Scientific Paper Recommendation System: Application of Sentence Transformers and Cosine Similarity Using arXiv Data

Ananda Pannadhika Putra[1]*, Desy Purnami Singgih Putri [2]*, AA.Kt.Agung Cahyawan Wiranatha [3]*
* Teknologi Informasi, Universitas Udayana
anandapanna@gmail.com [1], desysinggihputri@unud.ac.id [2], agung.cahyawan@unud.ac.id [3]

Article Info

Article history:

Received 2025-06-06
Revised 2025-07-01
Accepted 2025-07-09

Keyword:

Recommendation System,
Sentence Transformer,
Cosine Similarity,
arXiv,
Semantic Similarity.

ABSTRACT

Searching for relevant scientific literature faces complex challenges due to the proliferation of academic publications. This research develops a semantic similarity-based scientific paper recommendation system by utilizing Sentence Transformer (all-MiniLM-L6-v2 model) and cosine similarity algorithm on arXiv dataset (15,504 papers in Computer Science). The system is implemented as a Streamlit-based interactive web application that accepts user queries and recommends related papers based on semantic similarity. Performance evaluation using Precision, Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG) metrics showed that embedding text from the Introduction section without pre-processing yielded the best performance (NDCG: 0.7590; MAP: 0.6960; MRR: 0.7254), outperforming Abstract-based or text combination approaches. A user test of 45 respondents confirmed the effectiveness of the system: 95.5% expressed satisfaction with the relevance of the recommendations, and 93.3% confirmed a significant reduction in manual search time. The findings prove that retaining the raw text structure in the Introduction is optimal for semantic representation. Development suggestions include multidomain dataset expansion and transformer model optimization for accuracy improvement.

I. Introduction

Accurate and relevant literature search forms the foundation for producing high-quality academic work. As part of the scholarly process, researchers must understand prior research contributions to identify gaps and determine which literature is relevant to their study. However, this process often presents significant challenges [1]. The ever-increasing volume of scholarly papers has made it difficult for researchers to efficiently locate literature that is relevant, adequate, and fit for purpose [2].

Recommendation systems offer a promising solution to this problem [3]. This study proposes a recommendation system using the Sentence Transformer method and cosine similarity within a web application built with Streamlit. The application accepts user input in the form of a journal title and returns a list of scholarly journals with semantic similarity to the input. The proposed system employs Sentence Transformer to generate vector representations of text that capture rich semantic meaning. Sentence Transformer excels in generalizing models for diverse applications [4], particularly in its ability to generalize across textual domains. The vector representations are then analyzed using the cosine similarity algorithm to calculate similarity scores between journals [5]. By combining Sentence Transformer for embeddings and cosine similarity for measuring document similarity, these techniques demonstrate strong potential for building effective recommendation systems [2]. This approach allows the system to identify relationships between papers not solely based on shared keywords but also on contextual and research-purpose similarities.

This research aims to develop a scholarly journal recommendation system that relies on an existing journal dataset to generate user recommendations. The arXiv library repository was selected as the dataset source due to its notable advantages. arXiv hosts over 1.5 million articles across disciplines such as Computer Science, Physics, and Mathematics, making it a rich and relevant resource for academic research. Key benefits of arXiv include full-text availability, structured metadata, and citation networks between documents, all of which enhance recommendation accuracy [6].

By leveraging arXiv's data alongside Sentence Transformer and cosine similarity, the researchers anticipate enabling faster and more precise access to relevant literature.

II. Prosposed Method

A. Data Collection

Data collection was done through the arXiv website API. The focus category is Computer Science with sub-categories: cs.AI (Artificial Intelligence), cs.DB (Database), cs.SE (Software Engineering), cs.IR (Information Retrieval), cs.PL (Programming Languages), cs.CC (Computational Complexity), cs.GT (Game Theory), and cs.OH (Other Computer Science). A total of 16,000 initial documents were collected through the arXiv API. After cleaning duplicates and empty values, 15,504 valid journals were obtained. The dataset structure consists of columns: title, summary (abstract), categories, pdf_url, published (timestamp), and Introduction (introduction text). The text used for recommendations includes the title, abstract (summary), and introduction. The arXiv dataset was chosen because it provides structured metadata, full text, and citation networks, which support the accuracy of recommendation systems [6].

