International Journal of Applied Research and Sustainable Sciences (IJARSS)
Vol. 3 No. 9, 2025: 743 - 762

Classification of South Jakarta Language Sentence Structure (Indonesian-English Code Mixing)

Ari Yoko Saputra[1]*, Arif Bijaksana Putra Negara[2], Hafiz Muhardi[3]
Program Studi Informatika, Fakultas Teknik, Universitas Tanjungpura
Corresponding Author: Ari Yoko Saputra
yokosaputra@student.untan.ac.id

A R T I C L E I N F O

Keywords: Code-Mixing, South Jakarta Language, Classification, Support Vector Machine, Random Forest

Received : 14 July
Revised : 14 August
Accepted: 19 September

A B S T R A C T

Twitter can present a variety of information that keeps various users from various parts of the world always updated on what is being discussed (tweet). Various information is presented in various ways and deliveries, one of which is by using the South Jakarta language style mix, which is a code mix that mixes various vocabularies using both Indonesian and English in their conversations so that they are considered current for young people in Indonesia. Based on this phenomenon, sentence classification is made using the Support Vector Machine and Random Forest algorithm methods to classify the types of code mixing, namely alternation, insertion, and congruent lexicalization. Attributes on the data were Tag and Tag + Text. Text weighting is used with unigram, bigram, and trigram. The evaluation of the research used 10-fold cross validation for training data and a confusion matrix for test data. The best data test results on Tag data obtained by the Support Vector Machine using bigrams obtained an accuracy value of 97.33%, while the best data test results on Tag + Text data obtained by the Random Forest algorithm using unigrams obtained an accuracy value of 92%.

INTRODUCTION

Modern society is linked to globalism and openness to other nations. Contact between different speakers leads to language contact. A variety of phenomena within the domain of linguistics, including bilingualism, may emerge as a consequence of language contact. In the domain of linguistics, a notable phenomenon occurs when individuals possess the capacity to articulate themselves across multiple languages, often facilitated by interactions between speakers of different languages. The number of speakers who have mastered two or more languages is increasing, regardless of the extent of their proficiency. Consequently, this phenomenon can be designated as bilingual or multilingual (Dahniar & Sulistyawati, 2023). The phenomenon of linguistic diversity is a consequence of all communication activities. The linguistic variety is likely to expand if a significant number of speakers utilize the language across a considerable geographical area (Prasasti, 2020). Individuals who possess the ability to utilize two languages in their social interactions are designated as bilingual. The phenomenon of code-mixing, defined as the integration of two or more languages or linguistic varieties into a single utterance, is characterized by a linguistic pattern that involves the intermingling of distinct languages (Mulyani, 2020; Alimin & Ramaniyar, 2020).

The Indonesian nation has adopted a unified language. The inception of the use of the Indonesian language occurred on October 27–28, 1928. This event was initiated subsequent to the Second Youth Congress in Jakarta, which was referred to as the "Sumpah Pemuda" (Youth Pledge) (Sukaesih et al., 2023). According to Sumarni in Zaqi et al. (2023), the "Sumpah Pemuda" event was held, during which the pledge was stated. The aforementioned pledge pertained to notions of a nation, homeland, and a unified language, specifically referencing Indonesia. The Indonesian language has been subject to regulation and determination by the Government of the Republic of Indonesia (Negara Kesatuan Republik Indonesia) in two articles. The first is Article 36 of the 1945 Constitution, which pertains to the National Flag, Language, and Coat of Arms, as well as the National Anthem. The second is Article 25 of Law Number 24 of 2009 concerning the National Language (Ganiadi et al., 2023). According to Brutt-Giffler and Crystal in Perangin-Angin et al. (2023), the contemporary period of globalization has made English a global language. The interconnectedness of the world, facilitated by technology, has created many business opportunities. English has become essential for both oral and written communication. Each language should be used for what it is designed for. Native Indonesian speakers should prioritize the Indonesian language to ensure its preservation and advancement. Mastery of English as an international language is also crucial for the advancement of knowledge.

In this regard, Indonesian language diversity has developed quite rapidly, as evidenced by the emergence of the South Jakarta youth's language trend, colloquially referred to as Bahasa Jaksel (Jaksel language), where "Jaksel" is short for "Jakarta Selatan" (South Jakarta). The concept of "Bahasa Jaksel" was introduced by netizens and refers to a mixture of Indonesian and English that is inserted into a sentence during a conversation (Setiawan, 2023). As Putri in

Wicaksono (2022) notes, the trend of Jaksel language is not a recent development; it has been in existence for several years. The emergence of this language on social media can be traced to the year 2018, when its initial appearance was observed on the Twitter platform. Twitter is a popular social media platform for young people. People use it to share information and opinions. In some cases, the code-mixing may be disproportionate, potentially leading to misinterpretation of the text by the reader (Puspita et al., 2022). A thorough examination of the extant statistical data reveals that, as of April 2024, Twitter users (hereafter X) in Indonesia number approximately 24.85 million, placing the platform fourth in global rankings (Statista, 2024).

In light of the observed phenomenon, it can be posited that the code-mixing present in Jaksel language sentences disseminated on the social media platform Twitter can be classified according to the code-mixing typology proposed by Muysken, encompassing three categories: insertion, alternation, and congruent lexicalization (Melansari, 2023). The classification process is executed through two distinct methodologies: Support Vector Machine and Random Forest algorithms. The Support Vector Machine algorithm is a statistical learning method that finds the maximum point and the best hyperplane that can separate two classes (Hassanah et al., 2023). Whereas the Random Forest algorithm employs the bagging method, utilizing a voting system that prioritizes the most frequently selected option among multiple decision trees (Husin, 2023). The two algorithms are both part of a machine learning model that employs an algorithm capable of calculating the weight of each word in a text. This calculation is referred to as Term Frequency-Inverse Document Frequency (TF-IDF) (Naufal et al., 2023).

