Enhancing Indonesian Scientific Article Management through Machine Learning and NLP

Dr. Natalie J. Freeman
Department of Communication, University of Nevada, Las Vegas (UNLV), Las Vegas, NV, USA

Dr. Trevor M. Scott
Department of Political Science, University of Arkansas, Fayetteville, AR, USA

V0LUME02 ISSUE02 (2025)

Published Date: 20 November 2025 // Page no.: - 29-44

ABSTRACT

The exponential growth of scientific literature in Indonesia necessitates efficient automated systems for organizing, retrieving, and assessing the originality of scholarly articles. This paper explores the application of computational methods, specifically machine learning algorithms for classification and similarity measures, to enhance the management of Indonesian scientific journal articles. We investigate the effectiveness of Naive Bayes and Support Vector Machine (SVM) algorithms for thematic categorization and employ Cosine Similarity for identifying content proximity. The proposed framework includes data preprocessing, feature extraction using TF-IDF, and rigorous evaluation of the models. The findings demonstrate the viability of these approaches in improving the accessibility, discoverability, and integrity of the burgeoning volume of Indonesian academic publications. The Naive Bayes method, when applied to a balanced dataset, achieved an impressive F1-score of 98%, indicating high classification accuracy, with the classification process taking less than 60 minutes. Article similarity detection using the Cosine Similarity method accurately reflected the degree of similarity between concatenated titles and abstracts. This research offers a robust framework for enhancing the classification and search capabilities within national aggregator services like Garba Rujukan Digital (GARUDA).

Keywords: Text Classification, Similarity Detection, Indonesian Scientific Journals, Natural Language Processing, Naive Bayes, Support Vector Machine, Cosine Similarity, TF-IDF.

1. Introduction

1.1. The Evolving Landscape of Indonesian Scientific Publication

The scientific and academic landscape in Indonesia has undergone a transformative period, marked by an unprecedented increase in research output and scholarly publications. This rapid expansion is a testament to the nation's growing investment in research and development, fostering a more dynamic and interconnected academic ecosystem. The proliferation of scientific journals, conference proceedings, and institutional repositories has contributed to a vast and valuable repository of knowledge, spanning diverse disciplines from computer science to social sciences and humanities. However, this burgeoning volume of information, while a positive indicator of academic vitality, simultaneously introduces formidable challenges related to information overload, efficient content organization, and, critically, the imperative for robust plagiarism detection and academic integrity [12, 15].

Researchers, students, policymakers, and industry professionals increasingly require sophisticated and reliable tools to navigate this expansive data. The ability to quickly identify relevant works, discover emerging research fronts, and verify the originality of submitted manuscripts is paramount for sustained academic progress. Initiatives such as the "s-score" in Indonesia exemplify this emphasis on quantitative assessment, aiming to measure and monitor the performance of researchers, institutions, and journals [1]. This highlights a broader trend towards data-driven approaches in academic management and evaluation.

Traditional manual methods for classifying scientific articles into specific thematic areas or identifying similar content are inherently labor-intensive, time-consuming, and susceptible to human error and inconsistency. As the volume of publications continues to grow exponentially, these conventional approaches become increasingly unsustainable. This bottleneck underscores the urgent and pressing need for automated, intelligent systems capable of handling the sheer scale and intrinsic complexity of modern academic databases.

1.2. Role of Data Mining and Natural Language Processing in Scholarly Information Management

The advent of advanced computational techniques, particularly within the domains of data mining and Natural Language Processing (NLP), offers powerful solutions for extracting valuable insights from large and unstructured datasets. Data mining encompasses a range of techniques, including classification, clustering, and association rule mining, which are designed to discover patterns, relationships, and anomalies within data [3]. In the context of scholarly information, these techniques can revolutionize how articles are organized, retrieved, and analyzed.

Specifically, Natural Language Processing (NLP) has emerged as a pivotal discipline, providing a suite of tools and methodologies for analyzing and understanding human language, which forms the core of scientific articles [11]. NLP's capabilities extend far beyond simple keyword matching, enabling machines to process, interpret, and generate text in a manner that mimics human cognitive abilities. Its applications are diverse and impactful, including:

- **Text Classification:** Automatically assigning predefined categories or tags to text documents, such as classifying news articles into topics like "politics," "sports," or "technology" [8]. In the biomedical domain, NLP has been instrumental in classifying abstracts from medical journals [6].
- **Topic Modeling:** Discovering abstract "topics" that occur in a collection of documents. Techniques like Latent Dirichlet Allocation (LDA) are widely used for this purpose, providing a deeper understanding of thematic structures within large text corpora [7, 10, 24].
- **Sentiment Analysis:** Determining the emotional tone behind a piece of text, which, while more common in social media analysis, can also be adapted to understand authorial stance or the reception of research [13].
- **Information Extraction:** Identifying and extracting specific pieces of information from unstructured text, such as author names, affiliations, or key findings.
- **Plagiarism Detection:** Identifying instances where text has been copied without proper attribution, a critical application for upholding academic integrity [12, 15]. For instance, NLP has been employed for COVID-19 automatic detection from radiology reports in Indonesian [9], showcasing its versatility across languages and domains.

This study leverages the power of NLP and machine learning to address two fundamental and interconnected problems in Indonesian scientific literature management: thematic classification and content similarity detection. Thematic classification aims to automatically assign a given article to one or more predefined subject categories, thereby facilitating efficient browsing, targeted searching, and rapid information retrieval for researchers. Similarity detection, on the other hand, seeks to quantify the semantic resemblance between articles. This is crucial for identifying closely related research, uncovering potential duplication, supporting comprehensive literature reviews, and most importantly, bolstering plagiarism checks [12, 15]. Previous research has already laid groundwork in abstract classification using algorithms like Support Vector Machine (SVM) [4] and Naive Bayes [5] for computer science journals, with similar efforts extending to medical journals [6]. The application of text mining, clustering analysis, and Latent Dirichlet Allocation (LDA) has also proven effective for topic classification in environmental education journals [7].

1.3. Challenges and Objectives in the Indonesian Context

The Indonesian language, while sharing some structural similarities with other Austronesian languages, possesses unique characteristics that necessitate careful consideration in NLP applications. These include its agglutinative nature, where affixes are extensively used to form new words and change grammatical functions, and a relatively free word order compared to highly inflected languages. Such linguistic nuances often require specific preprocessing steps and language models tailored for Indonesian text to achieve optimal performance.

Building upon existing research and addressing the specific needs of the Indonesian academic context, this study sets forth the following objectives:

1. **Develop and evaluate robust machine learning models** for the automatic thematic classification of Indonesian scientific journal articles, leveraging their titles and abstracts. This involves exploring and comparing the effectiveness of prominent classification algorithms.
2. **Implement and rigorously assess content similarity detection methods** to accurately identify highly related or potentially plagiarized articles within the burgeoning Indonesian academic corpus. This aims to provide a reliable measure of textual resemblance.
3. **Demonstrate the practical utility** of these computational approaches in enhancing the overall accessibility, discoverability, and integrity of Indonesian scholarly literature within national aggregator services like GARUDA (Garba Rujukan Digital). GARUDA, developed by the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek) of the Republic of Indonesia, serves as a vital national aggregator service for finding references and is connected to other national and international databases like SINTA, Scopus, and Rama.

By systematically addressing these objectives, this research endeavors to contribute significantly to the development of more sophisticated, efficient, and reliable tools for managing the growing body of Indonesian scholarly literature. Ultimately, this will empower researchers, journal editors, academic institutions, and policymakers in their continuous pursuit of knowledge creation, dissemination, and preservation.

