JTAM (Jurnal Teori dan Aplikasi Matematika)
http://journal.ummat.ac.id/index.php/jtam
p-ISSN 2597-7512 | e-ISSN 2614-1175
Vol. 8, No. 3, July 2024, pp. 896-909

Chi-Square Feature Selection with Pseudo-Labelling in Natural Language Processing

Sintia Afriyani[1], Sugiyarto Surono[1], Mahmud Iwan Solihin[2]
[1]Departement of Mathematics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[2]Assoc Prof at Faculty of Engineering, UCSI University, Malaysia
sintia2000015036@webmail.uad.ac.id

Article History:
Received : 31-03-2024
Revised : 01-06-2024
Accepted : 03-06-2024
Online : 19-07-2024

Keywords:
Chi-Square Feature Selection;
Natural Language Pocessing;
Pseudo-Labelling;
Semi-supervised.

ABSTRACT

This study aims to evaluate the effectiveness of the Chi-Square feature selection method in improving the classification accuracy of linear Support Vector Machine, K-Nearest Neighbors and Random Forest in natural language processing when combined with classification algorithms as well as introducing Pseudo-Labelling techniques to improve semi-supervised classification performance. This research is important in the context of NLP as accurate feature selection can significantly improve model performance by reducing data noise and focusing on the most relevant information, while Pseudo-Labelling techniques help maximise unlabelled data, which is particularly useful when labelled data is sparse. The research methodology involves collecting relevant datasets, thus applying the Chi-Square method to filter out significant features, and applying Pseudo-Labelling techniques to train semi-supervised models. In this study, the dataset used in this research is the text data of public comments related to the 2024 Presidential General Election, which is obtained from the Twitter scrapping process. The characteristics of this dataset include various comments and opinions from the public related to presidential candidates, including political views, support, and criticism of these candidates. The experimental results show a significant improvement in classification accuracy to 0.9200, with precision of 0.8893, recall of 0.9200, and F1-score of 0.8828. The integration of Pseudo-Labelling techniques prominently improves the performance of semi-supervised classification, suggesting that the combination of Chi-Square and Pseudo-Labelling methods can improve classification systems in various natural language processing applications. This opens up opportunities to develop more efficient methodologies in improving classification accuracy and effectiveness in natural language processing tasks, especially in the domains of linear Support Vector Machine, K-Nearest Neighbors and Random Forest well as semi-supervised learning.

https://doi.org/10.31764/jtam.v8i3.22751

———————— ◆ ————————

A. INTRODUCTION

The Chi-Square feature selection method combined with Pseudo-Labelling offers an effective solution for tackling the challenge of extracting information from large and unstructured text data. Chi-Square is utilized to determine the statistical significance of each word towards specific categories or topics within text documents. This approach enables companies to identify the most relevant keywords for distinguishing different document classes or topics. Moreover, Pseudo-Labelling allows the use of unlabeled text data to enhance the classification model's performance thereby maximizing the extraction of value from big data (Syrotkina et al., 2020). This can significantly aid companies in optimizing operational efficiency, identifying hidden patterns, and improving their decision-making processes (Adnan

& Akbar, 2019). Online platforms like Twitter are pivotal for capturing opinions and sentiments expressed through tweets (Garg et al., 2020). Sentiment analysis, a critical component of Natural Language Processing (NLP), plays a key role in understanding and deriving meaning from textual data. However, tweets often contain irrelevant features that complicate analysis (N. K. Singh et al., 2020). Feature selection is crucial in improving sentiment analysis by identifying the most informative and relevant features (Tubishat et al., 2019). Techniques such as Chi-Square have proven effective in selecting the most significant features (Hamzah, 2021). This method ranks features based on their statistical relevance to improve classification accuracy (Alshaer et al., 2021).

In researchers A. Yang et al. (2016) proposed an improved feature selection method that combines TF-IDF with Chi-Square for Twitter sentiment analysis, improving classification accuracy when applied with Naive Bayes classifier and Support Vector Machine. They suggest combining more complex feature selection and extraction techniques for better results. A previous study by Paudel et al. (2019) entitled "A feature selection approach for Twitter sentiment analysis and text classification based on Chi-Square and Naive Bayes" achieved 80% accuracy using Chi-Square feature selection and Naive Bayes classification. Paudel et al. recommend further research to refine the training data and better group the data, noting that their study used a 50:50 data split. However, the main challenge lies in obtaining labeled data for training classification models, which is costly and difficult to acquire (Shan Lee et al., 2019). Semi-supervised learning addresses this issue by leveraging unlabeled data and Pseudo-Labelling methods, iteratively strengthening classification models and enhancing sentiment analysis performance (W. Yang et al., 2023).