LITERATURE REVIEW

1. Twitter

The term "Twitter" is derived from the English word for "tweeting". This particular social media site facilitates the expression of thoughts and actions in real time, enabling users to share information with a vast audience. Twitter is a social media website owned and operated by Twitter Inc. that provides a network for microblogging, enabling users to send and read messages known as "tweets." It is possible for users to articulate a concept or reflections in up to 280 characters. (Girnanfa & Susilo, 2022). Following the transition of ownership and operation to X Corp. in 2023, Twitter underwent a rebranding initiative, changing its name to X.

2. Code-Mixing

Octavia in Jannah & Anggraini (2023) posited that code-mixing is defined as an activity in which speakers integrate two or more language elements to communicate. Code-mixing is a linguistic phenomenon in which a speaker employs a blend of different languages or dialects during communication, often selecting words or phrases that they believe to be readily comprehensible to their interlocutor. Code-mixing is also employed to facilitate language mastery in communication with interlocutors. Code-mixing constitutes a particular

sociolinguistic phenomenon. That is to say, it is the mixing of two or more languages in the same speaking condition.

Muysken introduces a typology of code-mixing in sentences, including the following categories:

(1) Insertion

As illustrated by Figure 1, element "a" represents a phrase in the primary language. Element "b" represents a phrase in the second language that has been inserted by the speaker. The use of insertion is exemplified by the sentence "*Mereka setiap akhir pekan selalu* self healing *bersama tanpa sepengetahuan saya*". In this instance, the phrase "self healing" has been inserted by the speaker as a component of the discourse.

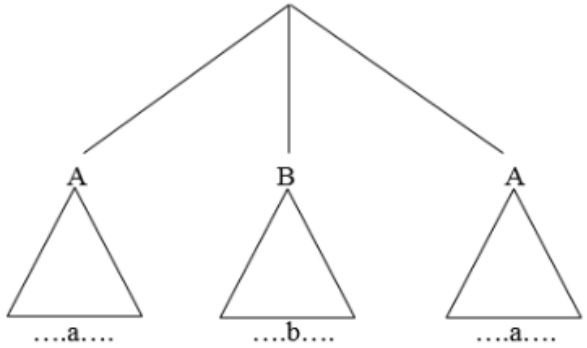

Figure 1. Illustration on Insertion in code-mixing

(2) Alternation

As illustrated by Figure 2, elements "A" and "B" are representative of two distinct languages, thereby illustrating the phenomenon of language alternation in code-mixing, manifesting as utterances produced by speakers. The use of alternation is exemplified by the sentence "I think I can, *soalnya setiap aku menyanyikan suatu lagu, penonton terhibur*".

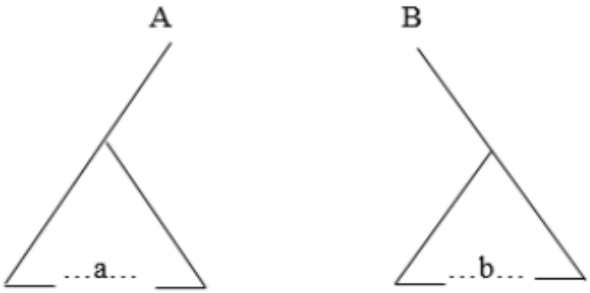

Figure 2. Illustration on Alternation in code-mixing

(3) Congruent Lexicalization

As illustrated by Figure 3, The elements "A" and "B" are representations of the two distinct languages. In this category, speakers exhibit a greater propensity to amalgamate the two languages when viewed from a grammatical perspective. This amalgamation entails the incorporation of lexical items from both languages to enrich phrases. The use of congruent lexicalization is exemplified by the sentence "Meeting *hari ini akan membahas tentang* urgent agenda *yang akan dilakukan* this week".

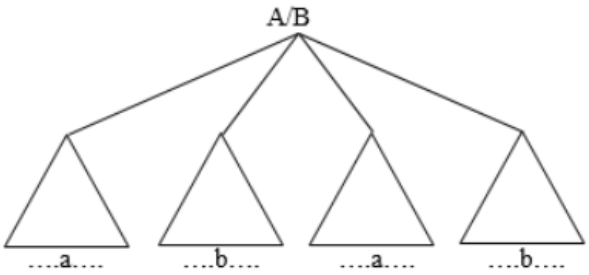

Figure 3. Illustration on Congruent Lexicalization in code-mixing

3. Text Preprocessing

According to Kearney et al., text preprocessing entails the initial cleaning and fixing of data to ensure its structural integrity prior to progression to subsequent stages of processing. (Sari et al, 2023). There are several stages in text preprocessing, which are as follows:

(1) Case folding, standardization process that converts all letter elements in the text into lowercase letters. (Rofiqi & Akbar, 2024).
(2) Cleaning, the process entails the elimination of superfluous characters with the objective of enhancing the text's significance for a particular purpose. (Sengar et al., 2021).
(3) Parsing, according to Jain et al., the process of parsing text entails the decomposition of a given text so that its processed text can be determined. (Natalie et al., 2023).
(4) Tokenizing, word-by-word truncation of text based on a predefined data dictionary is a common practice that is employed to identify words of value and facilitate the identification of the frequency of data in the corpus. (Fadhilah & Indriyanti, 2023).

4. Regular Expression

Regular expression (regex) is a library that is used to perform the cleaning process. (Vindua & Zailani, 2023). Regex are a kind of linguistic construct that is used to match text based on predefined patterns. They are especially useful in complex scenarios (Razaq et al., 2023).