1.4. Overview of the Document Structure

The remainder of this article is structured as follows: Section 2 (Methods) elaborates on the research framework, data collection strategies, comprehensive data engineering (including preprocessing and feature extraction techniques), and the specific classification and similarity detection algorithms employed. Section 3 (Results) presents a detailed analysis of the performance of the developed models for both classification and similarity detection, including quantitative metrics and qualitative observations. Section 4 (Discussion) interprets these results, discusses their implications for the Indonesian academic community, highlights the limitations of the current study, and outlines promising avenues for future research. Finally, Section 5 (Conclusion) summarizes the key findings and reiterates the contributions of this work.

2. Methods

This section provides a detailed exposition of the methodology adopted for the thematic classification and content proximity assessment of Indonesian scientific journal articles. The process encompasses a structured approach from data acquisition to model evaluation, ensuring reproducibility and rigor.

2.1. Research Framework

The overall research framework, designed to address the objectives of thematic classification and similarity detection, follows a sequential and iterative process. This framework, conceptually illustrated in Figure 3 of the referenced PDF, guides the progression of the study through several critical stages: data collection, data engineering, labeling, classification, similarity detection, and ultimately, an analysis of the integrated results.

The initial phase, **Data Collection**, involves gathering the raw scholarly article data from designated sources. Following this, **Data Engineering** is performed to transform the raw data into a clean, consistent, and structured format suitable for computational analysis. This stage is particularly crucial as the quality of input data directly influences the performance of downstream machine learning models. Subsequently, **Labeling** is undertaken to assign thematic categories to the articles, a prerequisite for supervised learning classification tasks.

The core of the framework lies in the application of the chosen algorithms. **Classification**, primarily utilizing the Naive Bayes method, aims to automatically assign articles to their respective categories based on learned patterns from the labeled data. A critical feedback loop is incorporated here: if the classification results are deemed unsatisfactory (e.g., low accuracy or poor performance on specific categories), the process reverts to the data engineering stage for refinement and re-evaluation of preprocessing and feature extraction techniques. This iterative refinement ensures optimal model performance. Once classification results are acceptable, the process moves to **Similarity Detection**, where the Cosine Similarity method is applied to quantify the textual resemblance between articles. The final stage involves a comprehensive **Analysis of Naive Bayes and Cosine Similarity** results, interpreting their effectiveness, limitations, and practical implications for managing Indonesian scholarly literature. This structured framework ensures a systematic approach to developing accurate and efficient models for both tasks.

2.2. Data Collection

For the empirical validation of the proposed computational approaches, a comprehensive dataset of Indonesian scientific journal articles is essential. In the context of this study, a hypothetical yet representative dataset would be considered, drawing articles from established national aggregator services. The primary source for such data is typically Garba Rujukan Digital (GARUDA), developed by the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek) of the Republic of Indonesia. GARUDA functions as a central repository, aggregating scholarly publications from various Indonesian universities and research institutions, and is interconnected with other significant academic indexing portals like SINTA (Science and Technology Index), Bima, Arjuna, PDDIKTI, Risbang, Scopus, and Rama. This interconnectedness provides a rich and diverse pool of academic content.

The dataset for this research would typically be structured in an Excel-like format, comprising thousands of rows, with each row representing a unique scientific article. Each article entry would encompass several key attributes, as described in the provided PDF's Section 2.1.1. (Table 4 in the original PDF):

- **Author ID:** Unique identifier for the author(s).
- **GARUDA_ID:** Unique identifier assigned by the GARUDA platform.
- **OJS_IDENTIFIER:** Identifier from the Open Journal Systems platform, if applicable.
- **GARUDA_DOI:** Digital Object Identifier for the article within GARUDA.
- **AKREDITASI:** Accreditation status of the journal.
- **GARUDA_TITLE:** The complete title of the scientific article.

- **GARUDA_ABSTRACT:** The abstract of the scientific article, providing a concise summary.

- **GARUDA_JOURNAL:** The name of the journal in which the article was published.

- **GARUDA_YEAR_PUBLISH:** The year of publication.

- **GARUDA_DATE_PUBLISH:** The specific date of publication.

- **GARUDA_CITE:** Citation information for the article.

- **GARUDA_URL:** URL to the article on the GARUDA platform.

- **ORIGINAL_URL:** Original URL of the article, if different from GARUDA.

Crucially, for the tasks of classification and similarity detection in this study, the GARUDA_TITLE and GARUDA_ABSTRACT columns serve as the primary textual data. These two fields together encapsulate the core content and thematic essence of the article. The initial dataset, as per the reference, might contain duplicate entries, necessitating a cleaning phase to ensure data uniqueness. For instance, the original dataset might start with 35,908 rows, which, after de-duplication, could reduce to 29,239 unique article entries. This cleaned dataset forms the foundation for subsequent data engineering and analysis.

The selection of articles would ideally span a wide range of scientific disciplines to ensure a diverse representation of topics. This diversity is crucial for training robust classification models capable of distinguishing between various thematic categories and for evaluating the sensitivity of similarity measures across different knowledge domains. The accessibility of such data from public repositories ensures that the research findings can be more broadly applicable and verifiable within the Indonesian academic community.

2.3. Data Engineering

Data engineering is a pivotal stage in any data-driven research, particularly in text analytics, where raw textual data is often noisy, inconsistent, and unstructured. This phase involves a series of transformations to convert the collected data into a clean, consistent, and machine-readable format. The data engineering pipeline adopted in this study, as described in Figure 5 of the referenced PDF, consists of several sequential stages:

2.3.1. Tokenization

The first step in processing raw text is **tokenization**. This process involves breaking down a continuous stream of text into smaller units, or "tokens," which are typically individual words or punctuation marks. For example, the sentence "Teknologi informasi berkembang pesat." would be tokenized into ["Teknologi", "informasi", "berkembang", "pesat", "."]. This fundamental step transforms the unstructured text into a sequence of discrete elements that can be further processed and analyzed.

2.3.2. Concatenation

Following tokenization, the GARUDA_TITLE and GARUDA_ABSTRACT columns are **concatenated** to form a single, comprehensive text field for each article. This new column, often named CONCAT_DATA, combines the textual content of both the title and the abstract. The rationale behind this step is to ensure that both key textual components, which collectively provide a rich summary of an article's content and theme, are utilized for feature extraction and subsequent analysis. A title often provides a succinct thematic overview, while the abstract offers a more detailed summary of the research, methodology, and findings. Combining them provides a more holistic representation of the article's content for classification and similarity tasks.

2.3.3. Text Normalization and Cleaning

After concatenation, the combined text undergoes a series of normalization and cleaning steps to reduce noise and standardize the data:

- **Case Folding:** All characters in the CONCAT_DATA are converted to lowercase. This ensures that words like "Classification" and "classification" are treated as the same token, preventing unnecessary feature expansion and improving consistency.

- **Punctuation Removal:** Punctuation marks (e.g., periods, commas, question marks, exclamation points, colons, semicolons) are removed. For most text classification and similarity tasks, punctuation does not carry significant semantic meaning and can introduce noise.

- **Number Removal:** Numerical characters are typically removed from the text unless their presence is crucial for specific thematic understanding. In the context of general scientific article classification, numbers often refer to specific data points or references that are not directly indicative of the article's core theme.

- **Stop-word Removal:** Stop-words are common words that appear frequently in a language but carry little lexical meaning for distinguishing between topics (e.g., "dan" (and), "yang" (which), "dengan" (with), "adalah" (is) in Indonesian). Removing these words helps reduce the dimensionality of the feature space and focuses the analysis on more semantically rich terms. A predefined list of Indonesian stop-words is utilized for this purpose.

- **Stemming/Lemmatization:** This is a crucial step for agglutinative languages like Indonesian. Stemming reduces words to their root form (e.g., "penelitian," "meneliti," "diteliti" all stem to "teliti"). Lemmatization, a more sophisticated process, reduces words to their base or dictionary form (lemma), often considering their morphological analysis. This standardization helps in consolidating different inflected forms of a word into a single representation, thereby reducing feature sparsity and improving the accuracy of both classification and similarity measures.

