

Android-Based Research Title Similarity Detection Using a Combined Word2Vec and TF-IDF with Cosine Similarity Score

Abyan Dzakwan Baksir^{1*}, Rosihan², Muhammad Ridha Albaar³

^{1,2,3}Department of Informatics, Faculty of Engineering, Universitas Khairun, Ternate, North Maluku, Indonesia
E-mail: ^{1*}baksirabyan@gmail.com, ²rosihan@unkhair.ac.id, ³ridha.albaar@unkhair.ac.id

(Received: 11 May 2026, revised: 4 Jun 2026, accepted: 8 Jun 2026)

Abstract

Research title similarity detection is needed to support academic topic checking because manual title comparison is time-consuming and may fail to identify related topics expressed with different terms. This study develops an Android-based research title similarity detection system using a lightweight hybrid lexical-semantic approach that combines TF-IDF, Word2Vec Continuous Bag of Words (CBOW), and cosine similarity. The dataset was obtained through scraping from the GARUDA Portal and curated into 10,000 research titles related to computer science and information technology. The titles were processed through case folding, tokenization, stopwords removal, and stemming using Sastrawi. TF-IDF was used to represent lexical term importance, while Word2Vec CBOW was used to capture contextual word relationships. The two similarity scores were integrated using weighted alpha configurations of 0.50, 0.60, and 0.70. The model was implemented in a Python FastAPI backend and tested through a Flutter-based Android application. Evaluation was conducted using 30 query titles with Top-5 retrieval results and manual relevance judgment based on a predefined 0–2 relevance rubric. The results show that TF-IDF only achieved the highest MAP@5 of 0.985972 and NDCG@5 of 0.939095. Among the hybrid configurations, alpha 0.70 produced the best performance with Precision@5 of 0.973333, MAP@5 of 0.981111, and NDCG@5 of 0.923408. These findings indicate that the hybrid model is competitive and more effective than Word2Vec only, while TF-IDF remains highly important for short research title matching.

Keywords: Research Title Similarity, Word2Vec, TF-IDF, Cosine Similarity, Android

I. INTRODUCTION

The rapid growth of academic publications has increased the need for efficient mechanisms to manage and examine research titles. In many academic environments, research titles are often checked manually to determine whether a proposed topic is similar to existing studies. This process becomes increasingly difficult when the number of available titles is large, because manual checking requires direct comparison, consumes time, and may produce subjective results. Prior studies on scientific article classification and recommendation show that the growth of digital academic documents creates information overload and requires automated NLP-based retrieval or recommendation mechanisms [1], [2]. Therefore, an automatic title similarity detection approach is needed to support a more consistent and efficient research title checking process.

Research title similarity detection is a specific text similarity problem that focuses on comparing short academic texts. Unlike long documents, research titles contain only a limited number of words, but each word usually carries

important information about the topic, method, object, or platform used in the study. Short and domain-specific texts are challenging because lexical overlap alone may fail to capture contextual meaning, semantic nuance, and morphological variation [3], [4]. For example, titles related to information systems, Android applications, data mining, classification, or recommendation systems may share similar research directions even when the exact wording is different. This condition makes title similarity detection challenging because the system must consider both lexical similarity and contextual meaning.

Several text representation methods have been widely used for similarity detection. Term Frequency-Inverse Document Frequency (TF-IDF) is effective in representing the importance of terms in a document and is useful for identifying lexical similarity between texts. In information retrieval tasks, TF-IDF transforms textual data into numerical term weights, while cosine similarity is used to measure the relevance between vectors [5]. However, TF-IDF mainly depends on term occurrence and may not fully capture semantic relationships between different but related words. On the other



hand, Word2Vec represents words as dense vectors and can capture contextual relationships between terms based on their surrounding words. Previous evaluations show that Word2Vec and FastText are widely used to capture semantic relationships in vector representations and can support similarity measurement for short texts. Nevertheless, Word2Vec may give less emphasis to specific domain terms that are important in short research titles. These characteristics indicate that TF-IDF and Word2Vec have complementary strengths for research title similarity detection [6].

Model performance in text similarity tasks depends on dataset characteristics and text context. Prior studies indicate that weighted and hybrid vectorization approaches can improve similarity estimation, while TF-IDF remains effective for distinguishing texts based on important term distribution and Word2Vec captures semantic relationships between words [7], [8], [9]. These findings support the combination of lexical and contextual representations for research title similarity detection.

Based on this consideration, this study proposes an Android-based research title similarity detection system using a combined TF-IDF and Word2Vec CBOW approach with cosine similarity. The proposed method calculates TF-IDF cosine similarity to represent lexical term importance and Word2Vec cosine similarity to represent contextual relationships between words. The two scores are then integrated through a weighted alpha-based combined similarity score to produce ranked research title recommendations. The dataset used in this study consists of 10,000 curated research titles from the GARUDA Portal, focusing on computer science and information technology topics.

The novelty of this study is not positioned as a new embedding architecture, because hybrid lexical-semantic approaches have been widely used in previous text similarity studies. Instead, the main contribution lies in the development and evaluation of a lightweight Android-based research title similarity checking system that integrates TF-IDF and Word2Vec CBOW on a curated Indonesian academic title dataset. This study also provides an empirical comparison between TF-IDF only, Word2Vec only, and three hybrid alpha configurations to examine how lexical and contextual weighting affects Top-5 similarity retrieval performance.

