OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Development of a Plagiarism Detection Application Based on Multidimensional Word Vector Representations

PhuongAnh Dao Thi[1], Trong ThaiVan[2]
[1]*Lecturer of Information Technology Faculty, Hanoi University of Natural Resources & Environment, Hanoi, Vietnam*
[2]*School Of Mechanical and Automotive Engineering, Hanoi University of Industries, Hanoi, Vietnam*
Corresponding Author: PhuongAnh Dao Thi

ABSTRACT: *This study focuses on the development of an effective plagiarism detection application for the Vietnamese language, aiming to address the challenges posed by semantic analysis and the complex relationships among words in this language. Although the availability of online textual data continues to grow, traditional methods often fall short in capturing semantic nuances—particularly due to Vietnamese-specific characteristics such as compound words, flexible word order, and strong contextual dependency. To overcome these limitations, we propose an approach based on multidimensional word vector representations, incorporating advanced Vietnamese text preprocessing techniques (word segmentation and normalization) and leveraging cutting-edge word embedding models such as Word2Vec (CBOW and Skip-gram), with a special emphasis on PhoBERT. These models convert text into semantic vectors, enabling the computation of document similarity using Cosine Similarity. This method integrates both semantic and syntactic features to enhance accuracy.*

The application is designed to provide users with a fast and reliable tool for plagiarism detection. Initial experiments demonstrate the potential of the proposed method in improving Vietnamese-language plagiarism detection and suggest promising directions for future research on intelligent text processing systems.

KEY WARDS: *Plagiarism,Detection, Multidimensional Word, Vector Representations, Application*

Date of Submission: 15-06-2025 Date of acceptance: 30-06-2025

I. INTRODUCTION

The explosive growth of the Internet has led to the creation and accumulation of vast amounts of digital textual data, turning it into a valuable and accessible information resource. In the field of text mining, numerous studies and applications have achieved notable success using vector-based text representations alongside traditional probabilistic and statistical methods. However, these approaches often face limitations in capturing deeper semantic aspects and complex word or phrase relationships, especially when applied to natural languages with diverse and rich structures like Vietnamese.

Vietnamese exhibits unique grammatical and semantic features, including the frequent use of compound words, flexibility in word order, and a strong reliance on context to determine meaning. These characteristics make semantic analysis more complicated compared to many other languages. Existing natural language processing (NLP) methods often fail to fully exploit semantic interactions between textual components, such as word frequency, syntactic positions, and contextual dependencies.

To address the issue of plagiarism in academia and publishing, a variety of tools and services have been widely developed. These systems typically provide text comparison capabilities against extensive databases comprising published documents, academic papers, websites, and other content. Below are several widely-used plagiarism detection tools:

- **JPlag**: An open-source tool developed at the University of Karlsruhe, specialized in detecting source code and text plagiarism, with support for the Vietnamese language.
- **Turnitin**: A leading platform in education and research, offering document comparison against a massive database and supporting multiple languages, including Vietnamese.

- **Plagiarism Checker X**: A standalone software application for checking text similarity, delivering detailed reports and supporting many languages, including Vietnamese.
- **Copyleaks**: A cloud-based service employing advanced algorithms to compare content against the internet and private databases, with multilingual support including Vietnamese.
- **iThenticate**: A professional-grade service for research, publishing, and enterprises, notable for its accuracy and its ability to compare Vietnamese texts against reputable academic sources.

These tools play a crucial role in maintaining academic integrity and protecting intellectual property, although each has its own distinctive features.

II. MATERIAL AND METHODS

This section presents an overview of the distinctive grammatical and lexical features of the Vietnamese language, which significantly affect natural language processing (NLP) tasks in general and the problem of plagiarism detection in particular.

2.1. Word Structure in Vietnamese

Vietnamese is an analytic (or isolating) language, characterized by the absence of inflectional morphology in both words and syllables. This means each syllable is typically pronounced independently and may correspond to a word, without morphological changes to indicate tense, case, number, or gender. This characteristic has profound implications on how the language is structured and used, while also introducing unique challenges for syntactic and semantic analysis in NLP.