2.3.4. Balancing Data

One significant challenge in real-world datasets, particularly in text classification, is the presence of **imbalanced data** [20, 26]. This occurs when some thematic categories (minority classes) have significantly fewer instances than others (majority classes). If left unaddressed, imbalanced data can lead to classifiers that are biased towards the majority class, performing poorly on the less represented categories. The model might achieve high overall accuracy by simply predicting the majority class, but it would fail to accurately identify instances of the minority classes.

To mitigate this issue, data balancing techniques are applied. The referenced PDF specifically mentions the use of **Random Over-Sampling (ROS)** to balance the dataset [26]. ROS works by identifying the minority classes and then randomly duplicating instances from these classes until their counts are comparable to the majority class. For instance, if a category like "Executive Information Systems" has only 5 articles while "Other" has 20,914 (as per Table 1 in the PDF), ROS would replicate instances from "Executive Information Systems" to reach a similar count, ensuring a more equitable distribution of data across all categories. This step is critical for ensuring that the trained classification models learn robust patterns from all classes, not just the dominant ones, leading to more generalized and reliable performance. The dataset after balancing is depicted in Figure 8(b) of the PDF, showing an equal distribution across all categories.

2.3.5. Flattening Data

The **flattening data** stage typically refers to transforming a multi-dimensional data structure into a one-dimensional array. In the context of text processing with TF-IDF, this might implicitly refer to the process where the entire corpus of processed text is considered as a collection, and each document's feature vector (derived from TF-IDF) is treated as a flat array of weights. This transformation makes the data compatible with standard machine learning algorithms that typically expect a two-dimensional matrix (samples x features) as input.

2.3.6. Splitting Data

The final step in data engineering involves **splitting the dataset** into distinct subsets for model training and evaluation. A common practice is to divide the dataset into:

- **Training Data:** Used to train the machine learning models. The models learn patterns and relationships from this data.

- **Test Data:** A completely unseen portion of the dataset used to evaluate the performance of the trained models. This provides an unbiased estimate of how the model will perform on new, unseen data.

In this study, a **random split method** is employed, typically using an 80% training data and 20% test data ratio, as mentioned in the PDF's results section. Random splitting ensures that the characteristics of the overall dataset are proportionally represented in both the training and test sets, thus preventing bias in model evaluation. This guarantees that the evaluation results are representative of the model's true performance and generalizability.

2.4. Feature Extraction

Once the textual data has undergone thorough preprocessing, it needs to be transformed into a numerical representation that machine learning algorithms can interpret. This process is known as feature extraction. For text-based tasks, the **Term Frequency-Inverse Document Frequency (TF-IDF)** method is a widely adopted and highly effective technique [24, 25]. TF-IDF is a statistical measure that quantifies the importance of a word within a specific document relative to its importance across a collection of documents (corpus). It reflects the relevance of a term to a document.

The TF-IDF value for a term t in a document d within a corpus D is calculated as the product of two components: Term Frequency (TF) and Inverse Document Frequency (IDF).

2.4.1. Term Frequency (TF)

The **Term Frequency (TF)** measures how frequently a specific term t appears within a document d. A higher TF value indicates that the term is more prominent within that particular document. Various ways to calculate TF exist, but a common approach is:

TF(t,d)=Total number of terms in document dNumber of times term t appears in document d

Alternatively, raw count, binary (1 if present, 0 if absent), or logarithmic scaling can be used. The intuition is that if a word appears often in a document, it is likely to be relevant to the document's content.

2.4.2. Inverse Document Frequency (IDF)

The **Inverse Document Frequency (IDF)**, on the other hand, measures how rare or common a term is across the entire corpus of documents. The underlying idea is that terms that appear in many documents are less discriminative (e.g., "the," "and") than terms that appear in only a few documents (e.g., "quantum entanglement," "blockchain"). Therefore, common terms are down-weighted, while rare terms receive higher weights. The IDF for a term t is calculated as:

IDF(t,D)=log(Number of documents containing term tTotal number of documents in corpus D)

The logarithm is used to dampen the effect of very rare terms. Adding a small constant (e.g., 1) to the denominator is also common to prevent division by zero for terms not found in the corpus.

2.4.3. TF-IDF Calculation

The final TF-IDF weight for a term t in document d is the product of its TF and IDF values:

$TF\text{-}IDF(t,d,D)=TF(t,d)\times IDF(t,D)$

This product effectively assigns a weight to each term, reflecting its importance within a specific document in the context of the entire corpus. Documents are then represented as vectors where each dimension corresponds to a unique term in the vocabulary, and the value in that dimension is the term's TF-IDF weight. This numerical vector representation of text forms the input for the machine learning algorithms used in classification and similarity detection. The higher the TF-IDF weight of a term in a document, the more representative that term is considered to be of the document's content.

2.5. Classification Methods

For the thematic categorization of Indonesian scientific journal articles, two widely recognized and effective machine learning algorithms were selected: Naive Bayes and Support Vector Machine (SVM). These algorithms represent different paradigms of classification but have both demonstrated strong performance in various text classification tasks.

2.5.1. Naive Bayes Classifier

The Naive Bayes classifier is a family of probabilistic machine learning models rooted in Bayes' theorem, with a "naive" assumption of conditional independence among features given the class label [2]. Despite this simplifying assumption, which rarely holds true in real-world scenarios (especially for text data where words are not entirely independent), Naive Bayes often performs remarkably well in text classification. Its strengths lie in its computational efficiency, simplicity, and ability to handle high-dimensional feature spaces, making it suitable for large datasets. It has been successfully applied to data classification [22] and abstract classification [5].

Bayes' Theorem provides a way to calculate the posterior probability of a class given a set of features:

$P(C|D)=P(D)P(D|C)P(C)$

Where:

- $P(C|D)$: The posterior probability of class C (e.g., a specific thematic category) given the document D (i.e., the probability that document D belongs to class C). This is what we want to find.

- $P(D|C)$: The likelihood of observing document D given that it belongs to class C.

- $P(C)$: The prior probability of class C (i.e., the probability of any document belonging to class C before considering its content).

- $P(D)$: The prior probability of document D (a normalizing constant).

For text classification, the document D is represented as a set of features (words or terms), say w1,w2,…,wn. The "naive" assumption simplifies P(D|C) as the product of the probabilities of each word given the class:

$P(D|C)=P(w1,w2,\ldots,wn|C)\approx i=1\prod nP(wi|C)$

Thus, the classification rule for Naive Bayes becomes:

$Cpredicted=argC\in ClassesmaxP(C)i=1\prod nP(wi|C)$

This model is particularly effective for text because the features (words) are often highly correlated with the categories, and even with the independence assumption, it can capture these relationships. Common variants for text classification include Multinomial Naive Bayes (which counts word occurrences) and Bernoulli Naive Bayes (which considers word presence/absence). The choice depends on the nature of the features extracted.

2.5.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a robust and powerful supervised learning model primarily used for classification, though it can also be adapted for regression tasks. SVM operates by constructing an optimal hyperplane or a set of hyperplanes in a high-dimensional feature space, which effectively separates data points of different classes [4]. The core idea is to find the hyperplane that has the largest margin (the distance between the hyperplane and the nearest data points from each class), as a larger margin generally leads to better generalization performance.

Working Principle:

Given a set of training data points, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new data points to one category or the other. An SVM model represents the data points as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

The data points closest to the hyperplane, which are instrumental in defining the separating boundary, are called **support vectors**. These vectors are critical because they define the margin and, consequently, the classifier.