This study contributes in three aspects. First, it prepares a curated GARUDA-based dataset of research titles related to computer science and information technology for short academic title similarity detection. Second, it evaluates TF-IDF, Word2Vec CBOW, and combined similarity scores using Precision@5, MAP@5, and NDCG@5 based on manually labeled relevance judgments. Third, it implements the similarity model in a Python FastAPI backend and a Flutter-based Android prototype, allowing users to input a research title, select the number of returned results, adjust the TF-IDF weight, and view ranked similar titles with their TF-IDF, Word2Vec, and final combined scores.

Although transformer-based embeddings such as BERT and Sentence-BERT have become popular in recent text similarity research, this study uses Word2Vec CBOW because

the system is designed as a lightweight prototype for short research title matching with lower computational requirements. Word2Vec CBOW is also suitable for learning contextual relationships from the available title corpus, while TF-IDF preserves the importance of discriminative domain terms that frequently determine title relevance in short academic texts. Therefore, the proposed system emphasizes a practical balance between computational efficiency, lexical matching, contextual similarity, and Android-based deployment.

II. RESEARCH METHODOLOGY

This study applies an experimental system development approach to detect the similarity of research titles using a combined similarity score. The proposed system integrates two text representation methods, namely TF-IDF and Word2Vec, with Cosine Similarity as the similarity measurement technique. TF-IDF is used to represent lexical term importance, while Word2Vec is used to capture contextual relationships between words. Similar methodological designs have been used in information retrieval and short-text similarity tasks, where TF-IDF supports lexical weighting and cosine similarity measures vector-based relevance [5], while Word2Vec-based embeddings strengthen semantic representation [3], [7]. The similarity scores generated from both methods are then combined into a final similarity score to rank the most relevant research titles. The model was implemented in Python and connected to a Flutter-based Android application through a local REST API.

A. Research Design

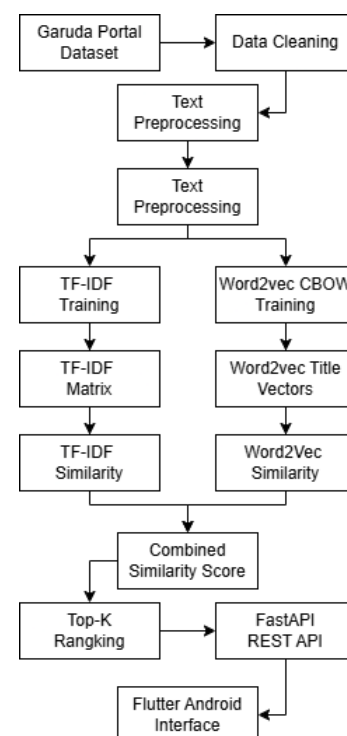


Figure 1. Proposed Research Workflow

The research process consists of dataset collection, text preprocessing, TF-IDF training, Word2Vec training, title vector construction, combined similarity calculation, REST API implementation, and Android-based testing. The overall workflow is shown in Figure 1.

Each query follows the same preprocessing pipeline, is transformed into TF-IDF and Word2Vec vectors, and is compared using cosine similarity. The two scores are then combined to rank candidate titles and display the results in the Android application.

B. Dataset Collection

The dataset was collected from the GARUDA Portal through a scraping process as shown in Table 1. The initial scraped data contained more than 500,000 academic title records from various scientific fields. Since this study focuses on research title similarity detection in computer science and information technology, the dataset was curated by selecting titles from journals and topics related to computer science, information systems, information technology, software development, data mining, artificial intelligence, mobile applications, and web-based systems. From this process, 10,000 research titles were selected as a controlled experimental dataset.

The final dataset was stored in a local SQLite database. Each record contains the original title, author information, preprocessed title, original token count, and processed token count. The title field was used as the main textual feature because the objective of this study is to detect similarity between short academic research titles. The use of a curated dataset was intended to ensure that the model was trained and tested on titles that were relevant to the target domain

Table 1. Dataset Description

Item	Description
Dataset source	Garuda Portal
Collection method	Scraping
Initial scraped records	More than 500,000 academic title records
Dataset domain	Computer science and information technology
Selected dataset	10,000 research titles
Storage format	SQLite database
Main textual feature	Research title
Main training column	Title_preprocessed
Additional information	Author, original token count, processed count

C. Text Preprocessing

Text preprocessing is applied to normalize the research titles before they are used in the similarity model. The preprocessing stage consists of four main processes: case folding, tokenization, stopwords removal, and stemming as shown in Table 2. These stages are commonly used in Indonesian text processing to reduce lexical variation, remove non-informative terms, and prepare clean input for TF-IDF or embedding-based representation [2], [5], [7], [10]. Case

folding converts all characters into lowercase form to reduce variations caused by capitalization. Tokenization splits each title into individual word units. Stopword removal eliminates common words that do not significantly represent the topic of the title. Stemming is applied to reduce words into their root forms.

The preprocessed titles are stored in the title_preprocessed column and used as input for TF-IDF and Word2Vec training. This column ensures that both representation methods use the same normalized textual input, allowing the similarity calculation to be performed consistently across all research titles. Relevant academic and technical terms are preserved to maintain domain meaning, while unnecessary lexical variation is reduced. By storing the preprocessed output separately from the original title, the system can retain the original title for display purposes while using the cleaned version for model training and similarity computation. This separation also supports more efficient processing because the preprocessing stage does not need to be repeated each time a similarity query is performed.

