Comparing text classification algorithms with n-grams for mediation prediction

Retzi Y. Lewu*[1], Kusrini[2], Ainul Yaqin[3]
[1,2]Magister of Informatics, Universitas AMIKOM Yogyakarta; Jl. North Ring Road, Sleman Regency, Special Region of Yogyakarta, Indonesia
[3]Computer Science Faculty, Universitas AMIKOM Yogyakarta; Jl. North Ring Road, Sleman Regency, Special Region of Yogyakarta, Indonesia
e-mail: ***[1]**retzi.lewu@students.amikom.ac.id, [2]kusrini@amikom.ac.id, [3]ainulyaqin@amikom.ac.id

Abstrak

Tingkat keberhasilan mediasi perkara perdata di pengadilan negeri dari tahun ke tahun sangat rendah dan menyebabkan penumpukan perkara yang harus ditangani dengan persidangan. Sementara itu, pendaftaran perkara baru dengan klasifikasi perkara serupa terus bermunculan dan wajib dimediasi. Penelitian ini dilakukan dengan memanfaatkan data mediasi perkara terdahulu sebagai dataset untuk memprediksi hasil mediasi perkara baru. Dataset yang berbentuk teks ini melalui tahapan preprocessing kemudian diklasifikasikan menggunakan n-grams. Dalam n-gram, "n" merujuk pada jumlah (number of) elemen yang berurutan yang dianggap sebagai single unit. Klasifikasi menggunakan n-gram hanya menemukan nilai pada unigram (n=1). Model kemudian dibangun menggunakan algoritma machine learning, yang menghasilkan akurasi yang sama sebesar 0.6875 pada Algoritma Naïve Bayes, Logistic Regression dan Support Vector Machine (SVM), sedangkan algoritma Decision tree menghasilkan akurasi paling rendah sebesar 0,375. Rendahnya nilai dikarenakan Decision Tree lebih cenderung overfit untuk digunakan dengan teks berbahasa Indonesia. Pola kalimat formal pada dokumen mediasi berbahasa Indonesia tidak memenuhi unsur – unsur kata majemuk, imbuhan, variasi susunan kata, dan semantik leksikal. Untuk penelitian selanjutnya direkomendasikan penggunaan algoritma klasifikasi lain, pemanfaataannya pada dokumen – dokumen lain seperti putusan pengadilan, penentuan rangking mediator berdasarkan keberhasilan mediasi serta implementasi model pada aplikasi e-mediasi yang terintegrasi dengan sistem informasi manajemen perkara.

Kata kunci*— Algoritma Klasifikasi Teks, N-gram, Prediksi hasil mediasi*

Abstract

The backlog of cases need to be tried since district courts' mediation programs have a relatively poor success rate regarding civil matters. New case registrations requiring mediation and comparable case classifications are still coming in the background. To forecast the outcomes of new case mediations, this study employed previous case mediation data as a dataset. This dataset, which are texts, were preprocessed and classified using n-grams. The "n" in n-grams stands for the number of tokens, characters, or words that come after one another and are regarded as a single unit. The classification using n-grams yields only values in unigrams (n=1). The model then built using machine learning algorithms, they are Naïve Bayes, Logistic Regression, and Support Vector Machine (SVM) algorithms which yielded the same accuracy of

Received June 1st,2012; Revised June 25th, 2012; Accepted July 10th, 2012

0.6875, while the Decision tree technique generated the lowest accuracy of 0.375. The Decision Tree found overfitted when applied to the Indonesian text, compensating for the low value. In mediation documents written in Indonesian, formal sentence patterns do not satisfy the requirements for compound words, affixes, word order variations, and lexical semantics. It is advised to employ different classification algorithms for future study and apply them to other documents like court rulings, rank mediators according to their success rate in mediation, and use models in case management information system-integrated e-mediation programs.

Keywords— *Text Classification Algorithms, N-gram, Mediation result prediction*

1. INTRODUCTION

Mediation has been one of the efforts to resolve civil cases between parties in district courts in Indonesia, which the Indonesian Supreme Court regulates. The newest regulation was the Republic of Indonesia Supreme Court Regulation number 3 of 2022 concerning electronic mediation in court, which stated that mediation can be held both conventionally or electronically; hence, it encourages mediators to mediate parties using teleconference and electronic documents. It explicitly benefits both parties by avoiding quarrels and disputes caused by face-to-face meetings, saving time, and lowering costs. Unfortunately, after a year of implementation, successful mediation is still low. The cases to be processed in trials are increasing and adding a number to the backlog as a court burden.

Culture, the parties' personalities, and even the place where the mediation is held might influence civil case mediation's success [1]. Also, there were efforts to apply data science technologies to resolve disputes [2]. Studies in the legal domain have shown that there is a significant awareness regarding the technology's benefits [3].