2.1.1. Concepts of "Tiếng" and "Từ"

In Vietnamese, tiếng refers to the smallest phonological unit, representing a sequence of sounds pronounced in a single utterance. A tiếng may carry an independent meaning or may require combination with others to form a meaningful unit. From a semantic perspective, tiếng can be classified as follows:

- Independent meaning units: Syllables that inherently convey a specific meaning.
- Non-independent meaning units: Syllables that do not carry meaning on their own and must be combined with others.
- Combined units: Some syllables are formed by merging meaningful and non-meaningful components to create compound expressions.

Words (từ) are composed of one or more syllables. Based on their structure, Vietnamese words are typically classified into:

- Simple words: Consist of a single syllable.
- Compound words, which include:

- Compound words (từ ghép): Formed by combining two or more meaningful syllables to create a new word with a complete meaning. These can be further categorized based on the semantic relationship between their components.

- Reduplicative words (từ láy): Comprised of two syllables with phonetic similarities in initial sounds, rhymes, or both. These often convey expressive or mimetic nuances.

Phrases (cụm từ) are groups of words functioning as a grammatical unit and expressing specific meanings within a sentence.

Statistical data on Vietnamese word lengths indicate a characteristic distribution, reflecting the prevalent use of both simple and compound words. Table I presents the frequency distribution of words by the number of constituent syllables:

Table I. Distribution of word lengths in Vietnamese.

Word Length	Frequency	Percentage (%)
1	8399	12,2
2	48995	67.1
3	5727	7.9
4	7040	9.7
≥5	2301	3.1
Total	72994	100

Table 1.1. Distribution of word lengths in Vietnamese.

2.2. Morphological Characteristics in Vietnamese

Unlike inflectional languages, Vietnamese does not utilize prefixes or suffixes to indicate grammatical variations such as tense, case, number, or gender. Instead, changes in meaning or grammatical function are often conveyed through word combinations, word order variations, or auxiliary words. While it lacks complex morphological transformations like those found in English or Romance languages, understanding "transformation" in Vietnamese refers to changes in word roles within a sentence or the creation of compound words to expand vocabulary and express nuanced meanings.

2.3. Synonymy and Orthographic Features in Vietnamese

2.3.1. Synonyms

Synonyms are words that have the same or similar meanings in specific contexts. Although they can often be interchanged without significantly altering sentence meaning, they usually differ in tone, connotation, or usage scope. These subtle distinctions enhance the richness and flexibility of the language. A word may belong to multiple synonym groups depending on context. Identifying and handling synonyms is a key aspect of various NLP tasks, including information retrieval, machine translation, and plagiarism detection, where grasping the true semantic content of text is critical.

2.3.2. Orthographic Features and Processing Challenges

Vietnamese orthographic features present several challenges in automated text processing, including:

- Confusion among homophones and near-homophones;
- Regional dialect variations;
- Capitalization rules;
- Transcription of foreign names and terms;
- Use of hyphens;
- Diversity in punctuation symbols.

These issues increase the complexity of text normalization and require sophisticated preprocessing techniques to ensure accurate NLP outcomes.

2.4. Vietnamese Character Encodings and Normalization

Vietnamese has historically been encoded using various character sets, resulting in a diversity of textual representations. Common legacy encodings include VISCII, VNI, and TCVN3. Currently, Unicode has become the de facto international standard, offering full support for Vietnamese characters and serving as the default for digital text encoding.

Due to the existence of multiple encoding systems, Vietnamese NLP pipelines must incorporate a preprocessing step to normalize text into a unified encoding standard (typically Unicode). This normalization is essential to prevent data loss and ensure consistency in downstream analysis.

III. METHODS AND TEXT PROCESSING PIPELINE

This section details the methods and procedures applied in the task of comparing and detecting plagiarism in Vietnamese texts, including preprocessing steps, text representation models, and similarity measures used.