Table 2. Example of Text Preprocessing

Original Title	Preprocessed Title
Sistem Deteksi Kemiripan Judul Penelitian Berbasis Android	sistem deteksi mirip judul teliti android

D. TF-IDF-Based Similarity Calculation

TF-IDF is used to calculate the importance of each term in a research title relative to the entire title corpus. This representation is suitable for short academic titles because it emphasizes discriminative terms and reduces the influence of common terms. Previous information retrieval studies have shown that TF-IDF can transform text into numerical term weights and support relevance measurement when combined with cosine similarity [5]. The TF-IDF value is calculated using Formula 1.

$$TFIDF(t, d) = TF(t, d) \times IDF(t)$$

(1)

where t represents a term, d represents a research title, TF(t,d) is the frequency of term t in title d, and IDF(t) represents the inverse document frequency of term t in the whole title corpus. After the TF-IDF matrix is generated, the input query title is transformed using the same vectorizer. The similarity between the query vector and each title vector in the dataset is then calculated using Cosine Similarity. Similar feature-fusion approaches that combine TF-IDF and Word2Vec have also been used to improve web information extraction and retrieval tasks [11]. TF-IDF has also been integrated into domain-aware retrieval frameworks to strengthen lexical-level document retrieval [12]. Cosine Similarity is suitable for this process because it compares the direction of vector representations and has been used in NLP-based text matching tasks to measure compatibility between textual inputs [13].

The TF-IDF-based similarity score is defined in Formula 2.

$$S_{TFIDF_i} = \text{Cosine}(V_{TFIDF(q)}, V_{TFIDF(d_i)}) \quad (2)$$

where S_{TFIDF_i} is the TF-IDF similarity score for the i -th candidate title, $V_{TFIDF(q)}$ is the TF-IDF vector of the query title, and $V_{TFIDF(d_i)}$ is the TF-IDF vector of the i -th title in the dataset. This score is used to measure lexical similarity based on term importance.

E. Word2Vec CBOW-Based Similarity Calculation

Word2Vec is used to represent contextual relationships between words in the research title dataset. This study uses the Continuous Bag of Words architecture because it learns word representations by predicting a target word from its surrounding context words. Prior evaluation of short educational texts shows that Word2Vec combined with CBOW and TF-IDF weighting can provide stable similarity estimates, particularly when cosine similarity is used to compare vector representations [3]. Each word is represented as a dense vector, and each research title is represented by averaging the vectors of its tokens.

The vector representation of a title is calculated using Formula 3.

$$V_{title} = \left(\frac{1}{n}\right) \sum_{j=1}^n V(w_j) \quad (3)$$

where V_{title} is the vector representation of a research title, $V(w_j)$ is the Word2Vec vector of the j -th token, and n is the number of tokens found in the Word2Vec vocabulary. Tokens that are not available in the Word2Vec vocabulary are ignored during vector construction. If no token is found in the vocabulary, the system assigns a zero vector to the title representation.

After the query title is converted into a Word2Vec-based vector, Cosine Similarity is calculated between the query vector and all title vectors in the dataset. The Word2Vec-based similarity score is defined in Formula 4.

$$S_{W2V_i} = \text{Cosine}(V_{W2V(q)}, V_{W2V(d_i)}) \quad (4)$$

where S_{W2V_i} is the Word2Vec similarity score for the i -th candidate title, $V_{W2V(q)}$ is the Word2Vec-based vector of the query title, and $V_{W2V(d_i)}$ is the Word2Vec-based vector of the i -th title. This score is used to capture contextual similarity between research titles.

F. Combined Similarity Score

The main contribution of this study is the use of a combined similarity score that integrates TF-IDF-based similarity and Word2Vec-based similarity. TF-IDF contributes lexical term importance, while Word2Vec contributes contextual similarity between words. Since both methods represent different aspects of text similarity, combining them is expected to produce a more balanced similarity score for short research titles. Hybrid similarity

studies show that combining statistical weighting with embedding-based representation can improve ranking quality, semantic relevance, and document recommendation accuracy [2], [4], [7].

The final combined similarity score is calculated using Formula 5.

$$SG_i = \alpha S_{TFIDF_i} + (1 - \alpha) S_{W2V_i} \quad (5)$$

where SG_i is the final combined similarity score for the i -th candidate title, S_{TFIDF_i} is the TF-IDF cosine similarity score, S_{W2V_i} is the Word2Vec cosine similarity score, and α is the weight assigned to TF-IDF. The value of α ranges from 0 to 1. If α is set to 0.6, the TF-IDF score contributes 60% to the final score, while the Word2Vec score contributes 40%.

Before calculating the combined score, the Word2Vec similarity score is adjusted to maintain an interpretable score range. Since Cosine Similarity on Word2Vec vectors may produce negative values, negative Word2Vec scores are clipped to 0. This adjustment ensures that the final combined score remains within a range that can be interpreted as a similarity level from 0 to 1.

After the final score is obtained for all candidate titles, the system sorts the titles in descending order based on SG_i .

G. REST API and Android Prototype

The similarity model was implemented in Python and served through a local FastAPI-based REST API. The API receives a query title, applies the same preprocessing pipeline, calculates TF-IDF and Word2Vec similarity scores, computes the final combined score, and returns ranked results in JSON format.

The system provides two main API endpoints. The first endpoint is `/health`, which is used to check whether the similarity service is active. The second endpoint is `/search`, which is used to search for the most similar research titles. The `/search` endpoint receives three main input fields: `query`, `top_k`, and `alpha` (Table 3). The `query` field contains the research title entered by the user, `top_k` determines the number of returned results (Table 4), and `alpha` determines the TF-IDF weight in the combined score calculation.

The Android prototype was developed using Flutter as a testing interface. It sends user queries to the REST API and displays ranked titles with TF-IDF (Table 5), Word2Vec (Table 6), and final combined scores. The similarity calculation is performed only on the backend.