Many studies proposed using classification algorithms and the n-gram method (or combined) in different datasets and language corpus. [4] use them to model the Arabic language sentiment analysis using LSTM. [5] use the Javanese dataset to do the classification. [6] modeled the Chinese court record documents using BiLSTM. [7][8] applied LSTM and CNN (and its variants) to analyze legal text, yield generative models of deep learning, and contribute an expected result. Research by [9][10] on European Convention on Human Rights (ECHR) decisions accomplished a model used to predict final decisions based on previous documents. Turkish document classification using Naïve Bayes was proposed and measured according to the basic evaluation criteria of precision, recall, accuracy, and f-measure and achieved a success rate of 92% [11]. Several algorithms applied in UK Legal Judgement Prediction showed that the k-NN and RF algorithms obtained the most consistent results across all feature sets. In contrast, the feature set influences the less consistent performance of the SVM and LR algorithms [12]. [13] compared Naïve Bayes, Decision Tree, Logistic Regression, and SVM to scrap and prepare datasets from the Legal-Crystal website and performed an empirical study on the cases related to laws under the Indian Income Tax Act of 1963, observed that SVM outperforms other models.[14] proved that in the case of the Naive Bayes algorithm, accuracy was again significantly higher than any other machine learning algorithm after applying the preprocessing steps, followed by maximum entropy and support vector machine algorithms

Some studies provided the experiment of choosing how to represent text as characteristics that can be processed by machine learning algorithms—a process known as the n-gram method [15][16]. In [17], N-Gram performs word processing in a document as part of feature generation. The Naïve Bayes Classifier method is used in the document categorization process. The results show that document classification accuracy while using N-Gram is 32.68%, while the accuracy when not using N-Gram is 84.97%. The number of features that emerge from solving the N-Gram that are unique or dominating to another category decreases in the classification results. The precision of the outcomes indicates that the utilization of N-Gram in document classification via the Naïve Bayes Classifier algorithm results in a reduced impact on classification performance.

Meanwhile, [18] presented an innovative classification method that integrated the **N-gram** and CNN technology. Research by [19] applied Bi-LSTM with n-grams to classify the text by first applying the n-gram method to generate structured features. The features are then sent into bidirectional LSTM neural networks to generate reliable predictions in both one-vs.-one and one-vs.- rest manners. [20] the study confirmed that the use of AI-based legal support approaches can increase the efficiency and uniformity of administrative court decisions. The experiment used various algorithms to obtain an F1 score of 0.900 with a Precision of 0.929 and a Recall of 0.873. It concluded that in the majority of cases, the system correctly predicts the final decision given the initial document of the proceeding.

The research on mediation dataset found in [21] on Chinese language court documents to lower the court's burden in settling cases. The study processes mediation text data using the LSTM (Long Short Term Memory) framework. Multiple classifiers anticipate mediation results, including TextCNN and BERT for modeling text data and XGBoost and LightGBM for modeling numerical data. A tool named LSTMEnsembler is used in this process. According to the research's experimental findings, the LSTM Ensembler generates an F score of 85.6%.

To the best of our knowledge, no research has examined civil case mediation data from Indonesian courts, which is presented in the form of unstructured text data sets. The goal of this research is to improve knowledge and use of machine learning and text analysis in the prediction of mediation outcomes. It entails establishing prediction models, investigating machine learning algorithms, and devising creative preprocessing strategies for textual data. Additionally, the research helps practitioners by identifying textual characteristics that are indicative of effective mediation resolutions and by promoting the advancement of mediation procedures. By deepening our understanding and extending the applications of these subjects, it also advances academic knowledge. Aside to social considerations, the research will focus on utilizing this data set as a knowledge base for classifying current mediation demands.

Several stages are then implemented inside the research, starting from data selection, to filter data using mediation results labels. Data preparation is the next stage, determining which labeled data will be used in preprocessing. Pre-processing includes case folding, data cleaning, tokenization, and stopword removal to prepare raw data in a format suitable for analysis, modeling, and machine learning tasks. Text classification is carried out after preprocessing using the n-gram approach. The number of consecutive elements taken sequentially for analysis and treated as one unit is indicated by "N" in n-grams. The results of the n-gram, namely terms, will be used in the next step, weighting using TF-IDF to determine how significant it is in a document to the corpus. The next step involves applying text categorization algorithms to convert the data into a machine-understandable format. Naïve Bayes, Decision Tree, Logistic Regression, and Support Vector Machine (SVM) were chosen to create the model. Several techniques are used to measure the success of research in evaluating models using matrices, namely accuracy, precision, recall, and F1 score which results are then used to compare algorithms. This experiment likely reveals insights into the performance of the algorithms and text analysis techniques for predicting or classifying outcomes related to the mediation process.