5. Transformation-Based Learning

Transformation-Based Learning was initially developed by Eric Bill and later refined by Ramshaw and Marcus through the integration of the IOB method, a technique that involves a non-recursive word structure initially characterized by I, O, and B marks. Within this method, the letter I denotes an interior element, O denotes an exterior element, and B marks the leftmost item that follows another element in the sequence. (Alves et al., 2024).

6. Term Frequency-Inverse Term Frequency

The two concepts of Term Frequency-Inverse Term Frequency (TF-IDF) converge when a word appears in a document and the document in question contains a different word. The importance of a word in a document lies in its presence, not its significance. The prevalence of a word in a document is

determined by the total number of words in said document. This may indicate that the correlation between words and documents can be substantial if the number of words in a document is high and the frequency of the document's content containing the word is significant during data processing (Anugrah, 2023).

The equation employed to calculate the probability (TF) of the appearance of word *i* in document *j* is as follows:

$$Tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Description:

Tf_{ij} = TF of word *i* in document *j*
$n_{i,j}$ = Number of words *i* in document *j*
$\sum_k n_{k,j}$ = Total words in document *j*

The equation employed to calculate the general relevance of a specific word (IDF) is as follows:

$$Idf_i = log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

Description:

Idf_i = IDF on the word *i*
$|D|$ = Total number of documents
$\{d_j : t_i \in d_j\}$ = Number of documents in which word i appears

The TF and IDF values are multiplied by the following equation:

$$TfIdf_{i,j} = Tf_{i,j} \times Idf_i$$

7. Grid Search

Grid search constitutes an alternative approach for identifying the optimal parameters within the model, thereby ensuring the accurate prediction of data. Grid search is a hyperparameter method that is used to achieve optimal performance. (Azmi, 2023).

8. Support Vector Machine

Support vector machine (SVM) is a machine learning algorithm that employs a hyperplane function to delineate components of each class. Hyperplane is a function that is useful as a separator between existing classes. (Setiana et al., 2023).

The fundamental premise of the SVM algorithm revolves around the optimization of the margin, defined as the extent of separation among the data categories illustrated in Figure 4 (Werdiningsih et al., 2020):

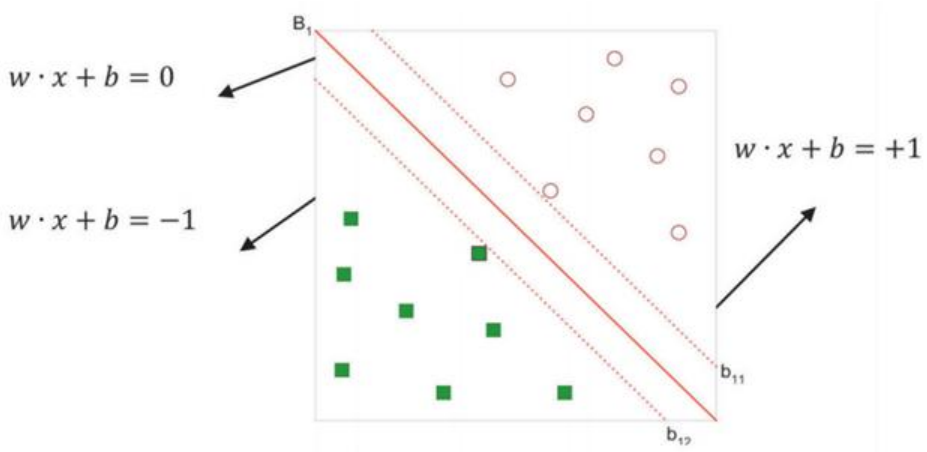

Figure 4. Hyperlane illustration

As illustrated in Figure 4, the dotted red lines, designated as b_{11} and b_{12}, represent support vectors that are determined by the weight and bias values. The red line in the center, B_1, serves as a hyperplane, effectively demarcating the data into two distinct classes. (Setiana et al., 2023). The following equation will be used to find the hyperplane:

$$x_i\, w + b \geq +1 \; for \; y_i = +1$$
$$x_i\, w + b \geq -1 \; for \; y_i = -1$$

Description:

w = Normal plane

b = Plane position relative to the center

9. Random Forest

Random Forest (RF) is a machine learning algorithm that integrates numerous decision trees and bagging techniques, consolidating them into a single forest of trees. The integration of outcomes from multiple decision trees has the potential to yield a more precise and comprehensive prediction model (Wibowo & Syahputra, 2022).

The Random Forest algorithm employs the bootstrap method, which involves the generation of bootstrap samples from a given sample and the replacement of the data source. This process is illustrated in Figure 5. (Sarosa et al., 2022):

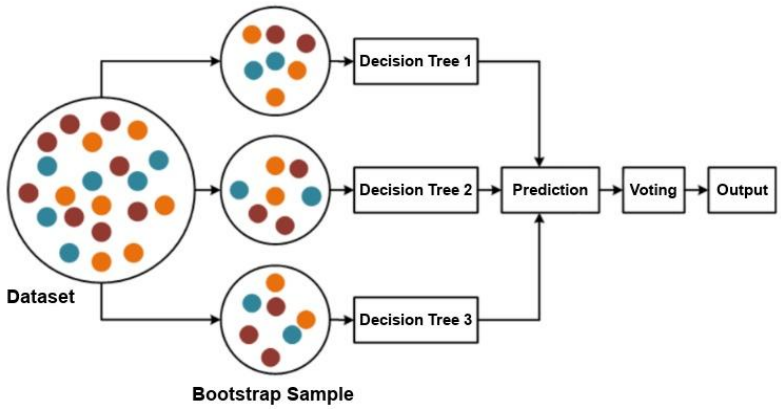

Figure 5. Bootstrap in Random Forest algorithm

The calculation stage commences with the identification of features that serve as the foundation of the decision tree. These features are determined by measuring the entropy value. Subsequently, the information gain is calculated, resulting in the acquisition of data. Once this is complete, the decision tree can be generalized (Wibowo & Syahputra, 2022).