Table 3. REST API Request Structure

Field	Data Type	Description
Query	String	Research title entered by the user
Top_k	Integer	Number of most similar titles returned by the system
Alpha	Float	TF-IDF weight used in the combined similarity score

Table 4. REST API Response Structure

Field	Description
Rank	Ranking position of the candidate title
Title_original	Original research title from the dataset
Title_preprocessed	Preprocessed version of the title
Tfidf_score	Similarity score generated using TF-IDF and Cosine Similarity
Word2Vec_score	Similarity score generated using Word2Vec and Cosine Similarity
Final_score	Final score generated using the combined similarity formula
Alpha	TF-IDF weight used in the score calculation

Table 5. TF-IDF Configuration

Parameter	Value
Input column	Title_preprocessed
Lowercase	False
Token pattern	(?u)\b\w+\b
N-gram range	(1,1)
Minimum document freq	1
Normalization	L2
Use IDF	True
Smooth IDF	True
Sublinear TF	False

Table 6. Word2Vec CBOW Configuration

Parameter	Value
Architecture	CBOW
Sg	0
Vector size	100
Window	5
Minimum	1
Workers	4
Epochs	30
Seed	42
Input	Tokenized preprocessed titles

H. Similarity Validation Procedure

Similarity validation was conducted to evaluate whether the retrieved research titles were relevant to the input query. Since the system produces ranked retrieval results, the validation focused on Top-5 search results. A total of 30 query titles were prepared to represent common topics in computer science and information technology, including information systems, Android applications, decision support systems, data mining, machine learning, geographic information systems, network security, recommendation systems, usability evaluation, and expert systems.

For each query, the system generated Top-5 results using five retrieval configurations: TF-IDF only, Word2Vec only, combined alpha 0.50, combined alpha 0.60, and combined alpha 0.70. This process produced 750 raw retrieval results. Duplicate query-title pairs appearing across different methods

were consolidated, resulting in 300 unique query-title pairs for manual relevance assessment.

The relevance of each query-title pair was assessed using a predefined 0–2 relevance rubric. A score of 0 indicates that the retrieved title is not relevant to the query. A score of 1 indicates partial relevance, where the retrieved title shares a general topic or domain but differs in important aspects such as method, object, platform, or research objective. A score of 2 indicates high relevance, where the retrieved title has strong similarity with the query in terms of topic, domain, method, object, platform, or research objective. The relevance judgment was conducted by the author using the predefined rubric to ensure consistency. This single-annotator approach is acknowledged as a limitation, and future work should involve multiple domain experts to improve the reliability of relevance assessment.

I. Evaluation Metrics

The model was evaluated using ranking-based retrieval metrics because the system returns a ranked list of similar research titles. The metrics used in this study were Precision@5, Mean Average Precision at 5 (MAP@5), and Normalized Discounted Cumulative Gain at 5 (NDCG@5). Precision@5 measures the proportion of relevant titles among the top five retrieved results. In this study, titles with relevance labels of 1 or 2 were treated as relevant for Precision@5 calculation.

MAP@5 was used to evaluate the ranking quality by considering the positions of relevant titles in the Top-5 results. A higher MAP@5 value indicates that relevant titles appear earlier in the ranking. NDCG@5 was used to measure ranking quality by considering graded relevance values of 0, 1, and 2. This metric is suitable because the relevance between research titles is not always binary; a title can be partially relevant or highly relevant depending on the similarity of topic, method, object, platform, and research objective.

J. Baseline and Alpha Testing

To evaluate the effectiveness of the proposed combined similarity score, this study compared five retrieval configurations. TF-IDF only was used as a lexical baseline, Word2Vec only was used as a contextual baseline, and three combined configurations were tested using alpha values of 0.50, 0.60, and 0.70. The alpha value determines the weight assigned to TF-IDF in the final combined score, while the remaining weight is assigned to Word2Vec.

The alpha 0.50 configuration represents an equal contribution between lexical and contextual similarity. The alpha 0.60 configuration gives a slightly higher weight to TF-IDF, following the initial implementation used in the Android prototype. The alpha 0.70 configuration gives a stronger emphasis to lexical similarity, which is expected to be important for short research titles because specific terms such as Android, Laravel, Naive Bayes, K-Means, and information system often determine the relevance of retrieved titles.

III. RESULT AND DISCUSSION

This section presents the results of dataset preparation, model resource generation, similarity search, method comparison, and Android prototype testing. The evaluation focuses on how the proposed combined similarity score ranks research titles based on lexical similarity from TF-IDF and contextual similarity from Word2Vec. The implementation results are also discussed to show how the Python backend and Flutter-based Android interface support the similarity search process.

A. Dataset and Preprocessing Results

The dataset used in this study was collected from the GARUDA Portal through a scraping process. The dataset consists of 10,000 research title records and was stored in a local SQLite database. Each record contains the original research title, author information, preprocessed title, original token count, and processed token count. The preprocessed dataset was used as the main input for TF-IDF and Word2Vec model training.

The dataset was cleaned through case folding, tokenization, stopword removal, and stemming using Sastrawi. These stages reduced lexical variation and prepared consistent input for TF-IDF and Word2Vec training.

The use of Sastrawi was considered appropriate because the dataset contains many Indonesian research titles mixed with technical terms from computer science and information technology. Since many method names and technical terms are written in English, such as “content management system”, “genetic algorithm”, “network simulator”, “data encryption standard”, and “Google Maps API”, English stemming was not applied. This decision was made to preserve technical terms and method names so that they remain meaningful during similarity calculation. Therefore, the preprocessing process focuses on normalizing Indonesian word forms while maintaining important English technical terms.