2. METHODS

The research will be held in several stages, as seen in Figure 1. It starts with problem identification to define the research problem and formulate specific research questions related to text analysis or classification. Problem identification also gives a clear explanation of the goals and expected outcomes of the research. A literature review conducted thorough research on the literature to learn about current approaches, strategies, and best practices in text analysis, machine learning, and natural language processing by finding relevant research papers, methodologies, and algorithms from related fields of study. It also examines the advantages and disadvantages of the current approaches and strategies so that novelty can be established. It is critical so that the significance and relevance of the work can be justified. Additionally, it guarantees that research papers benefit society and the academic community by making significant and lasting contributions. The next step is the data collection of a suitable dataset relevant to the research objectives and research question, that is, previous mediation documents. The data collected is selected by considering whether it meets the criteria or whether the variables are complete or not. If these requirements are met, the data is preprocessed using case folding, cleaning, tokenization, filtering, or stop word removal. The next stop is text classification using n-grams. The letter "n" in an n-gram is the number of tokens, characters, or words that appear after each other and are considered as one unit. Terms, which are the result of n-grams implementation, are used in TF-IDF weighting. It is applied to weigh the importance of those terms within a document relative to their frequency in the entire data collection. The output of TF-IDF is a vector representation of the text data, or weighted matrix, in which each term is given a numerical value that indicates its significance or importance within each document and throughout the collection of documents. The predictive model is then built using this weighted matrix as input features in text classification algorithms; they are Naïve Bayes, Logistic Regression, and Support Vector Machine (SVM). The comparison between algorithms is made regarding their performance evaluation using accuracy, precision, recall, and F1-score metrics. It leads to the conclusion of the research, in which recommendations are made for further research.

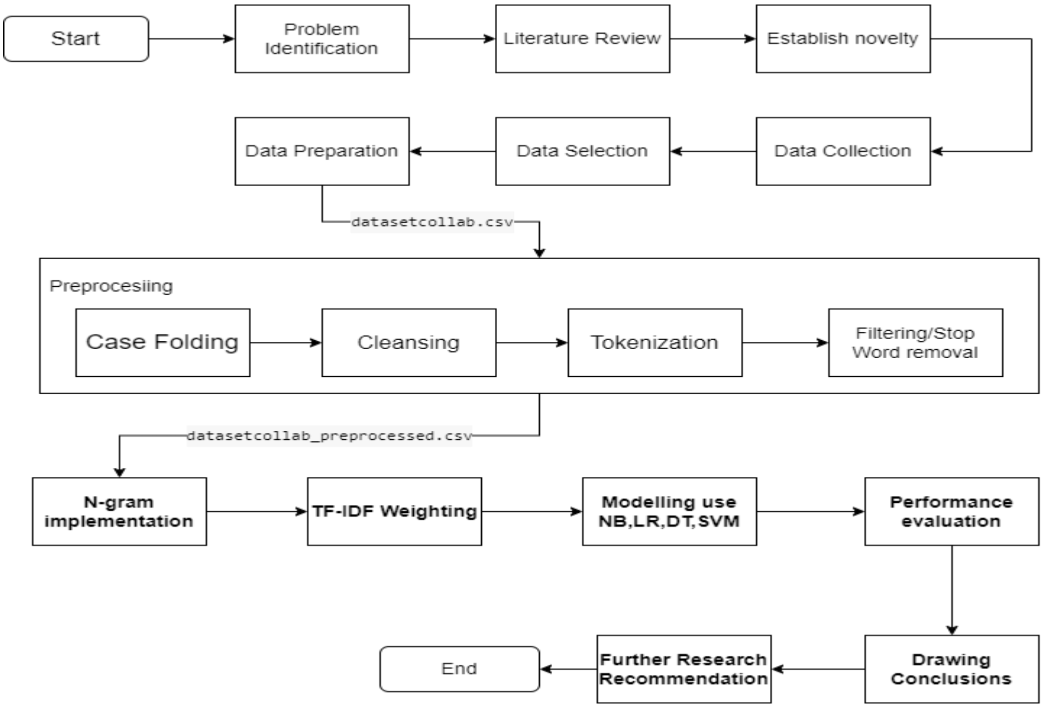

Figure 1 Research Flow Diagram

2.1 Research Approach

This experimental research will descriptively explain the object and the model produced from the dataset so the degree of correctness and potential errors that may occur are known. It will be conducted methodically with a quantitative approach. In the last part of the Result and Discussion, the model's performance will be evaluated for accuracy, precision, recall, and F1-score.

2.2 Data Collection

The method we use in data collection is observation, in which data was collected from the previous results of mediation of civil cases at the District Court within the jurisdiction of the North Sulawesi High Court. They ranged from 2020 to 2023, categorized as success or failure.

2.3 Data Selection and Preparation

From the collected documents, as many as 160 documents are selected from documents for this research purpose, consisting of 80 data from the success category (Y) and another 80 of the failure mediation result (T). The dataset is in Indonesian and will not be translated since we intend to study the use of algorithms and n-grams in the Indonesian language corpus. The variables that will be used in the research are ***Mediation ID***, which is different from each other; ***case classification*** as the type of case; ***information on the start and end times of the mediation***; ***posita / petitum*** is the contents of the plaintiff's lawsuit against the defendant, and ***the status of the mediation result*** (T or Y), as described in Table 1. The selected data is divided into training and test data, with a ratio of 70:30.