Future research will introduce semi-supervised techniques like Pseudo-Labelling for innovation in sentiment analysis of Twitter data. This study will focus on improving feature selection algorithm performance, particularly by using larger proportions of training and testing data, such as splits of 70:30, 80:20, and 90:10, as well as increasing the amount of training data. The goal is to enhance the accuracy of sentiment analysis models by applying classification methods such as linear Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Random Forest. The Chi-Square feature selection technique will be used to identify the most informative features. Unlike previous research, this study will expand on semi-supervised approaches with Pseudo-Labelling to maximize the use of unlabeled data, addressing the limitations of expensive and hard-to-obtain labeled data. The expected outcome is to provide an effective solution for improving sentiment analysis with higher accuracy, offering deep insights into opinions and sentiments expressed in tweets, and enhancing decision-making capabilities based on large, unstructured text data.

B. METHODS

This research uses NLP for text analysis. The Chi-Square method is used for feature selection, and Pseudo-Labelling is applied to semi-supervised data. Linear SVM, KNN, and Random Forest were used for classification. The research objective is to evaluate the effectiveness of Pseudo-Labelling in improving classification performance on semi-supervised datasets in the context of NLP.

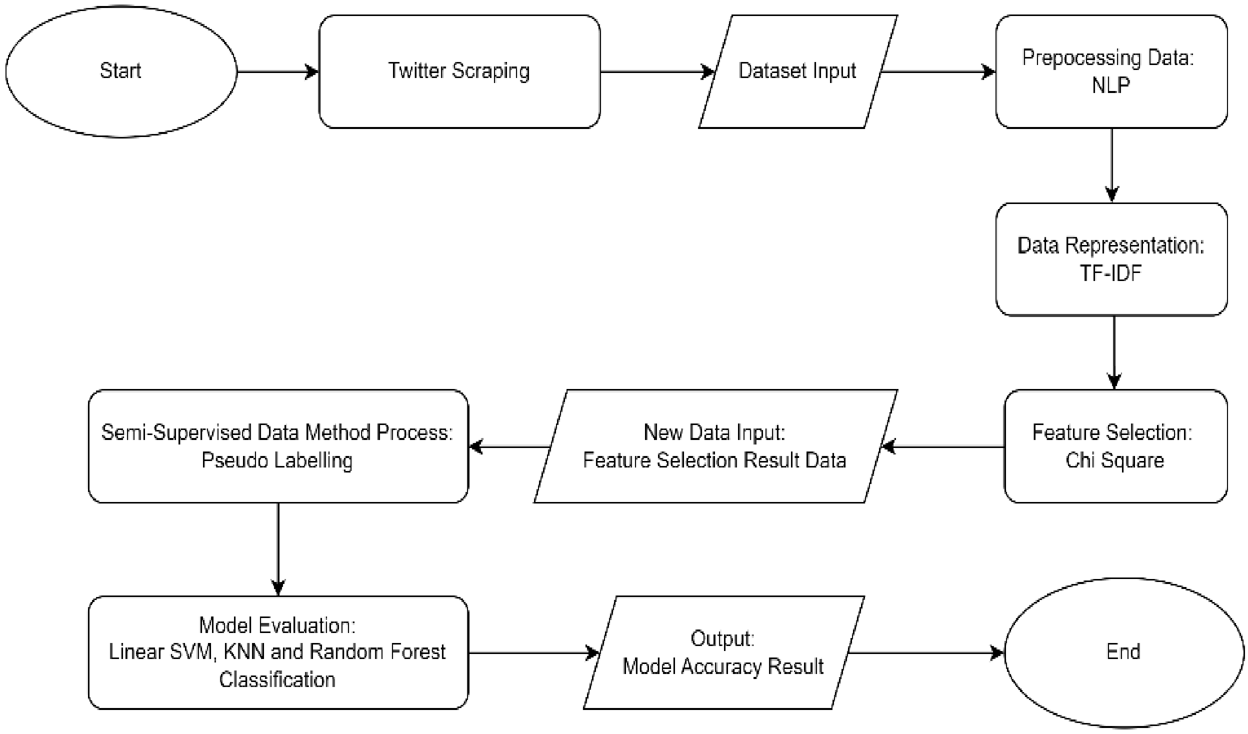
Start
Twitter Scraping
Dataset Input
Prepocessing Data: NLP
Data Representation: TF-IDF
Feature Selection: Chi Square
New Data Input: Feature Selection Result Data
Semi-Supervised Data Method Process: Pseudo Labelling
Model Evaluation: Linear SVM, KNN and Random Forest Classification
Output: Model Accuracy Result
End

Figure 1. Research Flowchart

Figure 1 shows the flowchart of this research involving text data collection by Twitter scraping, preprocessing using NLP, feature selection by the Chi-Square method, and the application of Pseudo-Labelling techniques on semi-supervised data before building and evaluating classification models such as linear SVM, KNN, and Random Forest to gain insight into their use in the realm of Natural Language Processing (NLP), the focus remains on refining sentiment analysis models through the utilization of different classification techniques such as linear SVM, KNN, and Random Forest.