Table 7. Dataset Summary

Item	Value
Dataset source	Garuda Portal
Collection method	Scraping
Initial number of titles	More than 500,000
Selected records	10,000
Total records used	10,000
Storage format	SQLite database
Empty original titles	0
Empty preprocessed titles	0
Unique original titles	10,000
Unique preprocessed titles	9,947
Duplicate preprocessed titles	53
Average original token count	12,66
Average processed token count	11,49
Minimum processed token count	1
Maximum processed token count	43

Table 7 shows that the dataset was curated from more than 500,000 initial scraped records into 10,000 research titles

related to computer science and information technology. The final dataset contains no empty original or preprocessed titles. The number of unique preprocessed titles is lower than the number of original titles because preprocessing can make different titles identical after case folding, stopword removal, and stemming.

Table 8 shows examples of the preprocessing results. The examples indicate that Indonesian words are simplified into root forms, while technical terms related to computer science and information technology are preserved. This preprocessing result is important because TF-IDF requires clean lexical terms to calculate term importance, while Word2Vec requires meaningful tokens to learn contextual relations between words. By reducing unnecessary word variations, the preprocessing stage helps create more consistent textual input for both representation methods. At the same time, preserving technical terms ensures that important domain-specific meanings, such as algorithm names, platforms, and system-related concepts, are not removed during text normalization.

Table 8. Example of Text Preprocessing Result

Original Title	Preprocessed Title
SISTEM PENUNJANG KEPUTUSAN PEMILIHAN PROGRAM STUDI DI UNIVERSITAS MULAWARMAN MENGUNAKAN METODE PROMETHEE	sistem tunjang putus pilih program studi universitas mulawarman guna metode promethee
Sistem Pendukung Keputusan Pemilihan Program Studi Di Universitas Mulawarman Menggunakan Metode Tsukamoto (Studi Kasus : Fakultas MIPA)	sistem dukung putus pilih program studi universitas mulawarman guna metode tsukamoto studi kasus fakultas mipa
ANALISIS JARINGAN DENGAN PARAMETER RANDOMIZE FILE SIZE MENGUNAKAN NETWORK SIMULATOR (NS)	analisis jaring parameter randomize file size guna network simulator ns

Based on Table 8, the preprocessing process reduces Indonesian word variations into simpler root forms. For example, the word “PEMILIHAN” is reduced to “pilih”, “menggunakan” is reduced to “guna”, and “keputusan” is reduced to “putus”. At the same time, English technical terms such as “Promethee”, “Dijkstra”, “multifile locker application”, “data encryption standard”, and “network simulator” are preserved because these terms represent important concepts or methods in computer science. This strategy helps maintain the meaning of technical terms while reducing unnecessary variation in Indonesian word forms.

B. Model Resource Results

After the preprocessing stage, the cleaned research titles were used to generate model resources for similarity

calculation. The model resource generation process produced two main representations: TF-IDF-based representation and Word2Vec-based representation. These resources were stored as model files so that the system could perform similarity search without retraining the models for every query.

The TF-IDF training process generated a fitted TF-IDF vectorizer and a TF-IDF matrix. The vectorizer was used to transform new query titles into TF-IDF vectors, while the TF-IDF matrix stored the vector representation of all research titles in the dataset. Based on the generated resource, the TF-IDF matrix has a size of $10,000 \times 13,803$, which means that 10,000 research titles were represented using 13,803 lexical features. The matrix contains 114,485 non-zero values, indicating that the representation is sparse. This condition is expected because research titles are short texts and each title only contains a small portion of the total vocabulary.

The Word2Vec training process generated a CBOW-based Word2Vec model with a vector size of 100. The trained model contains 14,379 vocabulary terms learned from the preprocessed title corpus. After the Word2Vec model was generated, each research title was converted into a title vector by averaging the vectors of all tokens found in the Word2Vec vocabulary. The resulting Word2Vec title vector matrix has a size of $10,000 \times 100$, which means that each research title is represented as a 100-dimensional vector. No zero title vector was found, indicating that every title contains at least one token recognized by the trained Word2Vec model.

The generated metadata file contains 10,000 records with six main fields: id, author, title_original, title_preprocessed, token_count_original, and token_count_processed. This metadata is used to display the original title and supporting information after the similarity score is calculated. Table 9 shows the generated model resources, while Table 10 summarizes the training output obtained from the generated files. Figure 2 shows the complete model generation workflow.

Table 9. Generated Model Resources

Resource	Output File	Description
TF-IDF vectorizer	tfidf_vectorizer.pkl	Stores the fitted TF-IDF vectorizer used to transform new query titles into TF-IDF vector
TF-IDF Matrix	tfidf_matrix.pkl	Stores TF-IDF representations of all research titles in the dataset
Word2Vec model	Word2Vec.model	Stores the trained Word2Vec CBOW model used to generate word embeddings
Word2Vec title vectors	Word2Vec_title_vectors.npy	Stores original titles, preprocessed titles, author information and token counts

Table 10. Model Resource Summary

Item	Value
Number of Metadata	10,000
Number of TF-IDF features	13,803
TF-IDF matrix size	$10,000 \times 13,803$
TF-IDF non-zero values	114,485
TF-IDF matrix density	0,083%
Word2Vec architecture	CBOW
Word2Vec vocabulary size	14,379
Word2Vec vector size	100
Word2Vec title vector size	$10,000 \times 100$
Zero Word2Vec titles vectors	0

The TF-IDF matrix is sparse because each research title contains only a limited number of terms compared to the total vocabulary. This characteristic is common in short-text representation, where most feature values are zero. However, the sparse TF-IDF representation remains useful because it highlights important lexical terms that distinguish one title from another.