Table 1 Dataset example

Mediation id	Case Classification	Mediation Start	Mediation End	Petitum	Mediation result status
941	Perbuatan Melawan Hukum	2020-01-28	2020-02-11	DALAM POKOK PERKARA: 1.Mengabulkan Gugatan Penggugat untuk seluruhnya. 2.Menyatakan sah dan berharga sita jaminan diletakan pengadilan dalam perkara ini 3.Menyatakan menurut hukum bahwa Perjanjian Pembiayaan Multiguna Dan Pemberian Jaminan Secara Kepercayaan (Fiducia) No.117000213861 tanggal 20 Oktober 2017 antara Penggugat dengan tergugat adalah cacat hukum dan batal demi hukum…	T
983	Perbuatan Melawan Hukum	2020-04-09	2020-04-23	Mengabulkan Gugatan Penggugat untuk seluruhnya. Menyatakan perbuatan Tergugat yang tidak mau mentaati hasil kesepakatan bersama antara Penggugat dengan Tergugat sebagaimana yang tercatat didalam Akta Pemisahan dan Pembagian Harta Campur Selama Perkawinan tertanggal 13 Februari Tahun 2020 Nomor. 39 adalah perbuatan melawan Hukum….Menghukum Tergugat untuk membayar biaya Perkara Mohon Keadilan	Y

2.4 Data Pre-processing.

The steps during pre-processing are ***case folding*** to change uppercase to lowercase, ***Clean***ing to remove numbers, punctuation marks, and symbols in the dataset, and *tokenization* conducted on the text of the mediation documents gathered during the data collection phase. Sentences will be used as the unit of analysis for the document's text. The next step is ***Filtering*** or ***Stop Word Removal***, executed using the tokens produced by the earlier procedure. Stop words like “yang”, ” di”, ”dari", and "oleh” will be eliminated during this process. *Stemming* is done by cutting off the end of a word or affix to reach the root of the word.

2.5 The n-gram.

Text Classification use n-gram. The “n” in n-grams stands for the number of tokens, characters, or words that come after one another and are regarded as a single unit. The frequency of stemmed words in the document will be counted as (n). Python will be used in n-gram computation, with n = 1 (unigram) to n = 3 (trigram). ***Terms*** are the result of the n-gram.

2.6 The TF – IDF (Term Frequency – Inverse Document Frequency).

The term as the output of n-gram becoming **t** in equation 1:

$$TF * IDF\ (d,t) = TF(d,t)*log\frac{N}{df(t)} \quad (1)$$

2.7 Modeling with Algorithms

The model will then be constructed using Naïve Bayes, Logistic regression, Decision trees, and SVM (Support Vector machine). These models are used to determine the frequency of root words in texts grouped during the preprocessing phase and then propose whether the mediation succeeded. Last, models will be evaluated regarding the accuracy, precision, recall, and F1-score. It is to compare the performance between algorithms and choose the best combination of algorithms and n-gram.

The Naïve Bayes method is employed to forecast the likelihood of a class membership[22] [23]. It uses the following formula:

$$P(y|X) = P(X|y) * \frac{P(y)}{P(x)} \quad (2)$$

Where P(y | X) is the posterior probability of class y given predictor X, P(X | y) is the likelihood of predictor X given class y, P(y) is the prior probability of class y and P(X) is the marginal probability of predictor X.

The Logistic Regression model predicts the probability of the target variable being in a particular class. The logistic function (sigmoid function) is used to model the probability as follows:

$$P(y = 1 \mid X) = 1 / (1 + \exp(-z)) \quad (3)$$

Where P(y=1 | X) is the probability of the target variable being 1 given predictor X, z is the linear combination of the predictors and their coefficients: z = w0 + w1*X1 + w2*X2 + ... + wn*Xn and exp() is the exponential function.

The next step is to implement the Decision tree. Recursively dividing the data into subsets according to the most essential feature at each stage is how decision trees work. Every leaf node represents a class label, every branch indicates the test's result, and every internal node represents a "test" on an attribute. The decision tree algorithm at each split seeks to decrease impurities or maximize information acquisition.

The SVM algorithm aims to find the hyperplane that best separates the classes in the feature space. The decision function for predicting the class label of a new data point

$$f(x) = sign(w \cdot x + b) \quad (4)$$

Where x is the input data point, w is the weight vector, b is the bias term, · denotes the dot product, and sign() returns the sign of the function value.