The calculation for entropy and information gain, in general, is as follows:

$$Entropy(S) = \sum_{i=1}^{n} P_i \, log2P_i$$

Description:
S = Set of cases
P_i = Proportion of S_i against S
n = Number of S partitions

$$Gain(S, A) = E(S) \sum_{i=1}^{n} \frac{|S_i|}{|S|} E(S_i)$$

Description:
E = Entropy
S = Set of cases
A = Attribute
n = Number of partitions of attribute A
$|S_i|$ = Number of cases in i-th partition
$|S|$ = Number of cases in S

10. K-Fold Cross Validation

K-fold cross-validation is a data testing method that divides data into multiple parts. One part is designated as test data, and the rest are used for training. This process is repeated k times, ensuring each part of the data is used as test data once (Rismayani et al., 2023).

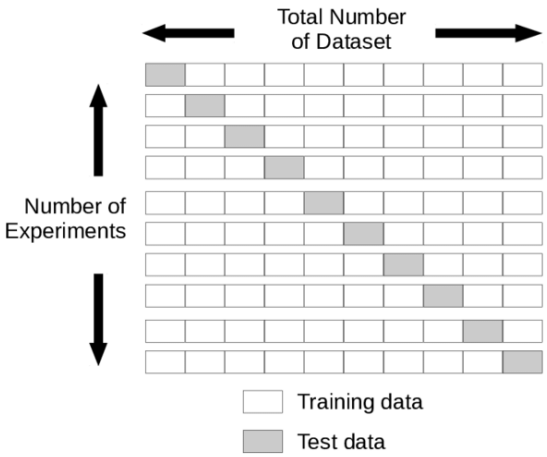

Figure 6. Cross Validation

11. Confusion Matrix

Confusion matrix is a type of table with a square matrix format. This table can be used to determine the difference between the actual value and the predicted value of a categorical variable (Baker, 2021; Rohman, 2021). As posited by Jiawei Han and Gorunescu, the confusion matrix is derived during the

training and testing phases to facilitate the determination of the accuracy of a classification decision. (Indriyanto, 2021).

A confusion matrix is a statistical tool used to predict a categorical variable in concise models, like statistical models or machine learning algorithms. The result is classified into rows and columns, and cells at the intersection show the frequency of true values.

Table 1. Confusion Matrix

Classification		**Prediction Class Result**	
		Positive	**Negative**
Actual Class	**Positive**	TP	FN
	Negative	FP	TN

According to Table I, the following four terms are present in the confusion matrix (Rohman, 2021):

- True Positive (TP) is defined as the total actual data of positive classes that are classified as positive classes in the dataset.
- True Negative (TN) is defined as the total actual data of negative classes that are classified as negative classes in the dataset.
- False Positive (FP) is defined as the total actual data of negative classes that are classified as positive classes in the dataset.
- False Negative (FN) is defined as the total actual data of positive classes that are classified as negative classes in the dataset.

The inaccuracy of the model in predicting outcomes can be elucidated through the utilization of a confusion matrix. The values in the confusion matrix can be used to evaluate the model in terms of accuracy, precision, recall, and F1-score. Accuracy is defined as the extent to which a model can accurately classify data. Precision is defined as the ratio of true positive (TP) predictions to the sum of all positive predictions. Recall is defined as the ratio of True Positive (TP) predictions to all true positive results. The F1-score is defined as the harmonic mean of precision and recall itself. (Indriyanto, 2021).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2(Precision)(Recall)}{Precision + Recall}$$

METHODOLOGY

This research will entail a series of steps, each employing a distinct method, as delineated below.

1. Data Collection

The data utilized in this study is composed of sentence data extracted from a collection of tweets that were retrieved via the Twitter API. The data was collected based on English keywords frequently used in Jaksel language (some of which are: which is, literally, honestly, in my opinion, etc.) and Indonesian language selection in full tweets. Subsequently, the data undergoes filtration to eliminate redundancy.

2. Text Preprocessing

The tweet data to be obtained is characterized by its raw and unstructured nature. Consequently, the data must undergo a process referred to as text preprocessing to ensure its transformation into structured data, aligning with the requisite format for conducting research. A number of the text preprocessing operations utilize regular expressions. The utilization of regular expressions is imperative in the identification of a string that serves to enhance the text string of the data to be processed.

The text preprocessing process employed in this research involves several steps, including case folding, cleaning, parsing, and tokenizing.

3. Labeling

Subsequently, data that has undergone text preprocessing is assigned labels based on the type of code mix, namely insertion, alternation, and congruent lexicalization (Melansari et al., 2023).

4. Rule Pattern Determination

The text that has been labeled must be assured of its labeling by adding a pattern to each word. Modified POS-Tagging provides patterns to each word, which are manually verified to match the rules. In this particular case, each word is assigned a linguistic tag, and the location of the word is represented by another word.

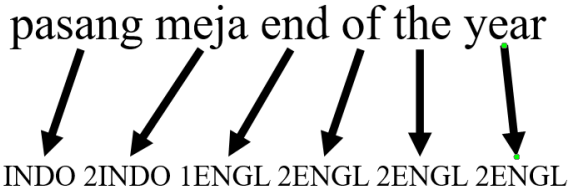

Figure 7. Rule patterns in the sample sentences of the Jaksel language

Description:

1 : The first word in a language
2 : The next word in a language
INDO : Indonesian word
ENGL : English word

5. Weighting

The application of the term frequency–inverse document frequency (TF–IDF) method will be employed to assign a weight to each term. Initially, the TF and IDF values are calculated separately. Subsequently, the two values are multiplied to obtain the TF–IDF result. Furthermore, n-grams are employed to ascertain the combination of patterns. In each scenario, three types are employed: unigrams (n-gram = 1), bigrams (n-gram = 2), and trigrams (n-gram = 3). The purpose of these calculations is to determine the frequency of the input data. (Komputer, 2013).