In contrast, the Word2Vec title vector representation is dense because each title is represented by a 100-dimensional vector. This representation supports contextual similarity calculation by capturing relationships between words learned from the title corpus. The absence of zero vectors indicates that all research titles contain at least one token that was recognized by the Word2Vec model. Therefore, both TF-IDF and Word2Vec resources are valid for use in the combined similarity calculation.

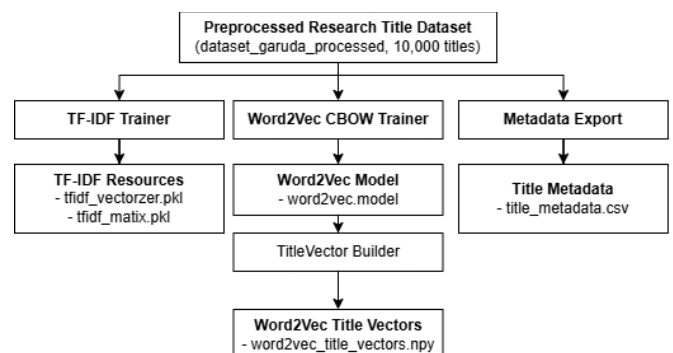


Figure 2. Model Resource Generation Workflow

C. Similarity Search Results

A similarity search example was conducted using the query “sistem informasi android” to illustrate how the proposed system ranks candidate research titles. The query was processed using the same preprocessing pipeline as the dataset. The system then calculated TF-IDF cosine similarity, Word2Vec cosine similarity, and the final combined score using an alpha value of 0.60.

The highest-ranked result was “DESAIN SISTEM INFORMASI SEKOLAH BERBASIS ANDROID,” with a final score of 0.720710. This result is reasonable because the candidate title contains the main terms from the query, namely “sistem,” “informasi,” and “android.” The result also shows that TF-IDF contributes strongly when the query and candidate

title share important lexical terms. Meanwhile, Word2Vec contributes contextual similarity by assigning high scores to titles that are semantically related even when exact term overlap is weaker.

This example indicates that the combined score can integrate lexical matching and contextual similarity in a single ranking mechanism. However, a single query example is not sufficient to demonstrate model performance. Therefore, a quantitative evaluation using 30 query titles and Top-5 retrieval results was conducted, as presented in the following subsection.

D. Quantitative Evaluation and Baseline Comparison

A quantitative evaluation was conducted to assess the retrieval performance of TF-IDF, Word2Vec, and the combined similarity score. The evaluation used 30 query titles representing topics in computer science and information technology. For each query, the system returned Top-5 candidate titles. The relevance of each query-title pair was manually assessed using a predefined 0–2 relevance rubric. The evaluation compared five retrieval configurations: TF-IDF only, Word2Vec only, combined alpha 0.50, combined alpha 0.60, and combined alpha 0.70.

Table 11 presents the quantitative evaluation results using Precision@5, MAP@5, and NDCG@5. TF-IDF only achieved the highest MAP@5 and NDCG@5 values, with MAP@5 of 0.985972 and NDCG@5 of 0.939095. This result indicates that lexical term matching remains highly important for short research title retrieval. Since research titles usually contain limited words, specific terms such as “Android”, “Laravel”, “Naive Bayes”, “K-Means”, “information system”, and “decision support system” strongly determine the relevance of retrieved titles.

Table 11. Quantitative Evaluation Results

Method	Queries	Precision@5	MAP@5	NDCG@5
TF-IDF	30	0.973333	0.985972	0.939095
Word2Vec	30	0.800000	0.888333	0.853949
Combined $\alpha = 0.50$	30	0.940000	0.972361	0.903253
Combined $\alpha = 0.60$	30	0.960000	0.981111	0.914284
Combined $\alpha = 0.70$	30	0.973333	0.981111	0.923408

Based on Table 11, Word2Vec only produced the lowest performance among the tested configurations, with Precision@5 of 0.800000, MAP@5 of 0.888333, and NDCG@5 of 0.853949. This result suggests that contextual similarity alone is not sufficient for short research title matching. Word2Vec can capture semantic relationships between terms, but it may assign high scores to titles that are contextually close while lacking important domain-specific keywords.

The combined configurations show that increasing the TF-IDF weight improves retrieval performance. Combined alpha 0.50 achieved Precision@5 of 0.940000, while alpha 0.60

increased the score to 0.960000. The best hybrid configuration was obtained by alpha 0.70, which achieved Precision@5 of 0.973333, MAP@5 of 0.981111, and NDCG@5 of 0.923408. This result indicates that the hybrid model performs best when lexical similarity is given a stronger weight than contextual similarity.

Although TF-IDF only achieved the highest MAP@5 and NDCG@5, the combined alpha 0.70 configuration produced competitive performance and outperformed Word2Vec only. Therefore, the proposed combined score should not be interpreted as a method that always surpasses TF-IDF, but as a lightweight hybrid approach that balances lexical and contextual similarity. This finding supports the use of TF-IDF-dominant hybrid weighting for short research title similarity detection, especially when the retrieved titles depend strongly on specific technical terms.

E. API and Android Prototype Testing

The similarity model was implemented using a Python FastAPI backend and a Flutter-based Android prototype. The backend provides two main endpoints, namely ‘/health’ to check service availability and ‘/search’ to process title similarity queries. The ‘/search’ endpoint receives three input parameters: ‘query’, ‘top_k’, and ‘alpha’. The output consists of ranked candidate titles with TF-IDF scores, Word2Vec scores, and final combined scores.

Functional testing was conducted on both the API and Android prototype. The API was tested using valid queries, empty queries, changes in ‘top_k’, and changes in ‘alpha’. The Android prototype was tested by entering valid titles, submitting empty input, displaying loading indicators, showing result cards, and handling failed API connections. All tested scenarios produced valid outputs. These results indicate that the backend and Android interface (Figure 3) can communicate properly and that the application can display ranked similarity results according to the selected query, Top-K value, and alpha configuration.