1. Scrapping Twitter

Scraping data from Twitter using Python can be done using libraries such as Tweepy, which allows access to the Twitter API (Herlawati et al., 2020). The process of scraping data from Twitter using Python can be mathematically represented by the equation $D = P(T, L, A)$. In this equation, D represents the data obtained from Twitter. The function P stands for the scraping process itself. The variable T denotes the set of tweets to be retrieved, while L refers to the library used for scraping data, which in this case is Tweepy. Lastly, A signifies access to the Twitter API. This concise formula encapsulates the steps involved in accessing and extracting tweet data using the specified tools and methods (Al Walid et al., 2019).

2. Natural Language Processing

Natural Language Processing encompasses a set of theoretically grounded computational techniques methods designed to analyze and represent text in a manner similar to human speech comprehension. These techniques aim to achieve language processing capabilities akin to those of humans across a range of tasks or applications (Ferrario & Naegelin, 2020). Text preprocessing is essential to improve data quality and remove noise before data clustering. This process starts with text cleaning, which involves removing unwanted information such as numbers, punctuation marks, and symbols, converting text to lowercase, and removing links (Sakthi Vel, 2021). After cleaning, tokenisation divides the text into meaningful units, known as tokens, which can be words or phrases, so that the text can be managed for analysis (Garcia-Teruel & Simón-Moreno, 2021). Character-level tokenisation adds flexibility in handling uncommon words or specialised terms (Kudo & Richardson, 2018). Furthermore, Stemming Process can reduce words to their basic form by removing affixes, normalising word variants, improving the quality of analysis (Jabbar et al., 2020). Finally, stop word removal is useful in removing common words that do not contribute meaningful information, streamlining data processing (Sarica & Luo, 2021). These preprocessing steps collectively improve the efficiency and accuracy of text clustering.

3. Term Frequency Inverse Document Frequency

Text representation is the process of converting text data into a format suitable for computer processing. There are several methods available for this purpose, including one-hot, bag-of-words (BoW), and TF-IDF coding. Each method has pros and cons. One-hot encoding represents words as binary vectors but lacks semantic information and is less efficient with high-dimensional data. Bag-of-words captures word frequency but does not consider word order and context, thereby limiting its ability to understand relationships between words. In contrast, TF-IDF evaluates the importance of each term based on its frequency within a document and across a collection of documents. This method reduces the impact of general terms and emphasizes specific terms, making it suitable for tasks such as text classification and information retrieval (Sonbol et al., 2022). The following is the formula for TF-IDF:

$$\mathrm{TF\ (t, d)\ =\ \frac{Sum\ term\ t\ in\ document\ d}{Sum\ total\ term\ in\ document\ d}} \tag{1}$$

$$\mathrm{IDF\ (T, D)\ =\ log\ \left(\frac{Sum\ document\ in\ corpus\ D}{Sum\ documen\ that\ contain\ term\ \ t}\right)} \tag{2}$$

$$\mathrm{TF - IDF\ =\ TF\ (t, d)\ \ x\ IDF\ (T, D)} \tag{3}$$

In (1), we calculate how often a word appears in a sentence by dividing the frequency of that word by the total number of words in that sentence. On the other hand, in (2), IDF measures the significance of a word in the context of a larger document or text corpus. When calculating TF, it's assumed that all words have equal importance. Therefore, the higher the IDF value of a word, the lower the importance of that word in the sentence. Consequently, (3) is the result of multiplying TF and IDF, used to evaluate the importance of a word in a sentence by considering its frequency in a larger document or text corpus.

4. Chi-Square Feature Selection

Chi-Square feature selection is a method employed in data analysis to determine the correlation between features is independent variables and targets dependent variables within a dataset. Its objective is to identify the most pertinent or informative features for predicting the target variable. (Paudel et al., 2019). The formula used is as follows:

$$\chi^2(\mathrm{t_k}, \mathrm{c_i}) = \frac{\mathrm{N(AD - CB)^2}}{\mathrm{(A + C)(B + D)(A + B)(C + D)}} \tag{4}$$

Where:

N= total data in the corpus

A = total data in the class c_i *that contain therm* t_k

B = total data containing therms t_k that are not class c_i

C = total data in the class c_i that does not contain therm t_k

D = total data that is not classified c_i does not contain t_k

Chi-Square (χ^2) feature selection in text data is a statistical technique used to assess the relationship between a term t_k and a class c_i . The Chi-Square formula used is given as (4), where N represents the total number of data instances in the corpus.