2.8 Performance Evaluation

As the goals is to compare between algorithms, some parameters are used to evaluate trained model performance, such as accuracy, precision, recall and F1-score [24]. The percentage of cases correctly classified by the model is measured by accuracy, a well-known set-based scoring metric for assessing the effectiveness of classification algorithms. In this case, accuracy can be used to measure the percentage of slogans that are successfully recognized by a technique [25]. The percentage of cases for which the assigned label is accurate is known as precision. The percentage of instances with a specific label that are successfully identified is known as recall. The harmonic mean of recall and precision is how one might characterize the F1-score [10]. The formula used to calculate those metrices are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (5)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (6)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (7)$$

$$\text{F1-score} = \frac{2 * \text{presisi} * \text{recall}}{\text{presisi} + \text{recall}} \tag{8}$$

3. RESULTS AND DISCUSSION

This research is conducted using Python in several stages :

A. Data Preprocessing

The dataset was cleaned and preprocessed, with cases folded for both ‘case_classification’ and ‘petitum’ columns. The process continued to remove HTML tags as the only cleansing stage in the 'petitum’ column. Punctuation is left in place so that the literal meaning of the petitum is not diminished. The next step included splitting the text into distinct tokens, or tokenization. The result is written in the 'petitum_tokens' column. Filtering or Stop word removal in the 'petitum' column is the next step, using the Sastrawi library. It was selected because it provides a pre-compiled list of stopwords in Indonesian that have been specially selected for the language. The result is stored in a new CSV file named *datasetcollab_preprocessed.csv*.

B. N-grams Implementation

To understand the text's context and word relationships, we looked at n-grams. The tokenized text of the 'petitum’ column in *datasetcollab_preprocessed.csv* was generated into N-grams to extract significant features. From the range of n = 1 to 5, the results are as follows:

```
Most common 1-grams:
'--', : 959
',', : 872
'tergugat', : 648
'dan', : 645
'yang', : 571
';', : 496
'.', : 460
'penggugat', : 380
'(', : 355
')', : 345

Most common 2-grams:
'--', '--', : 922
';', 'menyatakan', : 174
'dan', 'tergugat', : 151
'penggugat', 'dan', : 109
'tergugat', 'i', : 108
'perkara', 'ini', : 107
'objek', 'sengketa', : 99
'menurut', 'hukum', : 97
'turut', 'tergugat', : 93
'menghukum', 'tergugat', : 91

Most common 3-grams:
'--', '--', '--', : 885
'penggugat', 'dan', 'tergugat', : 94
'penggugat', 'untuk', 'seluruhnya', : 69
'menyatakan', 'menurut', 'hukum', : 69
'gugatan', 'penggugat', 'untuk', : 65
'untuk', 'seluruhnya', ';', : 48
'dalam', 'perkara', 'ini', : 45
```

```
'seluruhnya', ';', 'menyatakan', : 44
'sertifikat', 'hak', 'milik', : 44
';', 'menghukum', 'tergugat', : 43

Most common 4-grams:
'--', '--', '--', '--', : 848
'gugatan', 'penggugat', 'untuk', 'seluruhnya', : 63
'penggugat', 'untuk', 'seluruhnya', ';', : 47
'untuk', 'seluruhnya', ';', 'menyatakan', : 42
'antara', 'penggugat', 'dan', 'tergugat', : 41
';', 'menyatakan', 'menurut', 'hukum', : 40
'menyatakan', 'menurut', 'hukum', 'bahwa', : 36
'tanah', 'milik', 'yayasan', 'masarang', : 34
'penggugat', 'dan', 'tergugat', 'yang', : 33
['mengabulkan', 'gugatan', 'penggugat', 'untuk', : 32

Most common 5-grams:
'--', '--', '--', '--', '--', : 811
'gugatan', 'penggugat', 'untuk', 'seluruhnya', ';', : 43
'penggugat', 'untuk', 'seluruhnya', ';', 'menyatakan', : 41
['mengabulkan', 'gugatan', 'penggugat', 'untuk',
'seluruhnya', : 31
';', '--', '--', '--', '--', : 31
'mengabulkan', 'gugatan', 'penggugat', 'untuk', 'seluruhnya',
: 29
'tergugat', 'i', 'dan', 'tergugat', 'ii', : 24
'(', 'sekarang', ':', 'tanah', 'milik', : 24
'sekarang', ':', 'tanah', 'milik', 'yayasan', : 24
':', 'tanah', 'milik', 'yayasan', 'masarang', : 24
```

The bar plotting of n-grams implementation displaying 20 most frequent n-grams is shown in Figure 2:

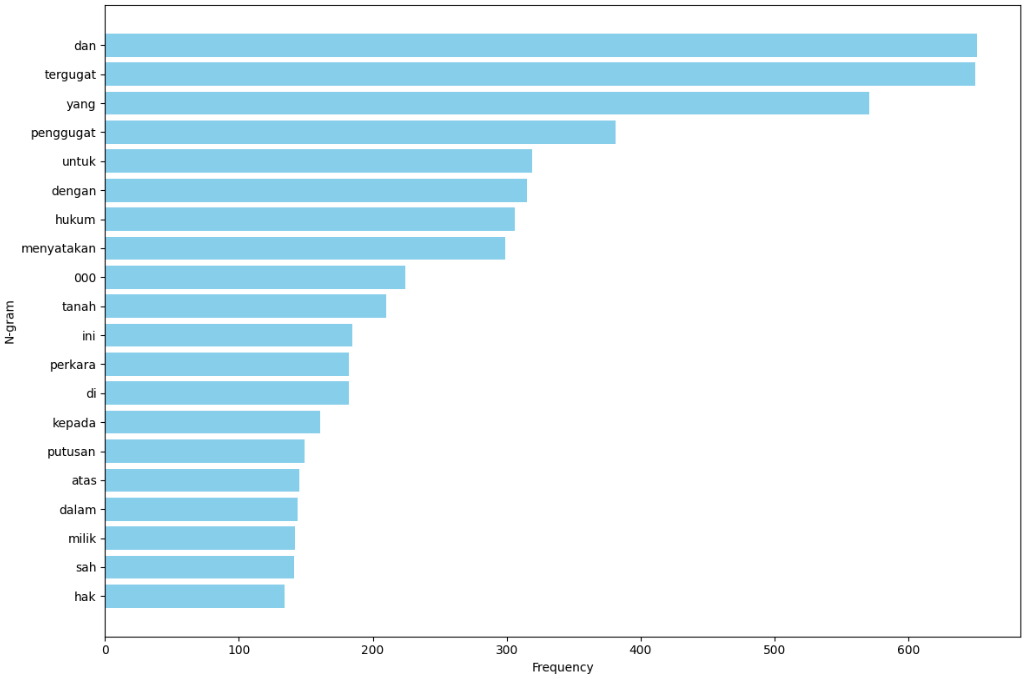

Figure 2 The most 20 frequent n-grams

C. TF – IDF (Term Frequency-Inverse Document Frequency)

To convert the text data into numerical features, TF-IDF (Term Frequency-Inverse Document Frequency) is used, which considers both the frequency of phrases and their significance throughout documents. To calculate the TF-IDF (Term Frequency-Inverse Document Frequency) for the obtained n-grams, the TF (Term Frequency) and IDF (Inverse Document Frequency) components are computed separately and then multiplied together. The implementation produced large numbers of terms that are highly similar to one another.

D. The implementation of algorithms

Various algorithms are used to experiment, in which the TF-IDF features are used to train and assess each algorithm to predict the result variable, or "mediation result status."

a. Naïve Bayes

By Naïve Bayes, the 'petitum' column is loaded as features and the 'status hasil mediasi' column as labels after reading the preprocessed CSV file. Then, the TF-IDF scores are counted for the documents and initialized in the TfidfVectorizer. Train_test_split is used to divide the data into training and testing sets. With the TF-IDF data, the MultinomialNB classifier from scikit-learn is used to model using the Naive Bayes technique. Using the learned classifier, it makes label predictions for the test set. It determines the classifier's accuracy and generates a classification report including the F1-score, support, precision, and recall for every class. The results are as follows:

```
Accuracy: 0.6875
Precision: 0.6875
Recall: 1.0
F1-score: 0.8148148148148148
```

b. Logistic Regression

LogisticRegression is imported from scikit-learn. The Logistic Regression classifier is initialized and trained using the TF-IDF features. Accuracy is calculated by predicting labels for the test set. Last but not least, a classification report is created to assess the Logistic Regression model's performance. The results are as follows:

```
Accuracy: 0.6875
Precision: 0.6875
Recall: 1.0
F1-score: 0.8148148148148148
```

c. Decision tree

DecisionTreeClassifier is imported from scikit-learn. The Decision Tree classifier is initialized and trained using the TF-IDF characteristics. For the test set, we forecast labels and compute accuracy. The Decision Tree model's performance is assessed by printing a classification report as follows:

```
Accuracy: 0.4375
Precision: 0.625
Recall: 0.45454545454545453
F1-score: 0.5263157894736842
```

d. Support Vector Machine (SVM)

SVC is imported using scikit-learn. By utilizing the TF-IDF characteristics, we initialize and train the SVM classifier. Accuracy is calculated by predicting labels for the test set. The SVM model's performance is assessed by printing a classification report as follows:

```
Accuracy: 0.6875
Precision: 0.6875
Recall: 1.0
```

```
F1-score: 0.8148148148148148
```

E. Comparison between Algorithms

The comparison between algorithms can be shown in Table 2 below:

Table 2 Comparison of Accuracy between Algorithms

Algorithm	Accuracy	Precision	Recall	F1 Score
Naïve Bayes (NB)	0.6875	0.6875	1.000000	0.814815
Logistic Regression (LR)	0.6875	0.6875	1.000000	0.814815
Decision tree (DT)	0.4375	0.6250	0.454545	0.526316
Support Vector Machine (SVM)	0.6875	0.6875	1.000000	0.814815