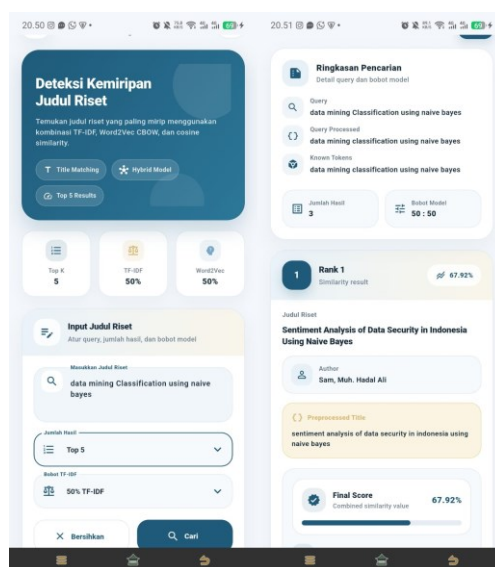


Figure 3. Android Prototype Interface: (A) Query Input and Alpha-Weight Setting; (B) Ranked Similarity Result Display.

F. Discussion

The quantitative evaluation shows that lexical similarity remains highly influential in short research title retrieval. TF-IDF achieved the highest MAP@5 and NDCG@5 values, indicating that important domain-specific terms strongly determine the relevance of retrieved titles. This finding is reasonable because research titles are short texts and usually contain limited but highly informative terms, such as method names, platforms, algorithms, and application domains. Therefore, TF-IDF is effective when the query and candidate titles share discriminative keywords.

These findings are consistent with previous comparative studies showing that TF-IDF can outperform Word2Vec on short review datasets because lexical cues remain highly influential in short-text classification and retrieval tasks [14], [15].

Word2Vec only produced the lowest performance among the tested configurations. This result indicates that contextual similarity alone is not sufficient for this task. Although Word2Vec can capture semantic relationships between words, it may assign high similarity to titles that are contextually related but do not contain important domain-specific terms. This condition is particularly relevant in short-text retrieval, where the absence of a specific term such as “Android,” “Laravel,” “Naive Bayes,” or “K-Means” can substantially change the relevance of a candidate title.

These findings are consistent with previous studies indicating that TF-IDF remains effective for short-text retrieval because it emphasizes discriminative lexical terms, while Word2Vec is useful for capturing contextual relationships between related words. However, unlike studies that report stronger gains from hybrid lexical-semantic representations in longer or more descriptive texts, the present results show that short research titles are still highly dependent on exact domain-specific terms. This explains why TF-IDF only achieved slightly higher MAP@5 and NDCG@5 than the hybrid configurations, while the hybrid model mainly improved performance over Word2Vec only. Therefore, the contribution of the proposed hybrid model should be interpreted as a lightweight and deployable lexical-semantic alternative rather than a universally superior similarity model.

The hybrid configurations show that increasing the TF-IDF weight improves retrieval performance. Combined alpha 0.70 achieved the best hybrid result and obtained the same Precision@5 as TF-IDF only. However, its MAP@5 and NDCG@5 values remained slightly lower than TF-IDF only. Therefore, the proposed combined score should not be interpreted as a method that consistently outperforms TF-IDF. Instead, it should be understood as a lightweight lexical-semantic approach that is competitive with TF-IDF and substantially better than Word2Vec only.

These findings imply that short research title similarity detection requires a careful balance between lexical and contextual representations. A TF-IDF-dominant hybrid configuration is more suitable for the dataset used in this study because title relevance is strongly influenced by exact technical terms. Nevertheless, Word2Vec remains useful as a contextual component because it can support similarity

estimation when the query and candidate titles do not fully share the same lexical surface forms. Future studies should evaluate whether transformer-based embeddings, multi-annotator relevance judgments, and adaptive alpha weighting can further improve retrieval quality.

Although this study justifies the use of Word2Vec CBOW for lightweight Android-based deployment, a direct comparison with transformer-based embeddings such as BERT or Sentence-BERT was not conducted. Therefore, the findings should be interpreted within the scope of lightweight lexical-semantic retrieval, and future work should include transformer-based baselines for a more comprehensive comparison.

IV. CONCLUSION

This study developed an Android-based research title similarity detection system using a combined TF-IDF and Word2Vec CBOW approach with cosine similarity. The system was designed to support research title checking by retrieving similar titles from a curated dataset of 10,000 computer science and information technology research titles collected from the GARUDA Portal. The model was implemented using a Python FastAPI backend and connected to a Flutter-based Android prototype, enabling users to submit a title query, select the number of returned results, adjust the TF-IDF weight, and view ranked similarity results.

The evaluation was conducted using 30 query titles and Top-5 retrieval results with manual relevance judgment based on a predefined 0–2 relevance rubric. The results show that TF-IDF only achieved the highest MAP@5 of 0.985972 and NDCG@5 of 0.939095, indicating that lexical term matching remains highly important for short research title retrieval. Word2Vec only produced the lowest performance, with Precision@5 of 0.800000, MAP@5 of 0.888333, and NDCG@5 of 0.853949. Among the hybrid configurations, combined alpha 0.70 achieved the best hybrid performance, with Precision@5 of 0.973333, MAP@5 of 0.981111, and NDCG@5 of 0.923408. These findings indicate that the combined model is competitive with TF-IDF and more effective than Word2Vec only, especially when the TF-IDF weight is dominant.