B. Data Cleaning and Pre-Processing

The dataset cleaning and pre-processing stage includes:

- Blank Value Removal
- Duplicate Removal
- Lowercasing and punctuation removal
- Tokenization and lemmatization using WordNetLemmatizer Lemmatization preserves semantic meaning, while tokenization breaks the text into the smallest units [7].
- English stopwords removal

Evaluation of the impact of lemmatization and stopword removal was done comparatively. The results show that the original text (without pre-processing) in the introduction section yields more optimal embedding performance, as it retains the full semantic context (see Figure 3).

C. Embeddings Creation

The Sentence Transformer all-MiniLM-L6-v2 model was used to convert text to numeric. Embeddings were generated for each combination of features (title, abstract, introduction) to evaluate the effect of input variation. The Sentence Transformer all-MiniLM-L6-v2 model was chosen because it balances accuracy and efficiency based on the study of Colangelo et al. The processing time (9 seconds) is faster than models such as MPNet (98 seconds), with the ability to generate rich semantic representations for long texts such as journal introductions [8]. The embedding results are stored in tensor form for cosine similarity calculation.

D. Cosine Similarity

Cosine similarity is an algorithm that is often applied as a method to measure the similarity of documents. Fundamentally, the calculation process utilizes a vector space similarity measure. This cosine similarity algorithm utilizes keywords in a document as a measure to calculate the level of similarity between documents represented in vector form [9]. The equation for the cosine similarity can be seen in equation 1.

$$\mathrm{CosSim}\ \alpha = \frac{\sum_{i=1}^{n} \mathrm{A}_i \mathrm{B}_i}{\sqrt{\sum_{i=1}^{n} \mathrm{A}_i^2} \cdot \sqrt{\sum_{i=1}^{n} \mathrm{B}_i^2}} \quad (1)$$

The formula above is the formula for cosine similarity. The following is an explanation of each variable used. A - B is the dot product result between vector A and vector B. D[10]ot product is calculated by summing the product of the corresponding elements of the two vectors. ||A|| is the norm (or length) of vector A, calculated using the square root of the sum of the squares of each element in vector A.

E. Ground Truth Clustering Approach

Clustering is an unsupervised machine learning method that analyses data sets without requiring target information or predefined outcomes. The purpose of clustering is to understand the structure of data and the relationship between its elements. Through clustering techniques, it will be possible to identify natural patterns and groupings in the data independently, without the guidance of values or predefined labels. The result of this process is a number of clusters that represent the characteristics of the data, such as groups of users with similar shopping habits or similar reading interests [11].

Relevant clusters are determined through the flow: (1) Combination of test and training data embedding, (2) Determination of the optimal number of clusters (K) by Canopy Algorithm (initial estimation) and Elbow Method (final validation), (3) Application of K-means. In this context, "relevant" journal means that the recommended journal is in the same cluster as the query journal

F. K-Means Clustering

K-Means Clustering is an unsupervised learning algorithm used to group data into K clusters based on similar features. The main objective is to minimize the within-cluster variance by optimizing the centroid position [12]

$$J(S, M) = \sum_{j=1}^{k} \sum_{x \in S_i} \left\| x_i - \mu_j \right\|^2 \quad (2)$$

The objective function $J(S, M)$, as defined in equation 2 [10], quantifies the total within-cluster variance in a partitioning clustering algorithm, such as K-means. Here, S={S1,S2,…,Sk} represents the set of *k* clusters, where each *Sj* is a subset of data points assigned to the *j*-th cluster. The set *M*={*μ*1,*μ*2,…,*μk*} corresponds to the centroids of these clusters, with *μj* being the mean of all data points in *Sj*.

The formula computes the sum of squared Euclidean distances between every data point xi*xi* and the centroid *μj* of the cluster to which xi*xi* belongs. The outer summation $\sum_{j=1}^{k}$ iterates over all *k* clusters, while the inner summation $\sum_{x \in S_i}$ aggregates the squared distances for every data point within a specific cluster *Sj*. The term $\|x_i - \mu_j\|^2$ denotes the squared Euclidean distance between a data point *xi* and its cluster centroid *μj*, serving as a measure of dispersion within the cluster. Minimizing *J*(*S*,*M*) optimizes the clustering configuration by ensuring data points are as close as possible to their respective centroids, thereby enhancing intra-cluster homogeneity. The number of clusters *k* is a predefined parameter, and the centroids *μj* are recalculated iteratively as the mean of all points in *Sj* during the clustering process. This formulation is fundamental to achieving compact and well-separated clusters in partitional clustering methods.