5. Pseudo Labelling

Pseudo-Labelling is a machine learning technique used to enhance model performance by leveraging unlabeled or improperly labeled data, commonly applied in semi-supervised learning scenarios where labeled data is scarce compared to unlabeled data (Asghar et al., 2020). Initially, an initial model M is trained on the labeled dataset D_1 , defined as $M = f(X_1, y_1)$, where X_1 represents the feature set of the labeled data and y_1 denotes the corresponding labels. After training, the model M predicts labels for the unlabeled data D_u, producing predictions $\widehat{y_u} = M(X_u)$, where X_u is the feature set of the unlabeled data. These predictions are then utilized as pseudo-labels for the unlabeled data, forming a new dataset $D_{Pseudo} = (X_u, \widehat{y_u})$. The falsely labeled data is combined with the original labeled data to create a unified dataset $D_{Combined} = (D_1 \cup D_{Pseudo})$. Subsequently, the model is updated or retrained using the combined dataset $D_{Combined}$, resulting in a new model $M' = f(X_1 \cup X_u, y_1 \cup \widehat{y_u})$. This iterative process can be repeated multiple times, adapting the steps according to the specific machine learning algorithm employed.

6. Classification

Classification methods for text sentiment analysis use machine learning algorithms to classify opinions or sentiments in text into relevant categories, such as positive, negative, or neutral. The steps include text preprocessing such as data cleaning and tokenization, followed by vectorization (K. N. Singh et al., 2022). Common classification algorithms are employed to understand the meaning and emotions in text to produce accurate sentiment predictions (Chen et al., 2020). The methods are as follows:

a. Support Vector Machine

The Support Vector Machine (SVM) classification technique employs a hyperplane to segregate data classes while maximizing the margin between them. In sentiment analysis, SVM endeavors to find a hyperplane that maximizes the margin between various sentiment classes by optimizing the margin using the following objective function (Mohd Nafis & Awang, 2021):

$$\min_{w,b} \frac{1}{2} w^2 + C \sum_{i=1}^{N} \max\big(0,1 - y_i(w \cdot x_i + b)\big)$$

The objective function aims to minimize the norm of the weight vector w taking into account the classification error, with the parameter C controlling the penalty for error. Selecting the parameter C appropriately is crucial to achieve optimal performance in SVM.

b. K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) method is a classification algorithm that predicts new data labels by identifying the K nearest data points to the data being predicted. This algorithm uses the Euclidean distance:

$$\text{Euclidean Distance}(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

KNN is effective for datasets with a moderate size and when interpretability of the model is important. However, it is sensitive to noise and outliers, necessitating careful selection of the k parameter to optimize performance (Deta Kirana & Al Faraby, 2021).

c. Random Forest

Random Forest is an ensemble method for classification and regression, comprising multiple decision trees that work collectively. Each tree is trained on a unique subset of the data through bootstrap sampling, where subsets D_i are created from the original dataset D with N samples. For each subset, decision trees are built by selecting the best features from a random subset of the original features $\{f_1, f_2, \dots, f_{m'}\}$, with $m' \ll m$, based on criteria like Gini impurity $Gini(A) = 1 - \sum_{i=1}^{C} p_i^2)\ orentropy(Entropy(A) = -\sum_{i=1}^{C} p_i \log(p_i)$ (Arora & Kaur, 2020). Final predictions for new data are obtained by majority voting for classification:

$$H(x) = \text{mode}\{h_1(x), h_2(x), \dots, h_T(x)\}$$

or averaging predictions for regression:

$$H(x) = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$$

Feature importance is estimated by measuring the average reduction in Gini impurity caused by feature f_j in each tree:

$$\text{Importance}(f_j) = \frac{1}{T} \sum_{t=1}^{T} \Delta\text{Gini}(f_j)$$

This method enhances accuracy and reduces overfitting by aggregating predictions from multiple trees. To better understand the implementation of selected methods, it is crucial to carefully choose parameters such as C in SVM and k in KNN as they significantly impact algorithm performance. In SVM, C controls the penalty for classification errors and should be optimized through cross-validation to find the optimal value. In KNN, k determines the number of neighbors contributing to predictions and must be balanced to manage bias and variance for accurate results (Deta Kirana & Al Faraby, 2021). Additionally, parameters like the number of trees and tree depth in Random Forest can be optimized to enhance performance and mitigate overfitting (Arora & Kaur, 2020).

7. Model Evalution

Model evaluation assesses the model's ability to correctly distinguish between different classes in the dataset. Careful analysis of the confusion matrix is essential for enhancing model performance (Krstinić et al., 2020). From the model evaluation, the accuracy value can be calculated as follows:

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1} - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Where: TP is True Positive; TN is True negative; FP is False Positive; and FN is False Negative. Model evaluation assesses for evaluating classification models in machine learning include accuracy, precision, recall, and the F1-score. Accuracy, determined by formula (5), calculates the proportion of correct predictions from the total number of predictions, accounting for TP, TN, FP, and FN. Precision, using formula (6), measures the proportion of true positive instances among all positive predictions. Recall, or sensitivity, calculated by formula (7), assesses the model's effectiveness in identifying all actual positive samples. The F1-score, given by formula (8), is the harmonic mean of precision and recall, offering a balanced metric especially valuable for datasets with uneven class distributions.