3.1. Vietnamese Text Preprocessing

Text preprocessing is an essential stage aimed at standardizing and refining raw data before feeding it into machine learning models or natural language processing (NLP) applications. This process improves input data quality, reduces noise, and enhances system performance. Key preprocessing steps include:

- Tokenization: This process divides the input text into the smallest meaningful units, typically words or phrases. Tokenization in Vietnamese presents unique challenges due to fixed compound words, the absence of clear spacing between syllables in multi-syllable words, and punctuation influence. Common issues in Vietnamese word segmentation include:
  - Concatenated words: Difficulty in identifying word boundaries when explicit spaces are lacking.
  - Compound and polysemous words: Accurate segmentation and disambiguation of compound words in context.
  - Punctuation and special characters: Handling punctuation marks (e.g., parentheses, hyphens) and misplaced or confusing special characters.
  - Non-standard writing: Variability in spelling, abbreviations, and typographical errors increases segmentation complexity.
  - Notable models addressing these challenges include:

- PhoBERT: A pre-trained Vietnamese language model based on BERT, capable of handling segmentation and other NLP tasks like text classification.
- Vietnamese Word Segmentation with CRFs: Utilizes Conditional Random Fields with contextual and lexical features.
- VnCoreNLP: A comprehensive NLP toolkit for Vietnamese providing effective word segmentation.
- Deep Learning Models: Uses Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN) for segmentation, yielding promising results on large datasets.

• Normalization: This involves transforming characters into a unified format, such as converting all letters to lowercase, removing punctuation, tonal markers (if task-specific), and unnecessary special characters. Normalization reduces word variations, enhancing the efficiency of subsequent processing steps.

3.2. Text Vector Representation Models

To enable computational comparison and analysis of texts, they must be transformed from natural language into numerical representations. Vector-based models represent each text as a vector in a multidimensional space, where each dimension reflects a specific semantic or syntactic feature.

3.2.1. Word Embedding

Word embeddings are machine learning models that map words or phrases to numerical vectors in a high-dimensional space. Semantically similar words are placed closer together. These representations better capture semantics and inter-word relationships compared to traditional methods like Bag-of-Words or TF-IDF. Common models include:

• Word2Vec: A foundational model offering two architectures:
• Continuous Bag of Words (CBOW): Predicts a target word based on its surrounding context.
• Skip-gram: Predicts surrounding context words from a target word.
• GloVe, FastText: Alternative models that generate high-quality word vectors. Embeddings enrich semantic understanding and reduce data dimensionality, improving NLP task performance.

3.2.2. Transformer Models and PhoBERT

Transformers are neural network architectures introduced for processing sequential data, especially effective in NLP tasks. Their breakthrough feature is the self-attention mechanism, which assigns dynamic weights to input sequence parts, allowing the model to capture long-range dependencies and complex contextual relationships. Transformers are the backbone of many large language models, including BERT.

PhoBERT is a pre-trained Vietnamese language model based on RoBERTa (a BERT variant), specifically fine-tuned on a large Vietnamese corpus. PhoBERT captures Vietnamese syntactic and semantic features more effectively than traditional word embeddings and excels in tasks such as text representation for comparison.

3.3. Text Similarity Measurement Methods

Once text is represented as numerical vectors, various methods are required to quantify similarity between them. The objective is to measure the degree of content similarity between two or more texts.

3.3.1. Cosine Similarity

Cosine similarity measures the cosine of the angle between two vectors in a multidimensional space. It ranges from -1 to 1 (or 0 to 1 for non-negative vectors like embeddings). A value near 1 indicates high similarity. The formula is:

$$Sim(Bi, Bj) = (Bi \cdot Bj) / (||Bi|| * ||Bj||)$$

where · denotes the dot product, and ||Bi||, ||Bj|| are the Euclidean norms. Cosine similarity focuses on vector orientation, making it less sensitive to document length.

3.3.2. Euclidean Distance

Euclidean distance computes the "straight-line" distance between two vectors. The smaller the distance, the more similar the texts. It is defined as:

$$ED(M, N) = \sqrt{\sum (Mi - Ni)^2}, \text{ for } i = 1 \text{ to } n$$

Unlike cosine similarity, Euclidean distance is affected by vector magnitude, meaning text length can significantly impact results.