This study confirms that a lightweight lexical-semantic approach can be applied to Android-based research title similarity detection. However, several limitations remain. The dataset was limited to research titles from the GARUDA Portal, and the relevance assessment was conducted using a single-annotator rubric. In addition, this study did not directly compare the model with transformer-based embeddings such as BERT or Sentence-BERT. Future studies should involve multiple annotators, use larger and more diverse datasets, compare the proposed approach with transformer-based models, and explore adaptive alpha weighting to improve retrieval performance across different query types.

REFERENCES

- [1] S. Rahman, H. K. Shanto, U. A. Koana, and S. M. Danish, "Automated Research Article Classification and Recommendation Using NLP and Machine Learning," in *2025 3rd International Conference on Foundation and Large Language Models (FLLM)*, 2025, pp. 837–844, doi: 10.1109/FLLM67465.2025.11391003.
- [2] E. W. Pratomo and E. Utami, "Hybrid TF-IDF and Embedding Model for Improving Similarity and Clustering Accuracy," *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 10, no. 1, pp. 33–40, 2025, doi: 10.31328/jointecs.v10i1.7344.
- [3] Tukino, E. Sedyono, H. Hendry, A. Hananto, E. Novalia, and F. Nurapriani, "Evaluation of Word2Vec and FastText Models for Text Similarity Measurement Assessment," *BIS Information Technology and Computer Science*, vol. 3, article V326007, 2026, doi: 10.31603/bistycs.485.
- [4] H. Hendry, T. Tukino, E. Sedyono, A. Fauzi, and B. Huda, "HyEWCos: A Comparative Study of Hybrid Embedding and Weighting Techniques for Text Similarity in Short Subjective Educational Text," *Information*, vol. 16, no. 11, Art. no. 995, pp. 1–28, 2025, doi: 10.3390/info16110995.
- [5] W. Tanuwijaya, C. E. Setiawan, H. Irsyad, and A. Rahman, "Implementasi TF-IDF dan Cosine Similarity untuk Penyaringan Dokumen Berita Program Makan Siang Gratis Pemerintah Indonesia," *DEVICE: Journal of Information System, Computer Science and Information Technology*, vol. 6, no. 2, pp. 322–334, 2025, doi: 10.46576/device.v6i2.6724.
- [6] S.-V. Oprea, A. Băra, and M. P. Cristescu, "A Comprehensive Analysis of Text Similarity Metrics and Vectorization Techniques for Content-Based Product Recommendation," *IEEE Access*, vol. 14, pp. 58670–58689, 2026, doi: 10.1109/ACCESS.2026.3683598.
- [7] E. Aprianto, D. Mahdiana, and A. Wibowo, "Optimizing Bag of Words and Word2Vec with Vocabulary Pruning and TF-IDF Weighted Embeddings for Accurate Chatbot Responses in Indonesian Treasury Services," *Jurnal Teknik Informatika (JUTIF)*, vol. 7, no. 1, pp. 587–605, 2026, doi: 10.52436/1.jutif.2026.7.1.5370.
- [8] Sutriawan, S. Rustad, G. F. Shidik, and Pujiono, "Performance Evaluation of Text Embedding Models for Ambiguity Classification in Indonesian News Corpus: A Comparative Study of TF-IDF, Word2Vec, FastText BERT, and GPT," *Ingénierie des Systèmes d'Information*, vol. 30, no. 6, pp. 1469–1482, 2025, doi: 10.18280/isi.300606.
- [9] J. Lu, "Text vectorization in sentiment analysis: A comparative study of TF-IDF and Word2Vec from Amazon Fine Food Reviews," *ITM Web of Conferences*, vol. 70, article 03001, pp. 1–11, 2025, doi: 10.1051/itmconf/20257003001.
- [10] C. Neves-Moutinho and M. F. Arámburo-Castell, "Literature Summary Based on TF-IDF with Term-Term Correlation Matrix Analysis," *Research in Computing Science*, vol. 154, no. 10, pp. 99–109, 2025.
- [11] X. Feng, "Web Crawling Algorithm Fusing TF-IDF and Word2Vec Feature Extraction," *Journal of Web Engineering*, vol. 24, no. 5, pp. 713–738, 2025, doi: 10.13052/jwe1540-9589.2452.
- [12] L. A. Utami, H. Rachmi, and S. Hidayatulloh, "A Hybrid TF-IDF and Knowledge Graph-Enhanced Retrieval-Augmented Generation Framework with Large Language Models for Domain-Aware Question Answering," *Journal of Applied Data Sciences*, vol. 7, no. 2, pp. 866–881, 2026, doi: 10.47738/jads.v7i2.1136.
- [13] M. W. J. Rahmatullah and A. Purnama, "Applying NLP and Cosine Similarity in the Preliminary Selection Process of Recruitment Systems," *Bit-Tech (Binary Digital - Technology)*, vol. 8, no. 2, pp. 2565–2576, 2025, doi: 10.32877/bt.v8i2.3287.
- [14] F. Azzahra, B. Irawan, A. Faqih, D. Pratama, and D. A. Kurnia, "Comparison of TF-IDF and Word2Vec Feature Representations for Emotion Classification of Tokopedia E-Commerce Review Using LinearSVC," *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 5, no. 2, pp. 3466–3471, 2026, doi: 10.59934/jaiea.v5i2.2215.
- [15] A. G. Sinaga, Robet, and O. Pribadi, "Comprehensive Comparison of TF-IDF and Word2Vec in Product Sentiment Classification Using Machine Learning Models," *Journal of Applied Informatics and Computing (JAIC)*, vol. 10, no. 1, pp. 184–191, 2026, doi: 10.30871/jaic.v10i1.11582.