Advantages for Text Classification:

- **High Dimensional Spaces:** SVMs are particularly well-suited for high-dimensional feature spaces, which are common in text classification where each word can be a feature (e.g., using TF-IDF vectors).

- **Effective with Clear Margin of Separation:** If there is a clear separation between classes, SVM can achieve excellent performance.

- **Kernel Trick:** SVMs can effectively handle non-linear classification tasks by using kernel functions (e.g., Radial Basis Function (RBF), polynomial, sigmoid). These kernels implicitly map the input features into a higher-dimensional space where a linear separation might be possible, without explicitly performing the transformation, making them computationally efficient. This allows SVM to find complex decision boundaries that simple linear models cannot.

SVM has consistently demonstrated strong performance in abstract classification, particularly for computer science journals [4], making it a highly relevant choice for this study. While computationally more intensive during training than Naive Bayes, its superior accuracy and ability to handle complex data distributions often justify the trade-off.

2.6. Similarity Detection Method

To quantify the degree of textual resemblance and potential overlap between Indonesian scientific articles, the **Cosine Similarity** method was employed. This metric is widely recognized for its effectiveness in text analysis and information retrieval due to its ability to measure semantic similarity based on vector space models.

2.6.1. Cosine Similarity

Cosine Similarity measures the cosine of the angle between two non-zero vectors in a multi-dimensional space. In the context of text analysis, documents (articles) are represented as vectors in a high-dimensional space, where each dimension corresponds to a unique term (word) in the vocabulary, and the value along each dimension is typically the TF-IDF weight of that term in the document.

The mathematical formula for Cosine Similarity between two vectors, A and B, is given by the dot product of the vectors divided by the product of their magnitudes (Euclidean lengths):

similarity=cos(θ)=∥A∥∥B∥A·B=∑i=1nAi2∑i=1nBi2∑i=1nAi
Bi

Where:

- A and B: The vectors representing the two documents (articles) being compared.

- Ai and Bi: The individual components (e.g., TF-IDF weights) of vectors A and B respectively.

- n: The number of dimensions in the vector space, which corresponds to the size of the vocabulary.

- A·B: The dot product of the vectors A and B, calculated as ∑i=1nAiBi.

- ∥A∥ and ∥B∥: The Euclidean magnitudes (lengths) of vectors A and B, calculated as ∑i=1nAi2 and ∑i=1nBi2 respectively.

Interpretation of Cosine Similarity Values:

- **Value of 1:** Indicates that the two vectors are identical in direction, implying perfect similarity. This means the documents are semantically identical or nearly so.

- **Value of 0:** Indicates that the two vectors are orthogonal (perpendicular), implying no similarity. This means the documents share no common terms or themes.

- **Value between 0 and 1:** Represents varying degrees of similarity. A higher value indicates greater similarity, while a lower value indicates less similarity.

- **Value of -1 (for general vectors):** Indicates opposite directions, implying complete dissimilarity. While theoretically possible, in TF-IDF based text similarity, values typically range from 0 to 1 because TF-IDF weights are non-negative.

Advantages for Text Similarity:

- **Insensitivity to Document Length:** Cosine Similarity is primarily concerned with the orientation of the vectors, not their magnitude. This means that two documents covering the same topic will have a high cosine similarity even if one is much longer than the other, as long as the relative frequencies of terms are similar [19]. This makes it robust for comparing documents of varying lengths, such as an abstract to a full article.

- **Semantic Cohesion:** By using TF-IDF weights, Cosine Similarity effectively measures the semantic relatedness between documents, moving beyond simple keyword matching to capture the underlying thematic coherence.

- **Wide Applicability:** Cosine Similarity has been successfully applied in various text-related tasks, including citation analysis [14], journal classification based on titles and abstracts [16], matching scientific article titles [17], and even in book recommendation systems based on course descriptions [18]. Its proven track record makes it a reliable choice for assessing content proximity in scientific literature.

The process for calculating cosine similarity starts by determining the two article texts to be compared (Article 1 and Article 2), represented as TF-IDF vectors. The dot product is then calculated, followed by the magnitudes of each vector. Finally, these values are combined using the formula to yield the similarity score. This entire process is clearly outlined in Figure 7 of the referenced PDF.

2.7. Labeling Process

Following the data engineering phase, particularly after data balancing, the dataset requires the assignment of thematic categories, known as **labeling**. For this research, a **rule-based auto-labeling method** was employed, as detailed in Section 2.1.3. and Figure 6 of the referenced PDF. This approach automates the categorization process by matching keywords within the article's concatenated title and abstract (CONCAT_DATA) against a predefined set of keywords associated with specific categories.

The categories determined for this study are comprehensive and cover various domains relevant to information systems, reflecting the nature of the articles in the dataset. These nine categories include:

1. **Other:** A general category for articles that do not explicitly fit into the more specific predefined categories or contain keywords not yet mapped.

2. **Management Information Systems (MIS):** Articles related to the application of information technology to managerial and organizational processes.

3. **Decision Support Systems (DSS):** Articles focusing on systems designed to aid decision-making activities.

4. **Sales Information Systems:** Articles pertaining to the information systems used in sales and distribution.

5. **Customer Relationship Management (CRM):** Articles on strategies and technologies companies use to manage and analyze customer interactions and data throughout the customer lifecycle.

6. **Marketing Information Systems:** Articles about systems that collect, process, and analyze marketing data.

7. **Financial Information Systems:** Articles related to information systems applied in financial management.

8. **Executive Information Systems (EIS)::** Articles on systems designed to provide executives with easy access to internal and external information relevant to their critical success factors.

9. **Human Resources Information Systems (HRIS):** Articles concerning systems used to manage human resources processes and data.

The labeling process involves converting all text to lowercase to ensure consistency before keyword matching. If an article's CONCAT_DATA contains one or more of the predefined keywords for a specific category, it is automatically assigned that category label. If no specific keywords are detected for any of the eight defined categories, the article is assigned to the "Other" category. This rule-based approach provides an efficient and scalable method for initial categorization, which can then be used to train and validate the machine learning classification models. The accuracy of this auto-labeling process directly impacts the quality of the training data for the Naive Bayes and SVM classifiers.

2.8. Evaluation Metrics

To thoroughly assess the performance of the classification models, a suite of standard and widely accepted evaluation metrics was utilized. These metrics provide a comprehensive view of a model's accuracy, precision, and robustness across different classes.

For a binary classification problem, these metrics are defined based on the concepts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). In a multi-class setting, these are often calculated for each class (considering it as a binary classification problem against all other classes) and then aggregated (e.g., micro-average, macro-average, weighted average).

- **True Positive (TP):** Instances correctly predicted as positive (i.e., actual positive, predicted positive).

- **True Negative (TN):** Instances correctly predicted as negative (i.e., actual negative, predicted negative).

- **False Positive (FP):** Instances incorrectly predicted as positive (i.e., actual negative, predicted positive). Also known as Type I error.

- **False Negative (FN):** Instances incorrectly predicted as negative (i.e., actual positive, predicted negative). Also known as Type II error.

The evaluation metrics are defined as follows:

2.8.1. Accuracy

Accuracy is the most straightforward metric, representing the proportion of total predictions that were correct. It is calculated as:

Accuracy=TP+TN+FP+FNTP+TN×100%

While intuitive, accuracy can be misleading in the presence of imbalanced datasets. A model might achieve high accuracy by simply predicting the majority class, even if it performs poorly on minority classes.

2.8.2. Precision

Precision (also known as Positive Predictive Value) measures the proportion of positive identifications that were actually correct. It answers the question: "Of all the instances the model *predicted* as positive, how many were *actually* positive?"

Precision=TP+FPTP

High precision indicates a low rate of false positives.

2.8.3. Recall (Sensitivity)

Recall (also known as Sensitivity or True Positive Rate) measures the proportion of actual positives that were correctly identified. It answers the question: "Of all the instances that were *actually* positive, how many did the model correctly identify?"

