70 %	11 minutes 2 seconds

Focus on the most frequently words especially related to suicide, each experiments have a different number of those words. In unigram words, it is difficult to label which topic is related to suicide because each experiment has a few suicidal words, such as die, kill, bad, etc. However, in bigram and trigram words, it is clearly to identify suicide topic by observing the suicidal words, such as panic attack, mental illness, suicidal ideation, commit suicide, etc.

From a total of 194,916 features, the reduction in the number of features for model construction demonstrates varying results in terms of accuracy, recall, and model training time.

However, the LDA model has limitations in that the features must adhere to the condition that the topics are independent and follow a Dirichlet distribution, which is difficult to verify for the features used in this study. Nevertheless, the results presented in Table 5 show that all models achieved high accuracy, despite being unsupervised machine learning models.

6. Conclusion

The goal of this study is to improve the efficiency of unsupervised machine learning models applied to text data from users showing signs of suicidal tendencies, based on posts made by English-language users on the Reddit platform. The research focuses on simplifying the input data while maintaining the model's performance. After conducting a literature review, it was found that the LDA model, which is the topic model applied in various works, can efficiently summarize topics from text data, reducing the time and resources needed for text identification.

Based on the results in this research, we conclude that utilizing the top 50,000 most frequently occurring words before inputting them into the LDA model yields the highest accuracy, achieving an accuracy of 85.45% and a recall of 95.55%. Additionally, using the top 50,000 most frequently featured words also results in the least computational time, taking only 8 minutes. On the other hands, the experimental set that analyses all words in the dataset achieves the highest recall, with an accuracy of 79.54% and a recall of 97.70%. This conclusion infers that reducing the number of top most frequently used features before inputting them into the LDA model improves classification performance. However, using fewer features may cause the model to overlook a significant number of actual suicidal cases, although this error remains rare. This shows that LDA model does not necessarily need to use all words in large dataset and can achieve high performance.

Although the results of this work may be difficult to compare directly with other studies due to differences in datasets, such as platform and number of data, a review of the most

similar studies, which also used data from the SuicideWatch subreddit, identified two relevant papers. The first, by Shah et al., used a smaller dataset of 7,098 posts [7]. Their best-performing technique, NB, achieved the highest accuracy and recall at 73.6% and 89.7%, respectively, clearly indicating that the performance of the LDA model is better. Another study by Chadha and Kaushik used a dataset of 20,000 posts [8]. Their best-performing model used a CNN combined with a bidirectional LSTM and attention model with Glove embedding, achieving an accuracy and recall of 88.48% and 94.57%. When comparing this with the highest-performing LDA model, which achieved an accuracy of 85.45%, we see only a slight difference, as unsupervised learning generally has lower accuracy than supervised learning [10]. On the other hand, the recall of LDA, at 95.55%, was still higher than the models mentioned above, even though it had the lowest recall among other models in this work.

The proposed model can also be extended to improve the performance of other supervised classification models for training and effectively classifying unlabeled data, particularly when working with datasets that lack labels [18]. It is additionally compatible with multiple languages, though text feature extraction methods should be tailored to the unique characteristics of each language. Therefore, the LDA model is another tool that assists in data annotation and saves time in manual annotation efficiently. However, selecting suitable hyperparameters (α and β) remains a challenge for creating model, since these hyperparameters do not have fix rules for selection to yield the most efficient model. This issue may lead to future works in this field.

7. Acknowledgement

This research paper has received courtesy and support from Development and Promotion of Science and Technology Talents Project (DPST) and the school of Mathematical Sciences and Geoinformatics, Institute of Science, Suranaree University of Technology.