3.3.3. Jaccard Similarity

Jaccard similarity is a statistical measure for set similarity, typically applied to sets of unique words or n-grams. It is calculated as:

Jaccard(M, N) = |M ∩ N| / |M ∪ N|

Values range from 0 (no overlap) to 1 (identical sets). While simple and effective for lexical overlap, it does not capture semantic or syntactic structure.

3.3.4. Jaro Distance

Jaro distance evaluates similarity between two strings based on matching characters and transpositions. It is especially useful for name matching and typo detection, rather than full-text semantic comparison. The formula is:

dj = (1/3) * (|s1| / m + |s2| / m + (m - t) / m)

where m is the number of matching characters and t is half the number of transpositions.

3.4. Evaluation Metrics

To assess the effectiveness of text comparison or plagiarism detection systems, standard machine learning classification metrics are used:

Precision: The proportion of true positive results out of all predicted positives.

Precision = TP / (TP + FP)

Recall: The proportion of true positive results out of all actual positives.

Recall = TP / (TP + FN)

F1-score: The harmonic mean of Precision and Recall, providing a balanced metric, especially useful for imbalanced datasets.

F1 = 2 * (Precision * Recall) / (Precision + Recall)

The goal is to achieve high Precision, Recall, and F1-score, indicating that the system effectively detects plagiarism without missing or falsely flagging content.

IV. 4PROPOSED SOLUTION AND SYSTEM ARCHITECTURE

This section outlines the overall architecture and detailed steps of the proposed Vietnamese plagiarism detection solution presented in this study. The solution integrates text preprocessing techniques, advanced vector representation models, and similarity measurement methods to accurately detect both semantic and structural similarities in text.

4.1. Overall System Architecture

The proposed plagiarism detection system is designed with a modular architecture, as illustrated in Figure1:

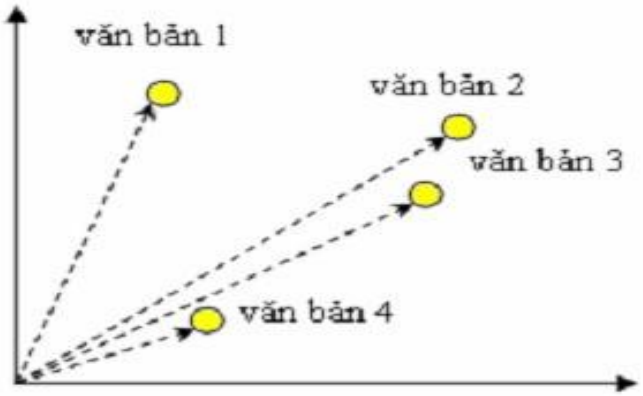

Figure 1. Vector-based text representation model.

4.2. Implementation Steps in the Proposed Solution

The plagiarism detection workflow is carried out in the following steps:

4.2.1. Data Collection and Preprocessing

The initial phase involves collecting Vietnamese text datasets from diverse and credible sources. The collected data then undergoes preprocessing to clean and standardize the content, including the following steps:

Tokenization

Normalization

Stopword Removal

Low-frequency Word Removal

Sentence Segmentation

4.2.2. Vector-based Text Representation

Following preprocessing, each document is transformed into a numerical vector representation in a multidimensional space. These representations enable the system to capture semantic content, allowing computational comparisons. In this study, we focus on leveraging advanced word embedding models:

Word Embedding Models: These models map words or phrases to numerical vectors, ensuring that semantically similar words are located close to each other in vector space. Models such as Word2Vec, using both Continuous Bag of Words (CBOW) and Skip-gram architectures, are employed to learn vector representations from large-scale text corpora based on contextual word usage. These embeddings provide richer semantic information than traditional methods like TF-IDF, while also reducing dimensionality and improving computational efficiency.