G. Canopy Algorithm

Canopy algorithm is a pre-processing method that is often used to improve the effectiveness and efficiency of the K-means algorithm in clustering, especially on large data. The combination of Canopy and K-means aims to overcome the weaknesses of K-means regarding the selection of initial cluster centers and the optimal number of clusters [13]. The goal of the Canopy algorithm is to provide an initial estimate of the number of clusters (K) that may be optimal. This estimate is then used to narrow down the range of K values to be tested.

There is no specific formula for applying Canopy algorithm, Canopy is used to perform a rough clustering by dividing the data into several “canopies” using two distance thresholds (T1 and T2). The result of Canopy is then used as the initial center and number of clusters for K-means, thus reducing the sensitivity of K-means to random initial center selection and speeding up the clustering process [13].

The use of Canopy as pre-processing in K-means is proven to speed up execution time and reduce computational burden, especially on large-scale text data [14]

H. Elbow Method

The elbow method is a popular technique used to determine the optimal number of clusters (k value) in the k-means clustering algorithm. The elbow method works by calculating the Sum of Squared Errors (SSE) value for various k values [15]. The SSE will decrease as the number of clusters increases, but the decrease will slow down at a certain point. The point at which the decrease in SSE starts to slow down significantly is referred to as the "elbow" and is considered the optimal number of clusters [16]. The equation for the elbow method can be seen in equation 2.

$$SSE = \sum_{i=1}^{k} \sum_{x \in c_i} dist\ (c_i, x)^2 \qquad (3)$$

where c_i denotes the centroid of the i-th cluster, and $dist\ (c_i, x)$ represents the Euclidean distance between a data point x and its corresponding centroid c_i. The outer summation $\sum_{i=1}^{k}$ iterates over all k clusters, while the inner summation $\sum_{x \in c_i}$ aggregates the squared distances of every data point x within cluster c_i. Squaring the distance emphasizes larger deviations and ensures non-negative values, thereby penalizing points farther from their centroids more heavily. A smaller SSE indicates higher intra-cluster homogeneity, as data points are positioned closer to their cluster centroids, reflecting well-formed and tightly packed clusters. The centroids cici are typically computed as the mean of all points in the cluster, and minimizing SSE is a central objective in iterative clustering processes to optimize partition quality. This metric is foundational for assessing cluster validity and guiding algorithm convergence toward compact groupings.

I. System Evaluation

The evaluation used a split of 80% training and 20% test data without cross-validation. The choice of single split is based on the large dataset size (15,504 documents). The evaluation metrics section in the recommendation system is used to measure the quality and performance of the system. The evaluation will assess the impact of lemmatization and stopword removal by comparing preprocessed data (with these steps applied) against unprocessed data (retaining original terms and stopwords). This comparative analysis is motivated by the recognition that lemmatization and stopword elimination may diminish the semantic context of textual content. Such a reduction could influence the performance of the Sentence Transformer model, given its reliance on comprehensive semantic comprehension to generate meaningful embeddings. The study aims to determine whether retaining raw linguistic features enhances or hinders the model’s ability to capture nuanced semantic relationships within the text. The system provides top-5 recommendations for each user query. The parameter top_k=5 was used consistently across the evaluation metrics that are used. Here are some of the main evaluation metrics that are often used in recommendation systems [17].

J. Precision

Precision in recommendation system evaluation metrics is the proportion of recommended items that are actually relevant to the user. Precision measures how accurate the system is in selecting suitable items from the set of recommendations generated. Precision is calculated by equation 4 [5].

$$Precision = \frac{Total\ Relevant\ Item}{Total\ Recommended\ Item} \qquad (4)$$

An example case of the calculation of the precision formula is if the system recommends 10 items (for example, movies, products, or music) to the user, and 7 of them are relevant. Then the number of relevant items = 7 and the total recommended items is 10, the value of 0.7 is obtained.