8. References

- [1] World Health Organization (2023), *Depressive disorder (depression)*. Available Online at <https://www.who.int/news-room/fact-sheets/detail/depression>, accessed on 19 April 2025.
- [2] A. Clark, C. Fox, and S. Lappin. *The Handbook of Computational Linguistics and Natural Language Processing*. Hoboken, NJ, USA: John Wiley & Sons, 2013.
- [3] P. Burnap, W. Colombo, and J. Scourfield. "Machine classification and analysis of suicide-related communication on Twitter." *26th ACM Conference on Hypertext & Social Media (HT '15)*, New York, NY, USA, pp. 75-84, 2015.
- [4] G. Astoveza, R. J. P. Obias, R. J. L. Palcon, R. L. Rodriguez, B. S. Fabito, and M. V. Octaviano. "Suicidal behavior detection on Twitter using neural network." *TENCON 2018 - 2018 IEEE Region 10 Conference*, Jeju, Korea (South), pp. 0657-0662, 2018.
- [5] S. Jain, S. P. Narayan, R. K. Dewang, U. Bhartiya, N. Meena, and V. Kumar. "A Machine Learning based Depression Analysis and Suicidal Ideation Detection System using Questionnaires and Twitter." *2019 IEEE Students Conference on Engineering and Systems (SCES)*, Allahabad, India, pp. 1-6, 2019.
- [6] M. M. Tadesse, H. Lin, B. Xu, and L. Yang. "Detection of Suicide Ideation in Social Media Forums Using Deep Learning." *Algorithms*, Vol. 13, No. 1, 2019.
- [7] F. M. Shah, F. Haque, R. U. Nur, S. Al Jahan, and Z. Mamud. "A Hybridized Feature Extraction Approach To Suicidal Ideation Detection From Social Media Post." *2020 IEEE Region 10 Symposium (TENSYP)*, Dhaka, Bangladesh, pp. 985-988, 2020.
- [8] A. Chadha and B. Kaushik. "A Hybrid Deep Learning Model Using Grid Search and Cross-Validation for Effective Classification and Prediction of Suicidal Ideation from Social Network Data." *New Generation Computing*, Vol. 40, No. 4, pp. 889-914, 2022.
- [9] A. Abdulsalam and A. Alhothali. "Suicidal ideation detection on social media: A review of machine learning methods." *Social Network Analysis and Mining*, Vol. 14, No. 1, 2024.
- [10] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. Sebastopol, CA, USA: O'Reilly Media, 2009.
- [11] N. Jones, N. Jaques, P. Pataranutaporn, A. Ghandeharioun, and R. Picard. "Analysis of Online Suicide Risk with Document Embeddings and Latent Dirichlet Allocation." *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, Cambridge, MA, USA, pp. 1-5, 2019. doi: 10.1109/ACIIW.2019.8925047.
- [12] R. Kushwaha and P. Kaur. "A federated LDA-based approach for topic modelling." *Intelligent Decision Technologies*, Vol. 19, No. 4, pp. 2738-2759, 2025. doi: 10.1177/18724981251319629.
- [13] K. W. Ng, G. L. Tian, and M. L. Tang. *Dirichlet and Related Distributions: Theory, Methods and Applications*. Hoboken, NJ, USA: John Wiley & Sons, 2011.
- [14] B. A. Frigyik, A. Kapila, and M. R. Gupta. "Introduction to the Dirichlet Distribution and Related Processes." *UWEE Technical Report Series*, Vol. 1, pp. 1-27, 2010.
- [15] D. M. Blei, A. Y. Ng, and M. I. Jordan. "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, Vol. 3, pp. 993-1022, 2003.
- [16] W. M. Darling. "A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling." *49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 642-647, 2011.
- [17] M. T. Pilehvar and J. Camacho-Collados. *Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2020.
- [18] A. H. Razavi, D. Inkpen, D. Brusilovsky, and L. Bogouslavski. "General topic annotation in social networks: A Latent Dirichlet Allocation approach." *in Advances in Artificial Intelligence: 26th Canadian Conf. Artificial Intelligence (Canadian AI 2013)*, Regina, SK, Canada, pp. 293-300, 2013.