6. Classification

The present study will employ Support Vector Machine (SVM) and Random Forest (RF) algorithms for the purpose of classification. A comprehensive examination of numerous parameters will be conducted to ascertain the most suitable parameters for each algorithm. These parameters include, but are not limited to, the following:

Table 2. Grid Parameters Queried in Each Algorithm

No.	Algorithm	Parameter Used	Parameter Value of Interest
1	Support Vector Machine (SVM)	SVM Type	SVM-C
		Kernel	RBF
		Gamma	1-101 (10 Steps)
		C	1-101 (10 Steps)
2	Random Forest (RF)	Criteria	Entropy
		Random State	42
		Total Trees	1-101 (10 Steps)
		Depth Maximal	1-101 (10 Steps)

7. Testing

Subsequently, classified data will enter the testing phase. The objective of this phase is to employ classified data as a performance measurement tool to calculate the value of the classification results.

The research utilizes a testing method known as K-fold cross-validation, where the number of folds is set at K = 10. This configuration is referred to as 10-fold cross-validation. The training data is subjected to this method, while the test data employs a confusion matrix.

In the testing scenario, 12 scenarios will be applied in this study, including:

Table 3. Testing Scenario

Scenario	Algorithm	Attributes Used	Number of n-Grams
1	SVM	Tag	1
2	SVM	Tag + Text	1
3	SVM	Tag	2
4	SVM	Tag + Text	2
5	SVM	Tag	3
6	SVM	Tag + Text	3

7	RF	Tag	1
8	RF	Tag + Text	1
9	RF	Tag	2
10	RF	Tag + Text	2
11	RF	Tag	3
12	RF	Tag + Text	3

However, the scenarios will be divided based on the utilization of attributes to ensure equitable comparison of data between attributes, yielding the following:

Table 4. Testing Scenario with Tag Attributes

Scenario	Algorithm	Attributes Used	Number of n-Grams
1	SVM	Tag	1
3	SVM	Tag	2
5	SVM	Tag	3
7	RF	Tag	1
9	RF	Tag	2
11	RF	Tag	3

Table 5. Testing Scenario with Tag + Text Attributes

Scenario	Algorithm	Attributes Used	Number of n-Grams
2	SVM	Tag + Text	1
4	SVM	Tag + Text	2
6	SVM	Tag + Text	3
8	RF	Tag + Text	1
10	RF	Tag + Text	2
12	RF	Tag + Text	3

RESEARCH RESULT

This section elucidates the findings of the research conducted, which encompass the following.

1. **Data Collection**

   The data is collected through the scraping of data from Twitter using the RapidMiner application and Twitter API. The data set under consideration consists of Indonesian tweets collected on August 1, 2022. The search utilizes English keywords that frequently appear in tweets designated as the language of Jaksel.

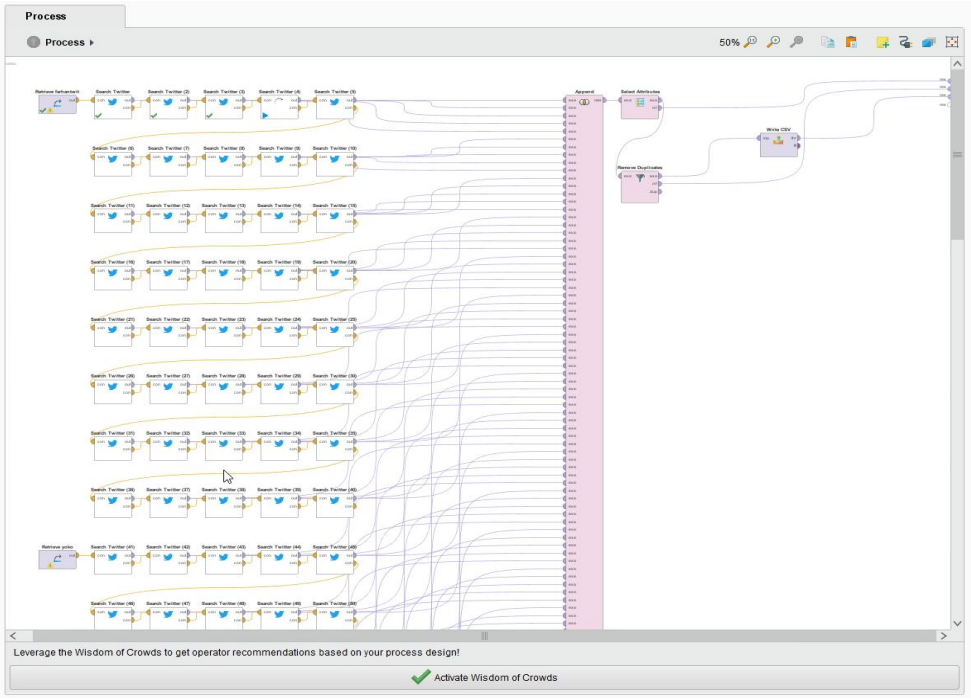

Figure 8. Scraping tweets data

2. Text Preprocessing

The data that has been extracted is not yet suitable for use due to its disorganized state. In order to prepare the data for further processing, it is necessary to implement a data preprocessing step that will enhance the structure of the data. Text preprocessing is facilitated by the utilization of the Jupyter Lab application, which employs the Python programming language. The text underwent several preprocessing steps, including cleaning, case folding, parsing, and tokenizing.