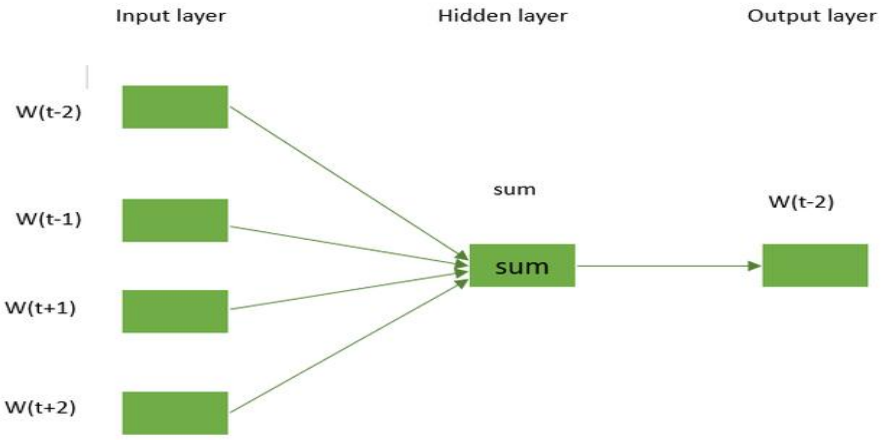

Figure 2. CBOW Architecture.

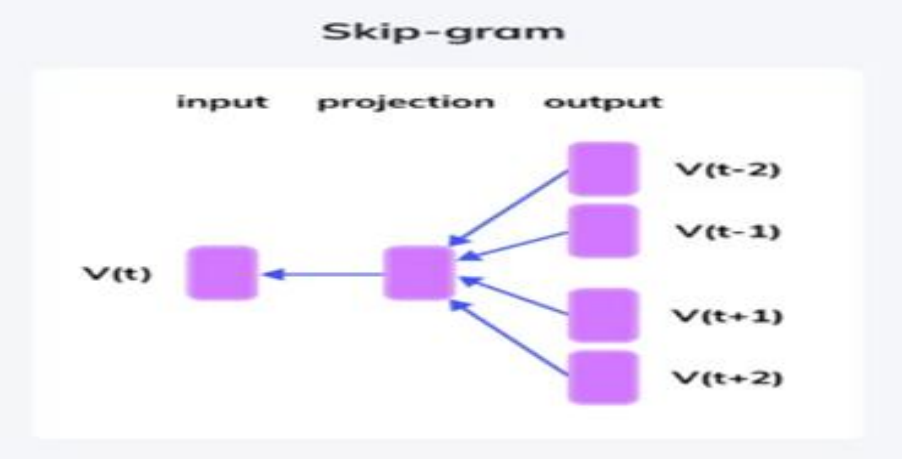

Figure 3. Skip-gram Architecture.

A brief explanation of the Softmax function in Word2Vec may be included here or referenced in Section 3 if the theoretical foundation has already been introduced.

Transformer Models and PhoBERT: To enhance the representation of semantic and contextual relationships in Vietnamese, we utilize PhoBERT—a variant of BERT (Bidirectional Encoder Representations from Transformers) pre-trained on large-scale Vietnamese corpora. The Transformer architecture employed in PhoBERT, particularly its self-attention mechanism, enables the model to capture long-range dependencies and contextual relationships in both directions. Using PhoBERT facilitates the generation of high-quality semantic vectors that more accurately reflect the meaning of Vietnamese text.