C. RESULT AND DISCUSSION

1. Text Dataset

This research successfully collected 8846 datasets from Twitter through a scraping process with Python, as shown in Table 1.

Table 1. Twitter Srapping Result Dataset

No	Text
1	Nggak ada sih pak. Maka dari itu kita piling Capres yang paling sedikit Parpolnya. Yaitu Anis Capres no 1
2	Ganjar sebagai capres memberikan harapan baru untuk pemerintahan yang lebih bersih dan efisien di tingkat nasional. Ganjar Pranowo, terbukti lebih baik
3	Dukung Ganjar, capres yang nggak cuma ngomong, tapi bener-bener bikin perubahan buat anak muda!
...	...
8876	Makin banyak yg mendukung Prabowo Gibran, yg tulus mbangun negeri. Mari para sahabat, teman2, warga pecinta nkri kita pilih Capres no urut 02. Akhirnya! Terungkap Arah

In Table 1, these datasets include a variety of information, such as tweet text and metadata. This data has the potential to be a source of information for sentiment analysis, trends, and Twitter user behavior.

2. Text Preprocessing

Text preprocessing is the process of cleaning and preparing text data for further analysis, through text cleaning, tokenization, stemming, and stop word removal. This process can improve the accuracy of text analysis and facilitate more effective data interpretation and modeling.

Table 2. Text Preprocessing Step

Step	Command	Function
Text Cleaning	re.sub(r'#', '', Text)	Eliminating hashtag characters
	re.sub(r'\n', '', Text)	Delete new line characters
	re.sub(r"\d+", " ", str(Text))	Eliminating numbers
	re.sub(r'[^a-zA-z0-9]', ' ', str(Text))	Delete characters other than alphanumeric letters

Tokenization	def tokenization(text):
	text = re.split(r'\W+', text.lower())
	return text
		Divide the text, which can be a sentence, paragraph or document, into tokens/parts.

Stemming	def stemming(text):
	return StemmerFactory().create_stemmer().stem(text)
		forms a word into its root word.

Stop Word Removal	nltk.corpus.stopwords.words('indonesian')	disposal of root words that have no meaning

Table 2 categorizes preprocessing steps into steps, commands, and functions, detailing specific actions executed during text preprocessing. It emphasizes how each step enhances text data quality by systematically removing irrelevant and common words, thereby focusing the analysis on meaningful content. Overall, Table 2 serves as a comprehensive guide for efficient text data preprocessing, improving the effectiveness of subsequent analysis tasks. Meanwhile, Table 3 displays the results of text preprocessing, showing modifications in the original text after steps like special character removal, normalization, tokenization, stemming, and stop word removal.

Table 3. Text Preprocessing Result

No	Text	Prepocessing Text
1	Nggak ada sih pak. Maka dari itu kita piling Capres yang paling sedikit Parpolnya. Yaitu Anis Capres no 1	ada maka piling capres paling sedikit parpol anis capres
2	Ganjar sebagai capres memberikan harapan baru untuk pemerintahan yang lebih bersih dan efisien di tingkat nasional. Ganjar Pranowo, terbukti lebih baik	ganjar capres beri harap pemerintah bersih efisien tingkat nasional ganjar pranowo bukti baik
3	Dukung Ganjar, capres yang nggak cuma ngomong, tapi bener-bener bikin perubahan buat anak muda!	dukung ganjar ngomong, bener bener bikin perubah anak muda
...	...	
8876	Makin banyak yg mendukung Prabowo Gibran, yg tulus mbangun negeri. Mari para sahabat, teman2, warga pecinta nkri kita pilih Capres no urut 02. Akhirnya! Terungkap Arah	banyak dukung prabowo gibran tulus bangun negeri sahabat teman warga cinta nkri kita pilih capres akhir terungkap arah

Table 3 shows the results of the text data preprocessing process that has been carried out in this study. The initial text data has gone through various preprocessing steps to improve the quality and consistency of the analysis that will be carried out next.

3. Term Frequency Inverse Document Frequency

According to formula (3), TF-IDF generates a numerical value that measures a word's importance in a document relative to the entire collection. A word that appears frequently in one document but rarely in others will have a high TF-IDF value. This technique is useful for document clustering, information retrieval, and text classification, as shown in Table 4.

Table 4. TF-IDF Result

No	Preposeccing Text	Abadi	Bejat	Buzer	...	Capres	Jenius
1	Greindruu sebatas mimpi pimpinan capres abadi	0.675	0.00	0.00	...	0.756	0.00
2	Capres moral bejat	0.00	0.382	0.00	...	0.756	0.00
3	bilang Anis butuh buuzer itu buuzer susah lihat mereka bodoh.	0.00	0.00	0.347	...	0.00	0.00
...							
8876	jenius Prabowo capres buat rakyat Indonesia yakin bisa Amerika	0.00	0.00	0.00	...	0.756	0.487

Table 4 illustrates the TF-IDF outcomes, showcasing the weight assigned to each word in the document, considering both its frequency within that specific document and its prevalence across the entire document collection.