3. Tagging and Labeling

Subsequently, the sorted data is tagged with modified POS-Tags and incorrect tags. The tagged data is subsequently labeled with insertion, alternation, and congruent lexicalization.

4. Data Execution

The data that has been labeled is executed using Google Colab, which utilizes the Python programming language and several libraries, including numpy, pandas, sklearn, and others, to run the functions available in the library. These functions are then executed based on the research methodology previously described above, which calculates the performance of the classification.

5. Evaluation

The evaluation results are obtained by executing data with various pre-prepared scenarios using cross validation on training data and a confusion matrix on test data. The evaluation results are displayed with accuracy, precision, recall, and f1-score benchmarks.

The outcomes of the training data evaluation are displayed in Tables 6 and 7, which utilize the "Tag" and "Tag + Text" attributes.

Table 6. Training Data Performance with Tag Attributes

Scenario	Method	Training Data Performance (in %)			
		Accuracy	Precision	Recall	f1-score
1	SVM Tag 1 n-Gram	91,98	92,82	91,98	91,92
3	SVM Tag 2 n-Gram	97,38	97,40	97,38	97,38
5	SVM Tag 3 n-Gram	96,74	96,75	96,74	96,74
7	RF Tag 1 n-Gram	92,84	93,61	92,84	92,79
9	RF Tag 2 n-Gram	97,23	97,25	97,23	97,23
11	RF Tag 3 n-Gram	97,38	97,40	97,38	97,38

Table 7. Training Data Performance with Tag + Text Attributes

Scenario	Method	Training Data Performance (in %)			
		Accuracy	Precision	Recall	f1-score
2	SVM Tag + Text 1 n-Gram	78,52	78,71	78,52	78,48
4	SVM Tag + Text 2 n-Gram	91,09	91,45	91,09	91,16
6	SVM Tag + Text 3 n-Gram	89,73	90,16	89,73	89,83
8	RF Tag + Text 1 n-Gram	89,56	90,18	89,56	89,57
10	RF Tag + Text 2 n-Gram	91,06	91,24	91,06	90,97

12	RF Tag + Text 3 n-Gram	91,36	91,56	91,36	91,31

In addition to the performance of the validated training data, the evaluation results of the training data are shown in Tables 8 and 9, which use the "Tag" and "Tag + Text" attributes.

Table 8. Testing Data Performance with Tag Attributes

Scenario	**Method**	**Testing Data Performance (in %)**			
		Accuracy	**Precision**	**Recall**	**f1-score**
1	SVM Tag 1 n-Gram	91,11	92,04	91,11	91,04
3	SVM Tag 2 n-Gram	97,33	97,33	97,33	97,33
5	SVM Tag 3 n-Gram	96,44	96,44	96,44	96,44
7	RF Tag 1 n-Gram	92,44	93,31	92,44	92,36
9	RF Tag 2 n-Gram	96,67	96,68	96,67	96,67
11	RF Tag 3 n-Gram	96,89	96,91	96,89	96,89

Table 9. Testing Data Performance with Tag + Text Attributes

Scenario	**Method**	**Testing Data Performance (in %)**			
		Accuracy	**Precision**	**Recall**	**f1-score**
2	SVM Tag + Text 1 n-Gram	78,89	79,29	78,89	78,89
4	SVM Tag + Text 2 n-Gram	56,44	70,70	56,44	53,60
6	SVM	50,44	70,18	50,44	44,82

	Tag + Text 3 n-Gram				
8	RF Tag + Text 1 n-Gram	92,00	91,99	92,00	91,96
10	RF Tag + Text 2 n-Gram	73,78	80,39	73,78	73,22
12	RF Tag + Text 3 n-Gram	75,56	82,21	75,56	72,71

DISCUSSION

A multitude of analyses can be described in light of the research that has been conducted. The following are some of the analyses that can be described:

1) In the vast majority of cases, there is an instance of overfitting. Overfitting is a condition in which the data is initially trained to exhibit optimal performance; however, when subsequently tested, its performance can exhibit a decline. It is notable that the most substantial overfitting is observed in scenarios 2 and 4, where the performance of the training data validation results can exceed 89%, yet the performance of the test data, particularly the accuracy, recall, and f1-score values, is below 57%. In scenarios 1 and 8 with Tag + Text and the use of 1 n-Gram, overfitting does not occur. The Support Vector Machine (SVM) algorithm demonstrates a 0.37% discrepancy in accuracy, 0.58% in precision, 0.37% in recall, and 0.41% in f1-score. The Random Forest (RF) algorithm exhibits a 90% training data threshold, yet its performance on test data can exceed 91%.
2) A comparison of the performance results reveals that the training data and test data exhibit identical accuracy and recall values. It is noteworthy that certain performance results exhibit uniform values across various metrics, such as accuracy, precision, recall, and F1-score. This uniformity is observed in scenario 3, which demonstrates a 97.33% value, and scenario 5, which exhibits a 96.44% value.

CONCLUSIONS AND RECOMMENDATIONS

A comprehensive analysis of extant research findings has yielded several salient conclusions. First, the results of the data that has been trained and tested demonstrate the optimal performance of the Random Forest algorithm in nearly all scenarios. The optimal scenario for language tag data from sentences is the Random Forest algorithm with three n-grams, which attained an accuracy value of 97.38% on the training data. The same algorithm and method yielded accuracy of 96.89% on the test data. The optimal Tag language and Text data scenario was achieved through the Random Forest algorithm with 3 n-Gram, yielding 91.36% accuracy on training data. The Support Vector Machine with 1 n-Gram algorithm achieved 92% accuracy on test data.This finding indicates that the Random Forest

algorithm is a more effective tool for classification in this particular study. Second, the phenomenon of "overfitting" can lead to a decline in the performance of training data relative to test data in a range of scenarios. This is particularly evident in the Support Vector Machine algorithm, where significant overfitting is observed.