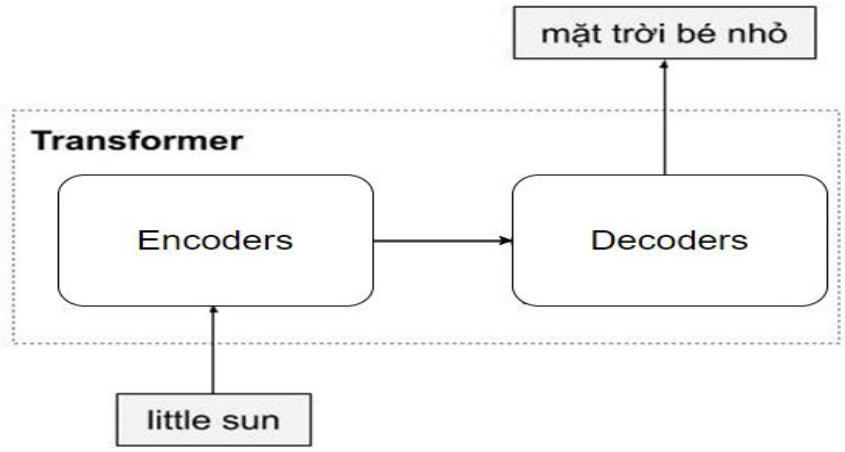

Figure 4. Transformer Model.

V. SIMILARITY MEASUREMENT METHODS

There are various methods for measuring textual similarity, including statistical approaches such as term frequency indices, context analysis, and machine learning techniques that analyze the syntax and semantics of the text.

For natural language processing (NLP) models such as BERT or Transformer, similarity is typically measured using mathematical operations like cosine distance or Euclidean distance between the vector representations of the compared entities in a high-dimensional space.

5.1. Cosine Similarity

Cosine similarity is a commonly used method to measure the similarity between two vectors. It evaluates the cosine of the angle between the two vectors, returning a value between 0 (completely dissimilar) and 1 (identical). The dimensionality of the vector space corresponds to the number of unique terms in the vocabulary. The value of each vector component represents the importance of the corresponding term in the sentence, which can be calculated using previously presented techniques such as term frequency-inverse document frequency (TF-IDF) or word embeddings (e.g., Word2Vec) [3][4].

Assume two documents are represented as vectors:
Vector $B_i = <v_1, ..., v_t>$, where v_t is the weight of the t-th term in document 1.
Vector $B_j = <v_1, ..., v_t>$, where v_t is the weight of the t-th term in document 2.
The cosine similarity is calculated as follows:

$$\text{Sim}(B_i, B_j) = \frac{Bi * Bj}{|Bi||Bj|} = \frac{\sum_{k=1}^{t}(Bi * Bj)}{\sqrt{\sum_{k=1}^{t}(Bi)^2 * \sum_{k=1}^{t}(Bj)^2}}$$

Trong đó:

- $Bi * Bj$ is the dot product of the vectors B_i and B_j
- $|Bi| * |Bj|$ denotes the product of the magnitudes of vectors B_i and B_j.

5.2. Euclidean Distance

Euclidean distance is a measure of the "straight-line" distance between two points in a multidimensional space, calculated based on the difference between corresponding vector components [11].

Each document can be represented as a vector using methods like TF-IDF or Word2Vec. Given two vectors M and N, the Euclidean distance is computed as:

$$\text{ED}(M, N) = \sqrt{\sum_{i=1}^{n}(M_i - N_i)^2}$$

Trong đó:

- M_i and N_i are the i-th components of vectors **M** and **N**, respectively.
- n is the dimensionality of the vector space

A smaller Euclidean distance indicates greater similarity between the documents in the vector space. However, it is important to note that Euclidean distance may not accurately reflect semantic relationships when using TF-IDF or Word2Vec vectors. In many cases, cosine similarity is preferred for measuring text similarity with vector-based representations.

5.3. Jaccard Similarity

Jaccard similarity is a metric for measuring the similarity between two sets. It is calculated as the size of the intersection divided by the size of the union of the sets, yielding a value between 0 and 1—where 0 indicates no shared elements and 1 indicates complete overlap. The Jaccard similarity between two sets M and N is defined as:

This metric is relatively weak in capturing semantic similarity and is therefore less suitable for conventional textual comparisons.

$$\text{Jaccard similarity (M, N)} = \frac{|M \cap N|}{|M \cup N|}$$

Where:

|M ∩ N| is the number of elements common to both sets.

|M ∪ N| is the total number of distinct elements across both sets.

This metric is relatively weak in capturing semantic similarity and is therefore less suitable for conventional textual comparisons.

Note:

Jaccard similarity is best used to evaluate the overlap between two sets in terms of shared elements. It is typically less effective than other methods when applied to standard textual content.