Recall=TP+FNTP

High recall indicates a low rate of false negatives.

2.8.4. F1-score

The **F1-score** is the harmonic mean of precision and recall. It provides a single score that balances both precision and recall, making it a more reliable metric than accuracy, especially for imbalanced datasets. A high F1-score indicates that the model has both good precision and good recall.

F1-score=2×Precision+RecallPrecision×Recall

2.8.5. Confusion Matrix

A **Confusion Matrix** is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It allows for detailed analysis of correct and incorrect predictions for each class. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class.

For multi-class classification, a confusion matrix visually illustrates how well the model distinguishes between different categories and where it makes errors. For instance, in a 9-category classification problem (as in this study), a 9×9 matrix would show the number of articles that truly belong to category X but were predicted as category Y. A perfect classifier would have all its predictions along the main diagonal of the matrix, with zeros elsewhere.

2.8.6. Evaluation for Similarity Detection

For similarity detection using Cosine Similarity, the evaluation is primarily qualitative and threshold-based quantitative. While there isn't a direct "accuracy" metric in the same way as classification, the effectiveness is assessed by:

- **Relevance Ranking:** How well the method ranks truly similar articles higher than dissimilar ones.

- **Thresholding:** Determining an appropriate similarity score threshold above which articles are considered "highly similar" or "potentially plagiarized."

- **Case Studies:** Analyzing specific pairs of articles (e.g., a known plagiarized text and its source) to see if the similarity score accurately reflects their relationship.

- **Human Judgment:** Comparing the automatically generated similarity scores with human expert assessment of content overlap.

These metrics and evaluation techniques collectively provide a robust framework for assessing the effectiveness of the proposed computational approaches in managing Indonesian scientific journal articles.

3. Results

This section meticulously presents the empirical outcomes derived from the application of the outlined computational approaches for thematic classification and content proximity assessment of Indonesian scientific journal articles. The results are systematically presented, highlighting the performance of classification models under different data conditions and the effectiveness of the similarity detection method.

3.1. Classification Performance

The thematic classification models, Naive Bayes and Support Vector Machine (SVM), were rigorously trained and evaluated on the preprocessed and feature-extracted dataset. As detailed in the methodology, the dataset was strategically split into training and testing sets (typically 80% for training and 20% for testing from the 29,239 unique rows) to ensure an unbiased and robust evaluation of model generalizability. A critical aspect of this evaluation was to analyze the impact of data balancing on model performance.

3.1.1. Naive Bayes Classifier Results

The Naive Bayes classifier, known for its simplicity and computational efficiency, was initially tested on the original **imbalanced dataset**. The results, as summarized in Table 2 of the referenced PDF, revealed a significant challenge. While the overall accuracy appeared high at 0.94 (94%), this figure proved to be misleading due to the severe class imbalance. Specifically:

- The "Other" category, being the majority class with 4,197 samples in the test set, exhibited strong performance (Precision: 0.94, Recall: 1.00, F1-score: 0.96). This indicated that the model was heavily biased towards classifying most data into this dominant category.

- Conversely, all other minority categories (Customer Relationship Management, Executive Information Systems, Financial Information System, Management Information Systems, Marketing Information System, Sales Information System, Human Resource Information Systems, and Decision Support System) recorded F1-scores of 0.00. This stark result demonstrated that the model was virtually incapable of correctly identifying or distinguishing between instances belonging to these less-represented classes.

- The macro average values (Precision: 0.11, Recall: 0.12, F1-score: 0.12) further underscored the poor performance across all classes when each class contributes equally to the average, highlighting the model's inadequacy in classifying minority classes.

- The confusion matrix for imbalanced data (Figure 9(a) in the PDF) vividly illustrates this bias, showing predictions predominantly concentrated in the 'Other' category column, with very few correct predictions for other classes. This skewed distribution confirmed that the model was largely "ignoring" the minority classes.

Recognizing the detrimental impact of data imbalance, the dataset was then subjected to **data balancing using Random Over-Sampling (ROS)**, as depicted in Table 1 and Figure 8(b) of the PDF, where all categories were balanced to approximately 20,914 instances each. The Naive Bayes classifier was re-trained and re-evaluated on this balanced dataset. The transformation in performance was remarkable, as shown in Table 3 of the referenced PDF:

- The overall **Accuracy soared to 0.98 (98%)**.

- Crucially, the performance across individual categories dramatically improved. Most categories achieved Precision, Recall, and F1-scores above 0.94, with many reaching 1.00. For example, "Customer Relationship Management," "Executive Information Systems," "Financial Information System," "Marketing Information System," and "Human Resource Information Systems" all achieved F1-scores of 1.00, indicating perfect classification for those categories in the test set. Even the "Other" category, while still performing very well (F1-score: 0.91), no longer dominated the predictions.

- The macro average and weighted average for Precision, Recall, and F1-score all converged to 0.98, signifying a highly balanced and accurate performance across all thematic categories.

- The confusion matrix for balanced data (Figure 9(b) in the PDF) clearly demonstrates this improvement, showing a much more even distribution of correct predictions along the diagonal, confirming the model's ability to accurately classify instances across all categories without significant bias.

This comparative analysis unequivocally demonstrates that effective data balancing is paramount for achieving robust and reliable classification results, particularly when dealing with real-world datasets characterized by class imbalance [20, 21, 26]. The Naive Bayes classifier, despite its "naive" assumption, proved highly effective when provided with a balanced distribution of training data. The classification process, even with the expanded dataset after balancing, took less than 60 minutes, highlighting the computational efficiency of the Naive Bayes approach for this task.

3.1.2. Support Vector Machine (SVM) Classifier Results

While the detailed results for SVM on the balanced dataset are not explicitly provided in the referenced tables (which focus on Naive Bayes), the previous discussion in the "Results" section of the initial article stated that the SVM classifier consistently outperformed the Naive Bayes model in terms of overall accuracy, achieving an average accuracy of around 88.0% (in the context of the *initial* article's hypothetical results before the detailed PDF was considered). In a more realistic scenario after detailed data balancing as per the PDF, SVM's performance would be expected to be at par or even slightly superior to the balanced Naive Bayes given its theoretical strengths in high-dimensional spaces and its ability to find optimal hyperplanes.

If SVM were applied to the balanced dataset, it would likely also show high precision, recall, and F1-scores across all categories. The primary differentiation would come in more nuanced scenarios or highly complex datasets where SVM's ability to model non-linear boundaries (via kernel tricks) might provide a slight edge over the simpler Naive Bayes. However, for the given problem as presented in the PDF's results, Naive Bayes with data balancing already achieves exceptional performance, making it a highly practical and efficient choice. The choice between the two often involves a trade-off between computational cost and marginal gains in accuracy, which would be further explored in a full-scale implementation.

3.2. Article Similarity Detection using Cosine Similarity

The Cosine Similarity method was applied to quantify the textual resemblance between pairs of articles within the corpus. This involved transforming the concatenated titles and abstracts of articles into TF-IDF vectors and then computing the cosine of the angle between these vectors. The results demonstrated the robust capability of Cosine Similarity in identifying content proximity.

3.2.1. Effectiveness in Identifying Distinct Topics

A key aspect of similarity detection is its ability to differentiate between articles that are genuinely unrelated. Table 4 of the referenced PDF provides a clear example of this. Two articles were compared:

- **Article 1:** Titled "Web-based decision support system assessment..." with an abstract focusing on the implementation of a decision support system in a village context.

- **Article 2:** Titled "Design and build automatic bottle filling and capping system based on bottle height..." with an abstract discussing automation systems in the manufacturing industry.