K. Mean Average Precision

MAP measures the quality of recommendations by checking whether relevant papers are listed in the top-k results or not. To calculate the Average Precision (AP) for a query paper, the average precision obtained after each Ground Truth Positive (GTP) is retrieved is taken, which is calculated by the equation in equation 2 [2].

$$AP@K = \frac{1}{GTP}\sum_{i=1}^{k}\frac{TPseen}{i} \qquad (5)$$

The above formula represents Average Precision at K (AP@K) with the explanation of the components in the formula as follows. AP@K is the Average Precision up to position K. It is the average precision of the relevant results found up to position K in the list of recommendation results. GTP is Ground Truth Positives, which is the total number of relevant items (in ground truth) for a particular query. In other words, it is the total items that are actually considered relevant. TPseen is the True Positives that have been found so far (up to position i). It refers to the number of relevant items that have been found in the result list at a particular position. i is the position in the result list (ranging from 1 to K). k is the upper limit of evaluated positions in the recommendation result list (usually the maximum number of K is specified).

After calculating AP@K in one query, the calculation can be done for the entire query and then the average AP@K value is calculated. The calculation will result in Mean Average Precision.

An example of calculating Mean Average Precision is when in the scientific paper recommendation system there are three users who search with different queries. For each query, the Average Precision (AP@5) value is calculated as follows: for Query 1, the AP@5 value is 0.756, for Query 2 is 0.840, and for Query 3 is 0.670.

By using the formula MAP@5 = (AP@5_1 + AP@5_2 + AP@5_3) / N, MAP@5 = (0.756 + 0.840 + 0.670) / 3 = 0.755. This result shows that on average, the recommendation system is able to provide relevant papers in a good order for various user queries.

L. Mean Reciprocal Rank (MRR)

Mean reciprocal rank calculates the average of the reciprocal rank of the first relevant item across queries. Reciprocal rank is the inverse of the position (rank) of the first relevant item in the result list.

$$MPP = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \qquad (6)$$

As The equation 6 is to calculate the mean reciprocal rank. The components of MRR are as follows. N The total number of queries evaluated. 〖 rank〗 _i is the position (rank) of the first relevant item in the result list for the i-th query. Reciprocal rank for the i-th query is calculated as$\frac{1}{rank_i}$.

An example of calculating the mean reciprocal rank is when a scientific paper recommendation system has three users searching with different queries. For each query, the system recommends a list of papers, and it is found that the first rank where relevant papers appear is position 1 for Query 1, 3 for Query 2, and 2 for Query 3.

Using the formula MRR = (1/N) ∑ (1/rank_i), the MRR value = (1/3) * (1/1 + 1/3 + 1/2) = 0.611. This value indicates that on average, the system can display relevant papers in a fairly high order in the recommendation list. The higher the MRR value, the faster the system can provide useful information to users.

M. Normalised Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain (DCG) measures the quality of the result order generated by the system based on the relevance of the recommended items to the user's preferences by comparing the ratio between actual DCG and IDCG. Discounted Cumulative Gain (DCG) DCG takes into account the position of the item in the result list. Relevance at the initial (top) position is given a higher weight than those at lower positions, with a logarithmic function to lower the weight. The formula is shown in equation 7.

$$DCG = \sum_{i=1}^{N}\frac{rel_i}{log_i(i+1)} \qquad (7)$$

Equation 7 is for obtaining the Discounted Cumulative Gain $\sum_{i=1}^{N}$ The sigma symbol (∑) indicates that this is the summation of the relevance (gain) values from position 1 to n, where n is the number of items in the result list. rally is the relevance of the i-th item in the result list. Relevance values are usually assigned based on the degree of relevance to the user's query. For example, a value of 3 means highly relevant, 2 relevant, and 0 irrelevant.

Meanwhile, Normalized Discounted Cumulative Gain (NDCG) can be calculated using equation 8.

$$NDCG = \frac{DCG}{IDCG} \qquad (8)$$

Ideal DCG (IDCG) is the DCG value for the ideal result list, where the most relevant item is placed in the top position. It is used as a comparison to measure how close the actual results are to the ideal [18].