The accuracy of the NB, LR, and SVM achieved the same accuracy of 0.6875, indicating that they correctly classified approximately 68.75% of the instances in the test set. In contrast, the Decision Tree achieved a lower accuracy of 0.4375, indicating that it correctly classified only 43.75% of the instances. The same precision of 0.6875 also occurred in those three, indicating that when they predicted a positive outcome, they were correct approximately 68.75% of the time. At the same time, Decision Tree achieved a precision of 0.6250, which means it had a slightly lower proportion of accurate optimistic predictions compared to the other algorithms. In Recall, the three achieved perfect recall (1.0), indicating that they correctly identified all instances of the positive class in the test set. In contrast, Decision Tree achieved a lower recall of 0.454545, indicating that it missed identifying some instances of the positive class. Three algorithms also achieved the same values for the F1-score of 0.814815, which is the harmonic mean of precision and recall. It balances both precision and recall. Decision Tree achieved a lower F1-score of 0.526316, indicating that it has a lower harmonic mean of precision and recall compared to the other algorithms.

4. CONCLUSIONS

To sum up, our thorough investigation of text classification tasks—which centered on estimating the "mediation result status " in text data written in the Indonesian language—highlights the need to utilize n-grams in conjunction with machine learning algorithms and domain-specific factors. N-grams are a powerful tool for identifying semantic linkages in text data, enabling precise predictions of " mediation status results." However, careful selection of 'n' and consideration of dataset size, computational limitations, and Indonesian language phrases can pose challenges.

The distinctive features of Indonesian language text data, such as compound words, affixes, word order variations, and lexical semantics, influence the efficacy of n-grams and machine learning algorithms in classification tasks. To get meaningful and accurate predictions of the "mediation result status," modeling methodologies must be adjusted to consider the linguistic complexity and cultural context of text data in Indonesian. Integration of N-grams in Prediction: Tokenization, n-gram generation, feature extraction with TF-IDF, model training, assessment, and the prediction of the "mediation result status" for newly uncovered text data are all steps in the integration of N-grams into prediction. N-grams improve the predictive performance and reliability of classification models by capturing the contextual subtleties and semantic linkages present in the text data. The model allows for more precise predictions of the "mediation result status." The analysis highlights the need for a comprehensive approach combining n-grams, machine learning algorithms, and domain-specific factors to predict Indonesian language text status effectively. When n-gram is used in Naive Bayes, Logistic

Regression, Decision Trees, and SVM, they demonstrated comparable performance in text classification tasks, with the Decision tree being the lowest. It indicated potential limitations in capturing complex patterns in text data. It is proved that the use of n-grams combined with TF-IDF will result in high accuracy for those three. Thus, the opinion of [17] can not be validated. The TF-IDF played a crucial role as it led to better discrimination between relevant and irrelevant terms.

With regard to practical implications, comprehensive insights into model performance are offered through accuracy, precision, recall, and F1 scores, enabling detailed knowledge of categorization effectiveness. The differences between various performance indicators are revealed through the interpretation of evaluation metrics, highlighting the importance of considering multiple metrics when assessing models. In other words, models created with Naïve Bayes, Logistic regression, and SVM are more likely to be applied in making mediation predictions. This will result in mediation efficiency because it saves time.

To further improve text categorization performance, future research may examine deep learning architectures, ensemble approaches, and sophisticated modeling techniques. Examining modifications relevant to a given domain and adding contextual data may enhance the generalization and robustness of the model across many domains and languages. Another suggested improvement is employing different classification algorithms and applying them to other documents like court rulings, ranking mediators according to their success rate in mediation, and using models in an integrated case management information system, specifically e-mediation.

REFERENCES

[1] P. Lumbantoruan, R. Mawuntu, C. J. J. Waha, and C. Tangkere, "E-Mediation in E-Litigation Stages in Court," *J. Law, Policy Organ.*, vol. 108, p. 66, 2021, doi: 10.7176/JLPG/108-0.

[2] M. C. Cohen, S. Dahan, C. Rule, and L. K. Branting, "Conflict Analytics: When Data Science Meets Dispute Resolution," *Manag Bus. Rev 2.2*, pp. 86–93, 2022.

[3] O. A. Alcántara Francia, M. Nunez-del-Prado, and H. Alatrista-Salas, "Survey of Text Mining Techniques Applied to Judicial Decisions Prediction," *Appl. Sci.*, vol. 12, no. 20, 2022, doi: 10.3390/app122010200.

[4] A. Setyanto *et al.*, "Arabic Language Opinion Mining Based on Long Short-Term Memory (LSTM)," *Appl. Sci.*, vol. 12, no. 9, 2022, doi: 10.3390/app12094140.

[5] A. P. Ardhana, D. E. Cahyani, and Winarno, "Classification of Javanese Language Level on Articles Using Multinomial Naive Bayes and N-Gram Methods," *J. Phys. Conf. Ser.*, vol. 1306, no. 1, 2019, doi: 10.1088/1742-6596/1306/1/012049.