The findings indicate that overfitting in both algorithms, particularly Support Vector Machine and Random Forest, can lead to a decline in the performance of test data relative to training data. Consequently, there have been recommendations to enhance the application of this model by introducing or removing features to mitigate the likelihood of overfitting. The text data has been tagged semi-manually; therefore, further research is necessary to develop a special library that facilitates automatic tagging. Furthermore, the development of this model can be adapted for generic applications to implement future research.

ADVANCED RESEARCH

Based on this research, several additions can be made for the future. First, there should be a system that automatically detects language tags on words using a library for labeling text data. Second, more varied methods should be used, such as data acquisition, algorithms, and data processing. Finally, a generic application of this research should be created to test it with data outside the scope of the research.

REFERENCES

Alimin, A. A., & Ramaniyar, E. (2020). *Sosiolinguistik dalam Pengajaran Bahasa: Studi Kasus Pendekatan Dwi Bahasa di Sekolah Dasar Kelas Rendah*. Putra Prabowo Perkasa.

Alves, D., Thakkar, G., & Tadić, M. (2025). UNER: Universal Named-Entity Recognition Framework. *Event Analytics across Languages and Communities*, 3-15. https://doi.org/10.1007/978-3-031-64451-1_1

Anugrah, R. R. (2023). Penerapan Cosine Similarity dan Pembobotan Tf-Idf untuk Klasifikasi Pengaduan Masyarakat Berbasis Web (Studi Kasus : Bagwassidik Ditreskrimum Polda Kalbar). *Coding: Jurnal Komputer dan Aplikasi, 11*(1), 100-109. https://doi.org/10.26418/coding.v11i1.55598

Azmi, U. (2023). Analisis Perbandingan Klasifikasi dan Penerapan Teknik SMOTE Dalam Imbalanced Data Pada Credit Card Default. *Jurnal Sains Dan Seni ITS (e-Journal), 12*(2), D127-D134. https://doi.org/10.12962/j23373520.v12i2.

Dahniar, A., & Sulistyawati, R. (2023). Analisis Campur Kode Pada Tiktok Podcast Kesel Aje Dan Dampaknya Terhadap Eksistensi Berbahasa Anak Milenial: Kajian Sosiolinguistik. *ENGGANG: Jurnal Pendidikan, Bahasa, Sastra, Seni, Dan Budaya, 3*(2), 55–65. https://doi.org/10.37304/enggang.v3i2.8988

Fadhilah, P. N., & Indriyanti, A. D. (2023). Analisis Sentimen terhadap Opini Publik Mengenai Childfree dalam Pernikahan pada Twitter Menggunakan K-Nearest Neighbor (K-NN). *Journal of Informatics and Computer Science (JINACS), 5*(1), 58-62. https://doi.org/10.26740/jinacs.v5n01.p58-62

Ganiadi, M., Asyamsi, M. R., Tamirullah, M., & Sugana, M. T. B. D., (2023). Peran Pendidikan Non Formal terhadap Perkembangan Bahasa Indonesia. *Jurnal Ilmu Pendidikan Muhammadiyah Kramat Jati, 4*(1), 9–13. https://doi.org/10.55943/jipmukjt.v4i1.45

Girnanfa, F. A., & Susilo, A. (2022). Studi Dramaturgi Pengelolaan Kesan Melalui Twitter Sebagai Sarana Eksistensi Diri Mahasiswa di Jakarta. *Journal of New Media and Communication, 1*(1), 58–73. https://doi.org/10.55985/jnmc.v1i1.2

Hassanah, I. N, Faisal, S., & Siregar, A. M. (2023). Perbandingan Algoritma Support Vector Machine dengan Decision Tree pada Aplikasi Ruang Guru. *Kumpulan Jurnal Ilmu Komputer (KLIK), 10*(1), 39-50. https://dx.doi.org/10.20527/klik.v10i1.602

Husin, N. (2023). Komparasi Algoritma Random Forest, Naïve Bayes, dan Bert Untuk Multi-Class Classification Pada Artikel Cable News Network (CNN). *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi dan Sistem Komputer, 7*(1), 75-84. https://doi.org/10.55886/infokom.v7i1.608

Indriyanto, J. (2021). *Algoritma K-Nearest Neighbor untuk Prediksi Nasabah Asuransi*. Penerbit NEM.

Komputer, W. (2013). The Best Encryption Tools. Elex Media Komputindo.

Melansari, N., B, A. W., Adu, B., & Narni, N. (2022). Code Mixing Used by the Teacher in Teaching English at SMP Negeri 14 Baubau. *International Journal of Education, Language, Literature, Arts, Culture, and Social Humanities, 1*(1), 14–28. https://doi.org/10.59024/ijellacush.v1i1.22

Mulyani. (2020). *Praktik Penelitian Linguistik*. Deepublish.