An example of the application of NDCG is when the recommendation system provides recommendations for five papers with a user relevance score for the system recommendations of (3, 2, 3, 0, 1) as shown in the equation 9.

$$DCG = 3 + \frac{2}{log_2(3)} + \frac{3}{log_2(4)} + \frac{0}{log_2(5)} + \frac{1}{log_2(6)} \qquad (9)$$

To assess the effectiveness of the recommendation order, the Discounted Cumulative Gain (DCG) is first calculated by summing up the relevance of each paper, where higher relevance at the initial position is given more weight.

$$IDCG = 3 + \frac{3}{log_2(3)} + \frac{2}{log_2(4)} + \frac{1}{log_2(5)} + \frac{0}{log_2(6)} \qquad (10)$$

The result of DCG calculation is 6.148. Then, the Ideal DCG (IDCG) is calculated by optimally sorting the relevance list into (3, 3, 2, 1, 0), resulting in a value of 7.014.

$$NDCG = \frac{6.148}{7.014} = 0.876 \qquad (11)$$

Dividing DCG by IDCG gives NDCG = 0.876, indicating that the recommender system does a very good job of ranking relevant papers. Since NDCG takes into account both relevance and position in the recommendation list, this metric is particularly useful in academic search systems, where users tend to pay more attention to the top results in the recommendation list.

N. Cronbach's Alpha

Cronbach's alpha is a coefficient used to assess the internal consistency of a measurement instrument, in the case of this research is a questionnaire, with a range of values between 0 and 1. This concept was developed by Cronbach (1951) as an indicator of the extent to which the items in the instrument are interrelated and measure the same construct or concept [19]. A high alpha value indicates that the items are strongly related in measuring the intended construct, while a low value indicates inconsistency or heterogeneity of items. However, alpha does not only depend on the correlation between items, but is also affected by the length of the test; the addition of relevant items can increase the alpha value, even if the test is not completely homogeneous. This coefficient operates under the assumption of the tau-equivalent model, which requires that each item measures the same latent construct with an equivalent measurement scale. If this assumption is violated-for example, due to construct multidimensionality-alpha tends to underestimate true reliability. In addition, alpha is sample-dependent, so it needs to be recalculated each time the instrument is used in a different population [20].

Generally accepted alpha values are in the range of 0.70 to 0.90. Values below 0.70 may be caused by too few items, weak inter-item correlations, or the presence of multidimensionality in the test. Conversely, values above 0.90 may reflect item redundancy, where multiple questions measure the same thing repeatedly. It is important to note that alpha does not guarantee the unidimensionality of a test. Despite a high alpha, the instrument may still contain multiple dimensions, thus requiring additional analysis such as factor analysis to verify the construct structure. Thus, Cronbach's alpha serves as an initial tool for assessing reliability, but its interpretation should be accompanied by a deep understanding of the measurement context and its methodological limitations [20]

$$\alpha = \left(\frac{k}{(k-1)}\right) * \left(1 - \left(\frac{\sum_{i-1}^{k}\sigma_{Y_i}^2}{\sigma_X^2}\right)\right) \qquad (11)$$

Equation 11 is the equation used in calculating Cronbach's alpha [19]. k is the number of items in the instrument, $\sum_{i-1}^{k}\sigma_{Y_i}^2$ is the total variance of each individual item, and σ_X^2 is the total variance of the overall instrument score. The $\frac{k}{(k-1)}$ component serves as a correction factor that takes into account the number of items, where the more items (k), the smaller the influence of this factor. Meanwhile, the ratio $\frac{\sum_{i-1}^{k}\sigma_{Y_i}^2}{\sigma_X^2}$ reflects the proportion of individual item variance to total variance. If the items are strongly interrelated, the total variance will be much larger than the sum of the individual variances, so this ratio approaches zero and the alpha value increases.

O. Web App Implementation

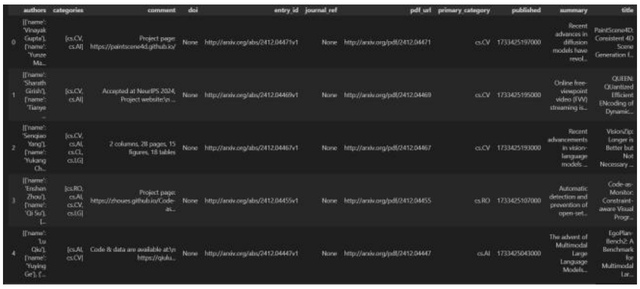
The application was built with Streamlit to accept user input, process recommendations, and display results in an interactive interface. User input is free text (example: 'Deep Learning For Medic'). Additional filters include: (1) range of publication years (slider), (2) choice of embedding source (title, abstract, introduction, or combination thereof). User interaction flow:

- Enter a text query in the search box.
- Set the year filter (optional).
- Select the embedding source.
- Click 'Search Journal'.