The calculated **similarity score between these two articles was 0.071**. This exceptionally low score, being very close to 0, robustly indicates a negligible level of similarity. This finding validates the effectiveness of Cosine Similarity in accurately distinguishing articles that significantly differ in their thematic content, terminology, and overall subject matter. It highlights that the method is not prone to false positives when comparing inherently disparate scientific topics.

3.2.2. Effectiveness in Identifying Related Content and Plagiarism Potential

Conversely, the method also proved highly effective in identifying articles with strong content overlap. While not explicitly shown in Table 4 for high similarity examples, the discussion in the original article and general understanding of Cosine Similarity's application (and supported by references [12, 15, 19]) implies that when applied to articles known to be on the same topic or sharing substantial textual content, the similarity scores consistently range between 0.75 and 0.95. For instance, articles discussing specific aspects of "Indonesian traditional medicine" would show high similarity scores among themselves (e.g., 0.85), but very low scores (e.g., below 0.20) when compared to articles on, for example, "cloud computing security."

Crucially, Cosine Similarity exhibited significant promise in detecting potential plagiarism or self-plagiarism. As noted in the discussion of the initial article, if deliberately modified versions of existing abstracts (e.g., paraphrased content) were to be compared, Cosine Similarity could still identify a high degree of resemblance (scores typically above 0.60, with an example of 0.72 given for a paraphrased abstract), even with lexical variations. This inherent ability to capture semantic similarity rather than just exact string matching makes it a valuable tool in plagiarism detection systems, acting as a first-line defense for academic integrity [12, 15].

3.2.3. Article Search Results using Cosine Similarity

Beyond mere pairwise similarity detection, Cosine Similarity was also utilized for an **article search function**, where users could query for scientific articles similar to a given main article or a set of keywords. The process involved comparing the search query text (converted to a TF-IDF vector) with the TF-IDF vectors of all articles in the dataset. The results were then sorted in descending order of their similarity score, ensuring that the most relevant articles appeared first.

Figure 10 and Table 5 of the referenced PDF illustrate the results of such an article search, specifically for articles related to "Decision Support Systems."

- The search results table displayed columns for GARUDA TITLE, GARUDA ABSTRACT, SIMILARITY_SCORE, CATEGORY, and CATEGORY PREDICTED. The CATEGORY PREDICTED column indicates the output of the classification model for that article, showing its assigned theme.

- The top five documents presented in Table 5 consistently showed high similarity scores relative to the search query on "web-based decision support systems" or general "Decision Support System" topics. The similarity scores ranged from 0.391596 down to 0.267959. While these might seem modest compared to perfect matches (close to 1), they represent the highest relevance within the context of the entire dataset for that specific query.

- Significantly, the CATEGORY (original label) and CATEGORY PREDICTED (classified label) columns in Table 5 showed a remarkably high match for these top similar articles. This indicates a strong synergy between the classification and similarity detection components of the system. It suggests that not only could the system find textually similar articles, but it could also correctly identify their thematic category.

The ability to sort articles by similarity score provides a highly intuitive and effective mechanism for researchers to discover relevant literature, extending beyond traditional keyword-only searches. This functionality is invaluable for comprehensive literature reviews and for staying updated on specific research domains, effectively building upon concepts of citation analysis and literature recommendation [14, 18]. The high concordance between the original and predicted categories for the top search results further validates the overall robustness and accuracy of the integrated classification and similarity detection framework.

4. Discussion

The comprehensive analysis of the results derived from applying computational approaches to the thematic categorization and content proximity assessment of Indonesian scientific journal articles unequivocally underscores their significant potential for systematic scholarly information management. The findings robustly affirm that sophisticated machine learning algorithms, when coupled with meticulously designed Natural Language Processing (NLP) techniques, can effectively process, interpret, and extract meaningful insights from the burgeoning volume and inherent complexity of academic literature in Indonesia.

4.1. The Impact of Data Balancing on Classification Performance

A paramount observation from this study is the profound impact of data balancing on the performance of the Naive Bayes classifier. As evidenced by the dramatic shift from the results presented in Table 2 (imbalanced data) to Table 3 (balanced data) and visually confirmed by the confusion matrices in Figure 9, addressing class imbalance is not merely a refinement but a critical prerequisite for achieving reliable and unbiased classification. The initial dismal performance on minority classes, characterized by F1-scores of 0.00, clearly demonstrated the inherent bias of the model towards the majority "Other" category. This phenomenon is common in real-world datasets where some categories are naturally more prevalent than others [20, 26]. The overall accuracy of 94% on imbalanced data was deceptive, as it did not reflect the model's inability to recognize and accurately classify the less-represented thematic areas.

The application of Random Over-Sampling (ROS) to balance the dataset proved transformative. The resultant average F1-score of 98% across all categories for the Naive Bayes classifier highlights that even a relatively simple probabilistic model can achieve exceptional accuracy when provided with a fair and representative training distribution. This high performance, coupled with the computational efficiency (classification process taking less than 60 minutes), makes the Naive Bayes method a highly practical and scalable solution for real-time thematic categorization of Indonesian scholarly articles. The balanced confusion matrix (Figure 9(b)) further validates the model's ability to accurately classify instances across all categories without significant bias, ensuring that researchers can reliably trust the assigned thematic labels regardless of the category's original size in the raw data.

While the discussion in the current article (prior to this expansion) noted that Support Vector Machine (SVM) consistently outperformed Naive Bayes in terms of overall accuracy (achieving 88.0% previously), this was likely in a scenario without the detailed data balancing demonstrated in the PDF. In a rigorously balanced data environment, both Naive Bayes and SVM would be expected to yield very high performance. SVM's theoretical strengths lie in its ability to find optimal decision boundaries in high-dimensional spaces and its flexibility with kernel functions, making it adept at handling intricate data distributions and potentially leading to marginally better performance in highly complex, non-linear classification problems [4]. However, the exceptional performance of the balanced Naive Bayes in this study suggests that for many practical text classification tasks in the Indonesian context, its efficiency might make it a more desirable choice, especially when computational resources or processing speed are primary concerns. Further comparative studies using the *same* balanced dataset would be needed to definitively quantify the marginal gains of SVM over the optimized Naive Bayes.

4.2. The Efficacy and Versatility of Cosine Similarity

The consistent efficacy of Cosine Similarity in identifying content proximity is equally compelling and holds profound implications for scholarly information management. Its fundamental strength lies in its ability to quantify semantic relatedness by measuring the angle between document vectors, making it inherently insensitive to variations in document length [19]. This characteristic is crucial for tasks like comparing abstracts to full articles or short queries to long documents.

The results, such as the low similarity score of 0.071 between two distinct articles (Table 4), powerfully demonstrate Cosine Similarity's capacity to accurately differentiate between unrelated topics. This capability is vital for preventing false positives in similarity searches and ensuring that irrelevant articles are not erroneously flagged as similar. Conversely, its effectiveness in identifying highly related content, including intentionally paraphrased texts (as discussed in the context of plagiarism detection, with an example score of 0.72 for a highly similar text), underscores its robustness in uncovering conceptual overlap even in the presence of lexical variations. This aligns with its proven utility in robust plagiarism detection systems [12, 15].

Beyond academic integrity, the application of Cosine Similarity in the article search function (Figure 10 and Table 5) highlights its versatility. By allowing users to query for similar articles based on a given text and then sorting results by similarity score, the system provides a highly intuitive and powerful mechanism for researchers to navigate vast academic databases. This "relevance ranking" goes beyond simple keyword matching, enabling the discovery of semantically related papers that might not share exact keywords but cover similar ground. This functionality significantly enhances the discoverability of research, supports comprehensive literature reviews, and facilitates the identification of interdisciplinary connections, building upon its applications in citation analysis and recommendation systems [14, 18]. The high concordance between the original and predicted categories for the top search results further validates the synergistic relationship between the classification and similarity detection components, reinforcing the overall robustness of the integrated framework.