4. Chi-Square Feature Selection

Formula (4) provides the results of feature selection using the Chi-Square method, indicating the Chi-Square value assigned to each feature or word in the dataset. Out of the initial 4752 features, the top 1000 most significant features were selected based on their association with the target class. This approach facilitated the identification of the most informative features for data analysis and classification.

Table 5. Feature Selection Result

No	Text Feature Selection	Feature Selection Score
1	Abadi	0.675
2	Bejat	0.382
3	Buuzer	0.347
...	...	...
1000	Jenius	0.487

Table 5 displays results from the Pseudo-Labelling process, showing selected words, their predicted labels, and feature selection scores. This method labels unlabeled text based on model predictions, expanding the labeled dataset and enhancing model training effectiveness.

5. Pseudo-Labelling

In semi-supervised learning, this method utilizes model predictions on unlabeled data to add extra labels to the training dataset, aiming to enhance model performance through dataset expansion. Table 6 displays the outcomes of leveraging information from model predictions, which enhances model accuracy by enriching the training data.

Table 6. Pseudo-Labelling Results

No	Preposeccing Text	Predict Label
1	ada maka piling capres paling sedikit parpol anis capres	Positive
2	prabowo rakyat arun deklarasi prabowo subianto capres rumah relawan prabowo	Neutral
3	alhamdulillah taun pilih prabowo kalah karir ningkat	Negative
...	...	...
8876	banyak dukung prabowo gibran tulus bangun negeri sahabat teman warga cinta nkri kita pilih capres akhir terungkap arah	Positive

Table 6 displays the results of the Pseudo-Labelling process, where text from the test data is labelled based on predictions to expand the amount of labelled data. This approach aims to enhance the model's performance by utilizing predictions on new unlabeled data, thereby creating a larger and more diverse dataset for training.

6. Classification Accuracy Result

Accuracy stands as a crucial evaluation metric in assessing the performance of classification models. However, in specific situations, additional metrics such as precision, recall, or F1-score also merit attention, depending on the needs and characteristics of the data being analyzed, as shown in Table 7.

Table 7. Hasil Accuracy Model Klasifikasi

Classification Model	Share Proportion	Precision	Recall	F1-score	Accuracy
Linear SVM	70:30	0.7723	0.8788	0.8221	0.8788
KNN		0.8168	0.8485	0.8283	0.8485
Random Forest		0.7723	0.8788	0.8221	0.8788
Linear SVM	80:20	0.8574	0.9038	0.8604	0.9038
KNN		0.8616	0.8987	0.8727	0.8987
Random Forest		0.8765	0.8567	0.8078	0.8567
Linear SVM	90:10	0.8893	0.9200	0.8828	0.9200
KNN		0.8890	0.9189	0.8890	0.9189
Random Forest		0.8893	0.9178	0.8828	0.9178

Table 7 displays experimental results using various machine learning methods: linear SVM, SVM with RBF kernel, polynomial SVM, KNN, and Random Forest. The highest accuracy, 0.9200, is achieved by the linear SVM model at data proportions of 70:30, 80:20, and 90:10. Specifically, at the 90:10 data split, the linear SVM model shows good precision, F1-score, and recall values, demonstrating its effectiveness in this scenario. While detailed results for other data split ratios are not provided, this underscores the superior performance of linear SVM in these experiments. Consideration of factors like computational cost and model training speed is important for selecting the optimal model for practical applications, as shown in Figure 1.

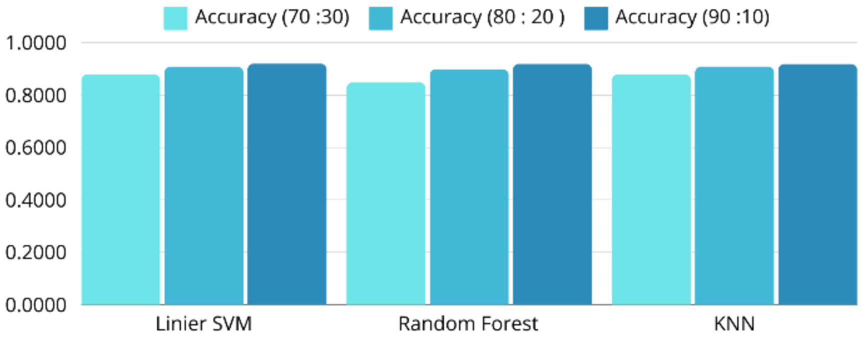

Figure 1. Classification Model Accuracy Result Chart

Figure 1 presents the sentiment analysis results for each participant based on the most significant features identified. These results were obtained using a linear SVM classification with a 90:10 data split, which provided the highest accuracy. This analysis illustrates how the sentiment of each participant is categorized using the classification model, demonstrating the effectiveness of the significant features in accurately identifying sentiment, the sentiment analysis for each participant is obtained as shown in Figure 2.