[6] D. Ji, P. Tao, H. Fei, and Y. Ren, "An end-to-end joint model for evidence information extraction from court record document," *Inf. Process. Manag.*, vol. 57, no. 6, p. 102305, 2020, doi: 10.1016/j.ipm.2020.102305.

[7] N. Bansal, A. Sharma, and R. K. Singh, "A Review on the Application of Deep Learning in Legal Domain," in *IFIP Advances in Information and Communication Technology*, 2019, vol. 559, pp. 374–381. doi: 10.1007/978-3-030-19823-7_31.

[8] D. Alghazzawi, O. Bamasag, A. Albeshri, I. Sana, and H. Ullah, "Efficient Prediction of Court Judgments Using an LSTM + CNN Neural Network Model with an Optimal Feature Set," *Math. - MDPI*, vol. 10, no. 5, p. 683, 2022, doi: https://doi.org/10.2290/math10050683.

[9] C. O. Sullivan and J. Beel, "Predicting the Outcome of Judicial Decisions made by the European Court of Human Rights," *27th AIAI Irish Conf. Artif. Intell. Cogn. Sci.*, 2019, doi: https://doi.org/10.48550/arXiv.1912.10819.

[10] M. Medvedeva, M. Vols, and M. Wieling, "Using machine learning to predict decisions of the European Court of Human Rights," *Artif. Intell. Law*, vol. 28, pp. 237–266, 2020, doi: https://doi.org/10.1007/s10506-019-09255-y.

[11] M. Baygin, "Classification of Text Documents based on Naive Bayes using N-Gram

Features," in *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*, 2019. doi: 10.1109/IDAP.2018.8620853.

[12] B. Strickson and B. De La Iglesia, "Legal Judgement Prediction for UK Courts," in *ACM International Conference Proceeding Series*, Mar. 2020, pp. 204–209. doi: 10.1145/3388176.3388183.

[13] S. Sengupta and V. Dave, "Predicting applicable law sections from judicial case reports using legislative text analysis with machine learning," *J. Comput. Soc. Sci.*, vol. 5, no. 1, pp. 503–516, 2022, doi: 10.1007/s42001-021-00135-7.

[14] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," *Comput. Math. Organ. Theory*, vol. 25, no. 3, pp. 319–335, 2019, doi: 10.1007/s10588-018-9266-8.

[15] T. Georgieva-Trifonova and M. Duraku, "Research on N-grams feature selection methods for text classification," in *IOP Conference Series: Materials Science and Engineering*, Feb. 2021, vol. 1031, no. 1. doi: 10.1088/1757-899X/1031/1/012048.

[16] J. Kruczek, P. Kruczek, and M. Kuta, "Are n-gram categories helpful in text classification?," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 12138 LNCS, pp. 524–537. doi: 10.1007/978-3-030-50417-5_39.

[17] F. Khoirunnisa, N. Yusliani, D. Rodiah, R. Bachelor, and O. Ilir, "Effect of N-Gram on Document Classification on the Naïve Bayes Classifier Algorithm," 2020. doi: https://doi.org/10.36706/sjia.v1i1.13.

[18] W. Haitao, H. Jie, Z. Xiaohong, and L. Shufen, "A Short Text Classification Method Based on N-Gram and CNN.pdf." Wiley Online Library, pp. 248–254, 2020. doi: https://doi.org/10.1049/cje.2020.01.001.

[19] Y. Zhang and Z. Rao, "N-BiLSTM: BiLSTM with n-gram Features for Text Classification," *Proc. 2020 IEEE 5th Inf. Technol. Mechatronics Eng. Conf. ITOEC 2020*, no. Itoec, pp. 1056–1059, 2020, doi: 10.1109/ITOEC49072.2020.9141692.

[20] H. Mentzingen, N. Antonio, and V. Lobo, "Joining metadata and textual features to advise administrative courts decisions: a cascading classifier approach," *Artif. Intell. Law*, no. 0123456789, 2023, doi: 10.1007/s10506-023-09348-9.

[21] H. Hsieh, J. Jiang, T.-H. Yang, R. Hu, and C.-L. Wu, "Predicting the Success of Mediation Requests Using Case Properties and Textual Information for Reducing the Burden on the Court," *Digit. Gov. Res. Pract.*, vol. 2, no. 4, pp. 1–18, 2022, doi: 10.1145/3469233.

[22] D. T. Larose, *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons., 2005.

[23] Kusrini and E. T. Luthfi, *Algoritma Data Mining*, I. Yogyakarta: ANDI OFFSET YOGYAKARTA, 2009.

[24] A. Fandango, I. Idris, and A. Navlani, *Python Data Analysis - Third Edition*. Packt Publishing, 2021.

[25] A. Mandal, K. Ghosh, S. Ghosh, and S. Mandal, "A sequence labeling model for catchphrase identification from legal case documents," *Artif. Intell. Law*, vol. 30, no. 3, pp. 325–358, Sep. 2022, doi: 10.1007/s10506-021-09296-2.