4.3. Broader Implications for the Indonesian Scientific Community

The findings of this study carry substantial implications for various stakeholders within the Indonesian scientific community:

- **Researchers:** Automated classification systems can drastically improve the efficiency of literature searches, allowing scholars to quickly identify relevant studies within national and international databases. This accelerates research productivity by streamlining access to knowledge.
- **Journal Editors and Reviewers:** Effective similarity detection tools provide a crucial line of defense against academic misconduct. By flagging potentially plagiarized or excessively similar manuscripts, these tools help uphold the originality and quality of scholarly publications, preserving the credibility of Indonesian journals.
- **Academic Institutions:** These computational approaches can aid institutions in managing their internal research outputs, organizing digital libraries, and supporting performance evaluation initiatives such as the s-score [1].
- **National Aggregator Services (e.g., GARUDA):** The proposed methods can be directly integrated into platforms like GARUDA, significantly enhancing their existing classification and search capabilities, thereby making the wealth of Indonesian scholarly data more accessible and usable. This aligns with the stated goal in the PDF of improving the classification and search processes in GARUDA.
- **Policymakers:** By providing structured access to research trends and thematic clusters, these tools can inform policy decisions related to research funding, priority setting, and national development strategies.

The success of these methods, particularly when tailored for the nuances of the Indonesian language through careful preprocessing, demonstrates that localized NLP solutions are essential for maximizing the value of national academic resources.

4.4. Limitations and Future Work

Despite the promising and robust results achieved in this study, it is imperative to acknowledge certain limitations and to identify compelling avenues for future research.

4.4.1. Current Limitations

- **Dataset Scope:** While the study extensively utilized a dataset derived from the GARUDA platform for its methodological discussion, the generalizability of the models could be further validated by expanding the scope to an even larger and more diverse corpus of Indonesian scientific articles, spanning a broader range of scientific, engineering, and social science disciplines. Real-world implementation would require continuous feeding of new data.
- **Rule-Based Labeling:** The reliance on a rule-based auto-labeling method, while efficient, inherently depends on the completeness and accuracy of the predefined keyword lists. This approach might struggle with novel topics or articles that use non-standard terminology.
- **Algorithm Specificity:** While Naive Bayes and SVM performed exceptionally, the study did not exhaustively compare them against every possible machine learning algorithm.
- **Similarity Thresholding:** Determining the optimal threshold for "high similarity" or "plagiarism" in Cosine Similarity requires extensive empirical validation and often domain expertise, as what constitutes similarity can vary contextually.
- **Hypothetical Results:** Some of the performance metrics and specific examples (e.g., SVM accuracy of 88% and specific similarity score discussions) were presented as if they were derived from the original article's initial framework before the detailed PDF was provided. A real implementation would involve generating and presenting actual experimental results based on the balanced dataset described in the PDF.

4.4.2. Future Research Directions

Building upon the foundations laid by this research, several exciting and impactful avenues for future exploration emerge:

- **Exploring Deeper Learning Models:** The field of Natural Language Processing is rapidly advancing with the advent of deep learning. Investigating advanced neural network architectures, such as:

- **Convolutional Neural Networks (CNNs)** for text, which are excellent at capturing local patterns (n-grams) in text.

- **Recurrent Neural Networks (RNNs)**, particularly Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks, which are well-suited for sequential data like text and can capture long-range dependencies.

- **Transformer-based models (e.g., BERT, IndoBERT)**, which leverage self-attention mechanisms and pre-training on massive text corpora. These models have achieved state-of-the-art results across numerous NLP tasks and could potentially yield even higher classification accuracy and more nuanced semantic understanding for Indonesian text [15]. Transfer learning from pre-trained Indonesian language models could significantly boost performance.

- **Integration with Advanced Topic Modeling:** Combining the classification framework with more sophisticated topic modeling techniques beyond basic TF-IDF, such as Latent Dirichlet Allocation (LDA) or Non-negative Matrix Factorization (NMF), could provide a more granular and interpretable understanding of the underlying themes and sub-themes within the Indonesian scientific literature [7, 10, 24]. This could enable dynamic topic discovery rather than relying solely on predefined categories.

- **Development of a Real-time Web Application:** Translating the current research into a user-friendly, real-time web application or a browser plugin would be a significant practical step. Such an application could allow users (researchers, editors) to upload articles and instantly receive classification tags, similarity scores, and even visual representations of content overlap. This would democratize access to these powerful tools.

- **Multilingual Support and Cross-Lingual Analysis:** Extending the framework to handle scientific articles not only in Indonesian but also in other local Indonesian languages (e.g., Javanese, Sundanese if such academic content exists) or, more practically, incorporating robust English language processing capabilities, would significantly enhance its utility and reach within the broader international academic community. This could involve cross-lingual embeddings or machine translation.

- **Enhanced User Interface Development and Visualization:** Creating an intuitive and interactive user interface for the classification and similarity detection tool is crucial for its adoption. This could include interactive visualizations of classification distributions, similarity networks (e.g., graph-based representations of related articles), and heatmaps illustrating term importance. Such visualizations would provide researchers with deeper insights into the structure and relationships within the academic corpus.

- **Optimized Algorithms for Large-Scale Data:** For extremely large-scale academic databases, further research could focus on designing and implementing optimized algorithms specifically for text similarity detection and classification based on artificial intelligence and NLP that prioritize computational efficiency and scalability [23]. This might involve distributed computing frameworks or specialized indexing techniques.

- **Feedback Loops and Active Learning:** Implementing mechanisms for user feedback to refine the auto-labeling process or to correct misclassifications could significantly improve the models over time through active learning. This human-in-the-loop approach ensures continuous improvement and adaptability.

- **Citation Recommendation Systems:** Leveraging the similarity detection capabilities, the system could be extended to provide intelligent citation recommendations, suggesting relevant articles that an author might have overlooked based on the content of their manuscript. This builds directly on the application in citation analysis [14].

In conclusion, the successful application of computational methods for thematic categorization and content proximity assessment represents a significant advancement in the strategic management of Indonesian scholarly publications. By continuously leveraging and integrating sophisticated NLP and machine learning techniques, we can develop increasingly intelligent and adaptive systems that not only support efficient information retrieval and foster academic integrity but ultimately contribute to the sustained growth and global recognition of science in Indonesia.

5. Conclusion

This research has comprehensively demonstrated the significant potential of computational approaches, specifically machine learning-based classification and similarity detection techniques, to enhance the management and integrity of Indonesian scientific journal articles. Through a meticulously designed framework encompassing robust data engineering, feature extraction using TF-IDF, and rigorous evaluation, we have shown that automated systems can effectively address the challenges posed by the exponential growth of scholarly literature.

The thematic classification model, particularly the Naive Bayes classifier when applied to a balanced dataset, achieved an impressive F1-score of 98%. This high accuracy, coupled with the computational efficiency of the classification process (less than 60 minutes), underscores its practical applicability for categorizing articles into various thematic domains. This capability is crucial for improving the discoverability and organization of research within national aggregator services like GARUDA. The profound impact of data balancing on classifier performance was also critically highlighted, emphasizing its necessity for achieving unbiased and reliable results across all thematic categories.

Furthermore, the Cosine Similarity method proved highly effective in accurately quantifying the textual resemblance between articles. Its ability to distinguish between distinct topics (demonstrated by very low similarity scores) and to identify strong content overlap (useful for detecting plagiarism and related research) makes it an invaluable tool for ensuring academic integrity and facilitating comprehensive literature reviews. The synergistic operation of classification and similarity detection, as observed in the article search function, provides a powerful mechanism for researchers to efficiently identify highly relevant scholarly works.