Natalie, C., Mawardi, V. C., & Sitorus, M. D. L. (2023). Optical Character Recognition Menggunakan Uipath dan Pencocokan Data Sertifikat dengan Algoritma Levenshtein Distance. *Jurnal Serina Sains, Teknik dan Kedokteran, 1*(1), 18-26. https://doi.org/10.24912/jsstk.v1i1.22747

Naufal, M. F., Arifin, T., & Wirjawan, H. (2023). Analisis Perbandingan Tingkat Performa Algoritma SVM, Random Forest, dan Naïve Bayes untuk Klasifikasi Cyberbullying pada Media Sosial. *Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika), 8*(1), 82-90. http://dx.doi.org/10.30645/jurasik.v8i1.544

Perangin-Angin, D. M., Manggala, S. A., Fitriati, A., Putranti, A., Rosiandani, N. L. P., Puri, A. D., & Pukan, E. O. (2023). Menjawab Kebutuhan Pekerja Migran Indonesia Berketerampilan Bahasa Inggris sebagai Bahasa Global. *Abdimas Altruis: Jurnal Pengabdian Kepada Masyarakat, 6*(1), 37-43. https://doi.org/10.24071/aa.v6i1.5082

Prasasti, W. P. (2020). Tuturan Bahasa Indonesia Masyrakat Etnik Keturunan Arab di Bangil. *JURNAL SATWIKA, 4*(2), 140-149. https://doi/org/10.22219/SATWIKA.Vol4.No2.140-149

Puspita, I. D., Kasih, B. H., & Wiedaningtyas, R. P. (2022). Fenomena Bahasa Jaksel Terhadap Penggunaan Bahasa Indonesia di Kalangan Pengguna Twitter dan Instagram. *Prosiding Seminar Nasional Ilmu Ilmu Sosial (SNIIS), 1*, 663-673.

Razaq, M. T., Nurjanah, D., & Nurrahmi, H. (2023). Analisis Sentimen Review Film Menggunakan Naive Bayes Classifier Dengan Fitur TF-IDF. e-Proceeding of Engineering, 10(2), 1698-1712. Retrieved from https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19997/

Rismayani, W. S., Sihotang, J. I., Aisa, S., Gunawan, H., Tamsir, N., Masturoh, S., Radiyah, U., Gustiana, Z., Harlina, S., Muslihi, M. T. (2023). Data Warehouse dan Data Mining. Yayasan Kita Menulis.

Rofiqi, L., & Akbar, M. (2024). Analisis Sentimen Terkait RUU Perampasan Aset dengan Support Vector Machine. *JEKIN-Jurnal Teknik Informatika, 4*(3), 529-538. https://doi.org/10.58794/jekin.v4i3.824

Rohman, A. (2021). *Prediksi Penyakit Jantung Menggunakan Algoritma C4.5 Berbasis Adaboost*. Penerbit Lakeisha.

Sari, S. N., Faisal, M. R., Kartini, D., Budiman, I., Saragih, T. H., & Muliadi, M. (2023). Perbandingan Ekstraksi Fitur dengan Pembobotan Supervised dan Unsupervised pada Algoritma Random Forest untuk Pemantauan Laporan Penderita COVID-19 di Twitter. *Jurnal Komputasi, 11*(1), 34-42. http://dx.doi.org/10.23960%2Fkomputasi.v11i1.6650

Sarosa, M., Muna, N., Kusumawardani, M., Suyono, A., & Aziz, Y. M., (2022). *Pemrograman Python dalam Contoh dan Penerapan*. Media Nusa Creative.

Sengar, N., Singh, A., & Yadav, V. (2021). Classification of Documents Using Bidirectional Long Short-Term Memory Recurrent Neural Network. *Advances in Intelligent Systems and Computing, 1325*, 149-156. https://doi.org/10.1007/978-981-33-6912-2_14

Setiana, E., Marwondo, Daanestiara, V. R., & Wiyanudin (2023). Analisis

Sentimen Pelaksanaan Kuliah Online Menggunakan Algoritma Support Vector Machine. *Nuansa Informatika, 17*(2), 66–70. https://doi.org/10.25134/ilkom.v17i2.11

Setiawan, Y. (2023). Fenomena Penggunaan Bahasa Jaksel (Code-switching Language) dalam Komunikasi Interpersonal Siswa di SMA Negeri 11 Medan. *KESKAP: Jurnal Kesejahteraan Sosial, Komunikasi dan Administrasi Publik, 2*(1), 24-34. https://doi.org/10.30596/keskap.v2i1.14483

Statista. (2024, April 29). Leading countries based on number of X (formerly Twitter) users as of April 2024. https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/

Sukaesih, D. P. K. E., Khairasyani, I., Listiani, S., Rachmadani, N. O., Sakiinah, A. N., Hanjani, S. S., Ainni., P. N., & Santoso, G. (2023). Sumpah Pemuda Sebagai Persatuan Bangsa Untuk Membangun Negara Yang Berdikari. *Jurnal Pendidikan Transformatif, 2*(2), 360–370. https://doi.org/10.9000/jpt.v2i2.359

Vindua, R., & Zailani, A. U. (2023). Analisis Sentimen Pemilu Indonesia Tahun 2024 dari Media Sosial Twitter Menggunakan Python. *JURIKOM (Jurnal Riset Komputer), 10(*2), 479-487. https://doi.org/10.30865/jurikom.v10i2.5945

Werdiningsih, I., Nuqoba, B., & Muhammadun (2020). *Data Mining Menggunakan Android, Weka, dan SPSS*. Airlangga University Press.

Wibowo, A., & Syahputra, H. (2022). Sistem Deteksi Konten Negatif pada Teks Website Menggunakan Metode Random Forest. *Journal of Informatics and Data Science, 1*(2). https://doi.org/10.24114/j-ids.v1i2.42737

Wicaksono, B., Nursanti, S., & Utamidewi, W. (2022). Motif dan Makna Penggunaan Bahasa "Jaksel" di Kalangan Mahasiswa Pengguna Bahasa "Jaksel" dalam Kehidupan Sehari-hari. *Jurnal Ilmiah Wahana Pendidikan, 8*(21), 388-396. https://doi.org/10.5281/zenodo.7275347

Zaqi, A. M., Raihan, M., Mahesa, S. F., & Santoso, G. (2023). Dampak Positif Sumpah Pemuda pada Organisasi Besar di Indonesia. *Jurnal Pendidikan Transformatif, 2*(2), 194–202. https://doi.org/10.9000/jpt.v2i2.309