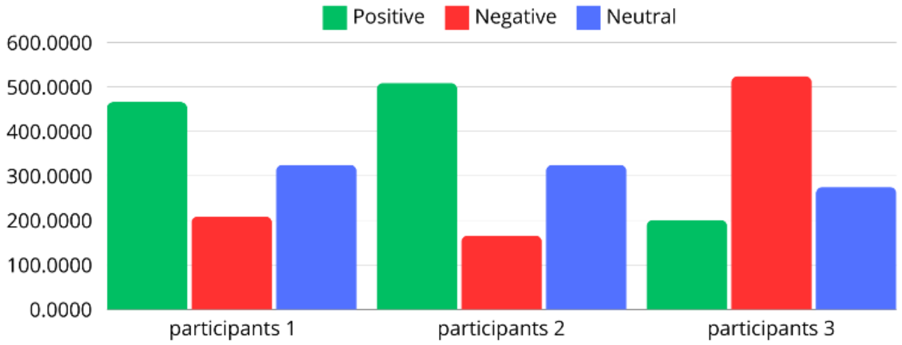

Figure 2. Sentiment Analysis Result Chart

Figure 2 shows the results of sentiment analysis for each participant using linear SVM classification with 90:10 split data, where the most positive is obtained by participant number 2 and the most negative is obtained by participant number 3.

D. CONCLUSION AND SUGGESTIONS

In the previous study, an accuracy of 0.8000 was obtained using Chi-Square feature selection and Naive Bayes classification, while in this study an accuracy of 0.9200 was obtained using Chi-Square feature selection with Pseudo-Labelling in Linear SVM classification. The Chi-Square feature selection method and Pseudo-Labelling technique show excellent results in improving the performance of classification systems, mainly due to their ability to effectively reduce feature dimensions and use unlabelled data for model training. However, it is crucial to study the key factors underlying these performance improvements, such as the proper selection of the Chi-Square threshold and the robustness of Pseudo-Labelling under different labelling accuracies. In addition, it is important to address potential limitations, such as the sensitivity of Chi-Square to the feature independence assumption and the risk of noisy pseudo-labels that adversely affect model training. Future research should prioritise exploring these factors in more diverse and larger data sets to assess generalisability across multiple domains. In addition, it is recommended to investigate alternative classification methods beyond linear SVM, such as KNN and Random Forest, to ensure the sustainability and accuracy of the results obtained.

ACKNOWLEDGEMENT

The authors gratefully acknowledge FAST Ahmad Dahlan University Indonesia, especially Mathematic Laboratory, and Faculty of Data Science and Assoc Prof at Faculty of Engineering, UCSI University, Malaysia for supporting this work.

REFERENCES

Adnan, K., & Akbar, R. (2019). An analytical study of information extraction from unstructured and multidimensional big data. In *Journal of Big Data* (Vol. 6, Issue 1). 56-70 Springer International Publishing. https://doi.org/10.1186/s40537-019-0254-8

Al Walid, M. H., Anisuzzaman, D. M., & Saif, A. F. M. S. (2019). Data Analysis and Visualization of Continental Cancer Situation by Twitter Scraping. *International Journal of Modern Education and Computer Science*, *11*(7), 23–31. https://doi.org/10.5815/ijmecs.2019.07.03

Alshaer, H. N., Otair, M. A., Abualigah, L., Alshinwan, M., & Khasawneh, A. M. (2021). Feature selection method using improved CHI Square on Arabic text classifiers: analysis and application. *Multimedia Tools and Applications*, *80*(7), 10373–10390. https://doi.org/10.1007/s11042-020-10074-6

Arora, N., & Kaur, P. D. (2020). A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. *Applied Soft Computing Journal*, *86*(11), 105936. https://doi.org/10.1016/j.asoc.2019.105936

Asghar, S., Choi, J., Yoon, D., & Byun, J. (2020). Spatial pseudo-labeling for semi-supervised facies classification. *Journal of Petroleum Science and Engineering*, *195*(August), 107834. https://doi.org/10.1016/j.petrol.2020.107834

Chen, R. C., Dewi, C., Huang, S. W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, *7*(1), 52. https://doi.org/10.1186/s40537-020-00327-4

Deta Kirana, Y., & Al Faraby, S. (2021). Sentiment Analysis of Beauty Product Reviews Using the K-Nearest Neighbor (KNN) and TF-IDF Methods with Chi-Square Feature Selection. *Open Access J Data Sci Appl*, *4*(1), 31–042. https://doi.org/10.34818/JDSA.2021.4.71

Ferrario, A., & Naegelin, M. (2020). The Art of Natural Language Processing: Classical, Modern and Contemporary Approaches to Text Document Classification. *SSRN Electronic Journal*, 3(1), 1–51. https://doi.org/10.2139/ssrn.3547887