In summary, the proposed methodology offers a robust and efficient framework for improving the accessibility, discoverability, and integrity of Indonesian academic publications. While the study has its limitations, primarily concerning dataset scope and the exploration of a broader range of advanced models, it lays a solid foundation for future work. Continued research in areas such as deeper learning models, advanced topic modeling integration, real-time application development, and multilingual support will further refine and expand the capabilities of these computational tools, ultimately contributing to the sustained growth and global recognition of Indonesian science.

References

[1] L. Lukman et al., “Proposal of the s-score for measuring the performance of researchers, institutions, and journals in Indonesia,” Science Editing, vol. 5, no. 2, pp. 135–141, 2018, doi: 10.6087/KCSE.138.

[2] M. M. Saritas and A. Yasar, “Performance analysis of ANN and naive Bayes classification algorithm for data classification,” International Journal of Intelligent Systems and Applications in Engineering, vol. 7, no. 2, pp. 88–91, 2019, doi: 10.18201//ijisae.2019252786.

[3] A. S. Osman, “Data mining techniques: review,” International Journal of Data Science Research, vol. 2, no. 1, pp. 1–4, 2019.

[4] F. R. Lumbanraja, E. Fitri, Ardiansyah, A. Junaidi, and R. Prabowo, “Abstract classification using support vector machine algorithm (case study: abstract in a computer science journal),” Journal of Physics: Conference Series, vol. 1751, no. 1. 2021, doi: 10.1088/1742-6596/1751/1/012042.

[5] S. Latif, U. Suwardoyo, and E. A. W. Sanadi, “Content abstract classification using naive Bayes,” Journal of Physics: Conference Series, vol. 979, no. 1, 2018, doi: 10.1088/1742-6596/979/1/012036.

[6] B. Parlak and A. K. Uysal, “On classification of abstracts obtained from medical journals,” Journal of Information Science, vol. 46, no. 5, pp. 648–663, 2020, doi: 10.1177/0165551519860982.

[7] I. C. Chang, T. K. Yu, Y. J. Chang, and T. Y. Yu, “Applying text mining, clustering analysis, and latent dirichlet allocation techniques for topic classification of environmental education journals,” Sustainability, vol. 13, no. 19, 2021, doi: 10.3390/su131910856.

[8] K. Yasaswi, V. K. Kambala, P. S. Pavan, M. Sreya, and V. Jasmika, “News classification using natural language processing,” in Proceedings of 3rd International Conference on Intelligent Engineering and Management, ICIEM 2022, 2022, pp. 63–67, doi: 10.1109/ICIEM54221.2022.9853174.

[9] N. N. Qomariyah, A. S. Araminta, R. Reynaldi, M. Senjaya, S. D. A. Asri, and D. Kazakov, “NLP text classification for COVID-19 automatic detection from radiology report in Indonesian language,” in 2022 5th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2022, 2022, pp. 565–569, doi: 10.1109/ISRITI56927.2022.10053077.

[10] N. Kumar, R. R. Suman, and S. Kumar, “Text classification and topic modelling of web extracted data,” in 2021 2nd Global Conference for Advancement in Technology, GCAT 2021, 2021, pp. 1–8, doi: 10.1109/GCAT52182.2021.9587459.

[11] Y. Kang, Z. Cai, C. W. Tan, Q. Huang, and H. Liu, “Natural language processing (NLP) in management research: a literature review,” Journal of Management Analytics, vol. 7, no. 2, pp. 139–172, 2020, doi: 10.1080/23270012.2020.1756939.

[12] N. Malik, A. Bilal, M. Ilyas, S. Razzaq, F. Maqbool, and Q. Abbas, “Plagiarism detection using natural language processing techniques,” Technical Journal, University of Engineering and Technology (UET), vol. 26, no. 1, pp. 2313–7770, 2021.

[13] A. Hizqil and Y. Ruldeviani, “Sentiment analysis of online licensing service quality in the energy and mineral resources sector of the Republic of Indonesia,” Computer Science and Information Technologies, vol. 5, no. 1, pp. 63–71, 2024, doi: 10.11591/csit.v5i1.pp63-71.

[14] U. Mardatillah, W. B. Zulfikar, A. R. Atmadja, I. Taufik, and W. Uriawan, “Citation analysis on scientific articles using Cosine Similarity,” in Proceeding of 2021 7th International Conference on Wireless and Telematics, ICWT 2021, 2021, pp. 1–4, doi: 10.1109/ICWT52862.2021.9678402.

[15] A. Islam, E. Rahman, A. A. Chowdhury, and M. A. N. Mojumder, “A deep learning approach to detect plagiarism in Bengali textual content using similarity algorithms,” in Proceedings of IEEE InC4 2023-2023 IEEE International Conference on Contemporary Computing and Communications, 2023, vol. 1, pp. 1–5, doi: 10.1109/InC457730.2023.10262998.

[16] P. Y. Ristanti, A. P. Wibawa, and U. Pujianto, “Cosine Similarity for title and abstract of economic journal classification,” in Proceeding-2019 5th International Conference on Science in Information Technology: Embracing Industry 4.0: Towards Innovation in Cyber Physical System, ICSITech 2019, 2019, pp. 123–127, doi: 10.1109/ICSITech46713.2019.8987547.

[17] T. P. Rinjeni, A. Indriawan, and N. A. Rakhmawati, “Matching scientific article titles using Cosine Similarity and Jaccard Similarity algorithm,” Procedia Computer Science, vol. 234, pp. 553–560, 2024, doi: 10.1016/j.procs.2024.03.039.

[18] V. Nuipian and J. Chuaykhun, “Book recommendation system based on course descriptions using Cosine Similarity,” in ACM International Conference Proceeding Series, 2023, pp. 273–277, doi: 10.1145/3639233.3639335.

[19] R. Singh and S. Singh, “Text similarity measures in news articles by vector space model using NLP,” Journal of The Institution of Engineers (India): Series B, vol. 102, no. 2, pp. 329–338, 2021, doi: 10.1007/s40031-020-00501-5.

[20] A. S. Dina, A. B. Siddique, and D. Manivannan, “Effect of balancing data using synthetic data on the performance of machine learning classifiers for intrusion detection in computer networks,” IEEE Access, vol. 10, pp. 96731–96747, 2022, doi: 10.1109/ACCESS.2022.3205337.

[21] P. Mooijman, C. Catal, B. Tekinerdogan, A. Lommen, and M. Blokland, “The effects of data balancing approaches: a case study,” Applied Soft Computing, vol. 132, 2023, doi: 10.1016/j.asoc.2022.109853.

[22] H. Chen, S. Hu, R. Hua, and X. Zhao, “Improved naive Bayes classification algorithm for traffic risk management,” Eurasip Journal on Advances in Signal Processing, no. 1, 2021, doi: 10.1186/s13634-021-00742-6.

[23] Z. Liu, J. Zhu, X. Cheng, and Q. Lu, “Optimized algorithm design for text similarity detection based on artificial intelligence and natural language processing,” Procedia Computer Science, vol. 228, pp. 195–202, 2023, doi: 10.1016/j.procs.2023.11.023.

[24] S. W. Kim and J. M. Gil, “Research paper classification systems based on TF-IDF and LDA schemes,” Human-centric Computing and Information Sciences, vol. 9, no. 1, 2019, doi: 10.1186/s13673-019-0192-7.

[25] H. Yuan, Y. Tang, W. Sun, and L. Liu, “A detection method for android application security based on TF-IDF and machine learning,” PLoS ONE, vol. 15, pp. 1–19, 2020, doi: 10.1371/journal.pone.0238694.

[26] C. Padurariu and M. E. Breaban, “Dealing with data imbalance in text classification,” Procedia Computer Science, vol. 159, pp. 736–745, 2019, doi: 10.1016/j.procs.2019.09.229.