5.4. Jaro Distance

Jaro distance is a metric for comparing the similarity between two character strings, frequently used in NLP and database applications.

Given two strings b_1 and b_2, the Jaro distance d is calculated as:

$$d = \frac{1}{3}\left(\frac{m}{|b_1|} + \frac{m}{|b_2|} + \frac{m-t}{m}\right)$$

Where: m is the number of matching characters, t is ½ of transpositions.

Each word in b_1 is compared with all words in b_2, and transpositions are counted as half of the mismatches in positions. The Jaro score ranges from 0 (completely dissimilar) to 1 (identical).

5.5. Text Similarity Evaluation

Text similarity can be assessed using distance metrics such as cosine similarity and Euclidean distance.

In Word2Vec, each word is represented as a numerical vector that captures its semantic and contextual information. Cosine similarity is typically used to evaluate the similarity between the vector representations of documents.

Using Word2Vec yields reliable and accurate similarity results. However, interpreting these results requires deep understanding of natural language and machine learning techniques. Incorporating both semantic similarity and word order provides a more comprehensive measure of textual similarity.

Assume the two tokenized documents are:

$$B_1 = \{ v_{11}, v_{12}, \ldots, v_{1a1}\}$$
$$B_2 = \{ v_{21}, v_{22}, \ldots, v_{2a2}\}$$

Trong đó: v_{ij} is the j-th term in document b_i (i=1,2)

a_i is the number of terms in document B_i.

Let $B = B_1 \cup B_2 = \{v_1, v_2, ..., v_a\}$ be the set of all distinct terms across both documents.

Semantic feature vector $U_1 = (U_{11}, U_{12}, ..., U_{1m})$ for document B_1 is constructed as follows:

Among the analyzed methods, semantic similarity captures meaning-level alignment between words, while structural similarity reflects word order relationships. Both aspects are essential for determining overall text similarity. Therefore, a comprehensive similarity measure should combine these two aspects, expressed as:

VI. CONCLUSIONS AND RECOMMENDATIONS

This study successfully developed an effective plagiarism detection application for the Vietnamese language. We addressed the linguistic challenges specific to this isolating language through a comprehensive preprocessing pipeline and by employing advanced semantic vector representations such as Word2Vec and PhoBERT.

The proposed solution uses cosine similarity in a hybrid approach that combines semantic similarity with word order similarity. This enables the system not only to detect exact word matches but also to capture semantic and structural nuances, thereby improving detection accuracy.

However, limitations remain regarding the detection of idea-level plagiarism and the optimization of similarity metric weights. In future work, we aim to enhance Vietnamese word embedding models, integrate advanced deep learning techniques for handling more complex plagiarism cases, automate parameter tuning, improve the user interface, and expand evaluation using larger datasets. Our ultimate goal is to develop more robust tools to uphold integrity in digital content.

REFERENCES

[1]. Circuit, W.V.P. (1943), "transFORMER™", The Massachusetts Institute of Technology, 1(1), tr.187-194.
[2]. Guerra, Francisco das Chagas Fernandes, and Wellington Santos Mota (2006), "Current transformer model", IEEE Transactions on Power Delivery, 22(1), tr.187-194.
[3]. Luo, C., Zhan, J., Xue, X., Wang, L., Ren, R., & Yang, Q. (2018), "Cosine normalization: Using cosine similarity instead of dot product in neural networks", Artificial Neural Networks and Machine Learning, 27(1), tr.382-391.
[4]. Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016), "Cosine similarity to determine similarity measures: A case study in online essay evaluation", 2016 4th International Conference on Advanced Computer Science and Information Systems, tr.1-6.
[5]. Onan, A., Korukoglu, S., & Bulut, H. (2016), "Ensemble of keyword extraction methods and classifiers in text classification", Expert Systems with Applications, 57(1), tr.232-247.
[6]. Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. C. J. (2019), "Evaluating word embedding models: Methods and experimental results", APSIPA transactions on signal and information processing, 8(19), tr.1-10.