Garg, S., Panwar, D. S., Gupta, A., & Katarya, R. (2020). A literature review on sentiment analysis techniques involving social media platforms. *PDGC 2020 - 2020 6th International Conference on Parallel, Distributed and Grid Computing*, *3*(1), 254–259. https://doi.org/10.1109/PDGC50313.2020.9315735

Hamzah, M. B. (2021). Classification of Movie Review Sentiment Analysis Using Chi-Square and Multinomial Naïve Bayes with Adaptive Boosting. *Journal of Advances in Information Systems and Technology*, *3*(1), 67–74. https://doi.org/10.15294/jaist.v3i1.49098

Herlawati, H., Trias Handayanto, R., Ekawati, I., Meutia, K. I., Asian, J., & Aditiawarman, U. (2020). Twitter scrapping for profiling education staff. *2020 5th International Conference on Informatics and Computing, ICIC 2020*. 3(1), 23-67. https://doi.org/10.1109/ICIC50835.2020.9288607

Jabbar, A., Iqbal, S., Tamimy, M. I., Hussain, S., & Akhunzada, A. (2020). Empirical evaluation and study of text stemming algorithms. In *Artificial Intelligence Review* (Vol. 53, Issue 8). 5559-5588. Springer Netherlands. https://doi.org/10.1007/s10462-020-09828-3

Krstinić, D., Braović, M., Šerić, L., & Božić-Štulić, D. (2020). *Multi-label Classifier Performance Evaluation with Confusion Matrix*. 3(1), 01–14. https://doi.org/10.5121/csit.2020.100801

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*, 3(8), 66–71. https://doi.org/10.18653/v1/d18-2012

Mohd Nafis, N. S., & Awang, S. (2021). An Enhanced Hybrid Feature Selection Technique Using Term Frequency-Inverse Document Frequency and Support Vector Machine-Recursive Feature Elimination for Sentiment Classification. *IEEE Access*, *9*(Ml), 52177–52192. https://doi.org/10.1109/ACCESS.2021.3069001

Paudel, S., Prasad, P. W. C., Alsadoon, A., Islam, M. R., & Elchouemi, A. (2019). Feature selection approach for twitter sentiment analysis and text classification based on chi-square and naïve bayes. *Advances in Intelligent Systems and Computing*, *842*(11), 281–298. https://doi.org/10.1007/978-3-319-98776-7_30

Sakthi Vel, S. (2021). Pre-Processing techniques of Text Mining using Computational Linguistics and Python Libraries. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, *3(1),* 879–884. https://doi.org/10.1109/ICAIS50930.2021.9395924

Sarica, S., & Luo, J. (2021). Stopwords in technical language processing. *PLoS ONE*, *16*(8 August), 1–13. https://doi.org/10.1371/journal.pone.0254937

Shan Lee, V. L., Gan, K. H., Tan, T. P., & Abdullah, R. (2019). Semi-supervised learning for sentiment classification using small number of labeled data. *Procedia Computer Science*, *161(2019)*, 577–584. https://doi.org/10.1016/j.procs.2019.11.159

Singh, K. N., Devi, S. D., Devi, H. M., & Mahanta, A. K. (2022). A novel approach for dimension reduction using word embedding: An enhanced text classification approach. *International Journal of*

Information Management Data Insights, *2*(1), 100061. https://doi.org/10.1016/j.jjimei.2022.100061

Singh, N. K., Tomar, D. S., & Sangaiah, A. K. (2020). Sentiment analysis: a review and comparative analysis over social media. *Journal of Ambient Intelligence and Humanized Computing*, *11*(1), 97–117.https://doi.org/10.1007/s12652-018-0862-8

Syrotkina, O., Aleksieiev, M., Moroz, B., Matsiuk, S., Shevtsova, O., & Kozlovskyi, A. (2020). Mathematical Methods for optimizing Big Data Processing. *Proceedings - International Conference on Advanced Computer Information Technologies, ACIT*, 1(9), 170–176. https://doi.org/10.1109/ACIT49673.2020.9208940

Tubishat, M., Abushariah, M. A. M., Idris, N., & Aljarah, I. (2019). Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. *Applied Intelligence*, *49*(5), 1688–1707. https://doi.org/10.1007/s10489-018-1334-8

Yang, A., Zhang, J., Pan, L., & Xiang, Y. (2016). Enhanced twitter sentiment analysis by using feature selection and combination. *Proceedings - 2015 International Symposium on Security and Privacy in Social Networks and Big Data, SocialSec 2015*, 9(*November)*, 52–57. https://doi.org/10.1109/SocialSec2015.9

Yang, W., Zhang, R., Chen, J., Wang, L., & Kim, J. (2023). Prototype-Guided Pseudo Labeling for Semi-Supervised Text Classification. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, *1(july)*, 16369–16382. https://doi.org/10.18653/v1/2023.acl-long.904