The system displays 5 journal recommendations with: title (PDF link), year, category, similarity score and abstract.

III. RESULT AND DISCUSSION

Data collection through the arXiv website API managed to collect as many as 16,000 journals. However, in the process of downloading journals, not all documents can be downloaded. This can be caused by the availability of documents on the API, the quality of the documents or there is an error in the journal download link. After the download process was complete, a total of 15,504 journals were successfully downloaded.

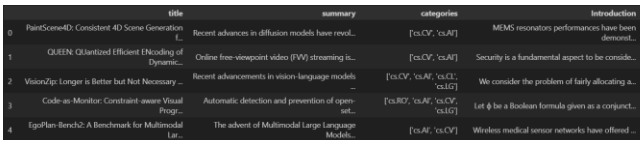


	authors	categories	comment	doi	entry_id	journal_ref	pdf_url	primary_category	published	summary	title
0	[{'name': 'Vinayak Gupta'}, {'name': 'Yunze Ma...	[cs.CV, cs.AI]	Project page: https://paintscene4d.github.io/	None	http://arxiv.org/abs/2412.04471v1	None	http://arxiv.org/pdf/2412.04471	cs.CV	1733425197000	Recent advances in diffusion models have revol...	PaintScene4D: Consistent 4D Scene Generation f...
1	[{'name': 'Sharath Girish'}, {'name': 'Tianye ...	[cs.CV, cs.AI]	Accepted at NeurIPS 2024, Project website:\n ...	None	http://arxiv.org/abs/2412.04469v1	None	http://arxiv.org/pdf/2412.04469	cs.CV	1733425195000	Online free-viewpoint video (FVV) streaming is...	QUEEN: QUantized Efficient ENcoding of Dynamic...
2	[{'name': 'Senqiao Yang'}, {'name': 'Yukang Ch...	[cs.CV, cs.AI, cs.CL, cs.LG]	2 columns, 28 pages, 15 figures, 18 tables	None	http://arxiv.org/abs/2412.04467v1	None	http://arxiv.org/pdf/2412.04467	cs.CV	1733425193000	Recent advancements in vision-language models ...	VisionZip: Longer is Better but Not Necessary ...
3	[{'name': 'Enshen Zhou'}, {'name': 'Qi Su'}, {...	[cs.RO, cs.AI, cs.CV, cs.LG]	Project page: https://zhoues.github.io/Code-as...	None	http://arxiv.org/abs/2412.04455v1	None	http://arxiv.org/pdf/2412.04455	cs.RO	1733425107000	Automatic detection and prevention of open-set...	Code-as-Monitor: Constraint-aware Visual Progr...
4	[{'name': 'Lu Qiu'}, {'name': 'Yuying Ge'}, {...	[cs.AI, cs.CV]	Code & data are available at:\n https://qiulu...	None	http://arxiv.org/abs/2412.04447v1	None	http://arxiv.org/pdf/2412.04447	cs.AI	1733425043000	The advent of Multimodal Large Language Models...	EgoPlan-Bench2: A Benchmark for Multimodal Lar...

Figure 1. Data Collection Result

Figure 1 shows a table containing data collected from scientific journals retrieved from the arXiv website. Each row in the table represents an article, with columns providing information about the article. There are several crucial columns that will be used in this research. The Authors column lists the names of the authors of the article, while the Categories column indicates the relevant field of science, such as Computer Vision (cs.CV) or Artificial Intelligence (cs.AI). The PDF URL column provides a link to access the PDF version of the article. The Summary column provides a short abstract that explains the core of the article content.

With the available data, the introduction section of each journal will be retrieved. First, we will download the journal obtained through the link in the PDF URL column. Through the downloaded journal, part of the introduction will be obtained through the application of regular expression.

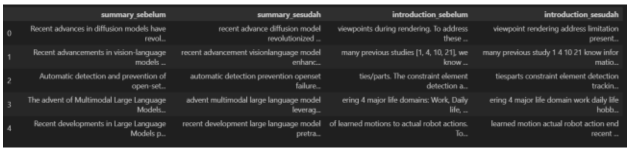


	title	summary	categories	Introduction
0	PaintScene4D: Consistent 4D Scene Generation f...	Recent advances in diffusion models have revol...	['cs.CV', 'cs.AI']	MEMS resonators performances have been demonst...
1	QUEEN: QUantized Efficient ENcoding of Dynamic...	Online free-viewpoint video (FVV) streaming is...	['cs.CV', 'cs.AI']	Security is a fundamental aspect to be conside...
2	VisionZip: Longer is Better but Not Necessary ...	Recent advancements in vision-language models ...	['cs.CV', 'cs.AI', 'cs.CL', 'cs.LG']	We consider the problem of fairly allocating a...
3	Code-as-Monitor: Constraint-aware Visual Progr...	Automatic detection and prevention of open-set...	['cs.RO', 'cs.AI', 'cs.CV', 'cs.LG']	Let ϕ be a Boolean formula given as a conjunct...
4	EgoPlan-Bench2: A Benchmark for Multimodal Lar...	The advent of Multimodal Large Language Models...	['cs.AI', 'cs.CV']	Wireless medical sensor networks have offered ...

Figure 2. Journals With Introduction

Figure 2 is a table that contains the introduction section of the journal. The table contains columns named title, summary, categories, and introduction.

The process of data cleaning and pre-processing consist of blank value removal, duplicate removal, lowercasing and punctuation removal, tokenization, lemmatization, and english stopwords removal.



	summary_sebelum	summary_sesudah	introduction_sebelum	introduction_sesudah
0	Recent advances in diffusion models have revol...	recent advance diffusion model revolutionized ...	viewpoints during rendering. To address these ...	viewpoint rendering address limitation present...
1	Recent advancements in vision-language models ...	recent advancement visionlanguage model enhanc...	many previous studies [1, 4, 10, 21], we know ...	many previous study 1 4 10 21 know infor matio...
2	Automatic detection and prevention of open-set...	automatic detection prevention openset failure...	ties/parts. The constraint element detection a...	tiesparts constraint element detection trackin...
3	The advent of Multimodal Large Language Models...	advent multimodal large language model leverag...	ering 4 major life domains: Work, Daily life, ...	ering 4 major life domain work daily life hobb...
4	Recent developments in Large Language Models p...	recent development large language model pretra...	of learned motions to actual robot actions. To...	learned motion actual robot action end recent ...

Figure 3. Difference Before and After Pre-Processing

Figure 3 shows the differences in the data before and after the lemmatization and stopwords process. The difference can be seen in some of the removal of stopwords in paragraphs and word changes in the lemmatization process.

Following the cleaning and preprocessing of the dataset, an embedding process will be implemented using the Sentence Transformer model "all-MiniLM-L6-v2".

TABLE I
TESTING RESULTS BEFORE LEMMATIZATION AND STOPWORDS PROCESSES

Embeddings	Optimal K	Cluster Precision	NDCG	MAP	MRR
Title	904	0.4813	0.7206	0.6573	0.6861
Abstract No Pre-processing	904	0.4813	0.7316	0.6629	0.6947
Intro No Pre-processing	904	0.5221	0.7590	0.6960	0.7254
Combined No Pre-processing	902	0.5044	0.7564	0.6905	0.7194

Table 1 is the result of testing the dataset before lemmatization and stopwords. The results show that the introduction embeddings before pre-processing are superior compared to other pre-processed datasets. Comparing these results with the results of research conducted by Bereczki & Girdzijauskas [21] shows that embeddings-based recommendation systems are superior to Graph Neural Networks (GNN) methods that combine user interaction data (implicit) and article content. Another comparison to research by Ali et al [2], shows that the embeddings-based recommendation system is superior to the heterogeneous network embedings method. The research by Ali et al. focuses on the integration of heterogeneous networks (author, venue, label, topic, and journal relationships) to represent the dynamics of researcher preferences. This focus allows the semantic representation of the text to be less than optimal due to the focus on the network structure.

Evaluation using top-5 recommendations showed the best performance on Introduction without pre-processing:

- NDCG: 0.7590
- MAP: 0.6960
- MRR: 0.7254
- Cluster-based Precision: 0.5221

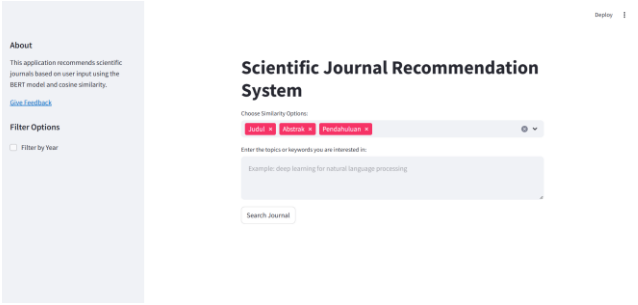
This value outperforms both GNN-based and heterogeneous network embedding approaches.

TABLE II
TESTING RESULTS AFTER LEMMATIZATION AND STOPWORDS PROCESSES

Embeddings	**Optimal K**	**Cluster Precision**	**NDCG**	**MAP**	**MRR**
Abstract Pre-processing	917	0.4615	0.7269	0.6577	0.6906
Intro Pre-processing	870	0.4804	0.7292	0.6636	0.6937
Combiuned Pre-processing	886	0.5055	0.7486	0.6812	0.7152

Table 2 is the result of testing the dataset after lemmatization and stopwords processing. Based on the evaluation data, the recommendation system with pre-processing and without pre-processing shows performance differences. In the Optimal K metric, the method with pre-processing produces more varied values (870-917), while without pre-processing tends to stabilize in the range of 902-904. This indicates that pre-processing affects the granularity of the clusters, although the difference is not very significant. However, on the Cluster Precision metric, the method without pre-processing actually excels in some cases. Intro No Pre-processing achieved the highest precision (0.5221), higher than Combined Pre-processing (0.5055).

The performance of the NDCG, MAP, and MRR evaluation metrics also showed a similar pattern. In all three metrics, Intro No Pre-processing recorded the highest values (NDCG: 0.7590; MAP: 0.6960; MRR: 0.7254), superior to all methods with pre-processing. This suggests that the original text structure (without pre-processing) in Intro is more effective in representing document relevance, both in terms of result placement and ranking accuracy. Overall, the data from the evaluation metrics show that pre-processing is not necessary in the implementation of embeddings-based recommendation systems.



This application recommends scientific journals based on user input using the BERT model and cosine similarity.
Give Feedback
Filter Options
□ Filter by Year
Scientific Journal Recommendation System
Choose Similarity Options:
Judul ×
Abstrak ×
Pendahuluan ×
Enter the topics or keywords you are interested in:
Example: deep learning for natural language processing
Search Journal

Figure 4. Streamlit Interface

This journal search web app in figure 4 provides a number of features designed to make it easier for users to find relevant scientific references. First, users can conduct searches based on specific topics, ideas, or keywords through the text boxes provided, such as “Deep Learning For Medic” or “Natural Language Processing”. In addition, there is a filter based on publication year that allows users to set a specific year range using a slider, thus narrowing the search results according to temporal needs. A similarity criteria feature is also provided to select journal sections that are considered crucial in matching results, such as title, abstract, introduction, or a combination of several sections. After specifying the search parameters and pressing the “Search Journal” button, the system will display a list of the most suitable journals. Each recommendation comes with the journal title linked directly to the PDF document, year of publication, category, as well as a summary abstract for easy initial evaluation. Thus, the app offers an integration of search flexibility, data filtering, and easy access to scientific reference sources.

Application trials are carried out to ensure that the application developed is in accordance with the needs and expectations of users. A total of 45 students from the Udayana University Information Technology Study Program have tried and assessed the application. The assessment was conducted on the quality of journal recommendations, ease of use and system interface, system speed and efficiency, and technical evaluation and system accuracy. The research focuses on the quality of the journal recommendation, the technical evaluation and system accuracy.

The assessment of the quality of journal recommendations contains 5 questions for users to assess the quality of journal recommendations provided by the system. The questions cover the relevance of the recommendation results, the suitability of the recommendation results to user needs, the consistency of the quality of the recommendations provided, the completeness of the dataset, and the quality comparison with manual searches.

TABLE III
RELIABILITY TEST OF JOURNAL RECOMMENDATION QUALITY

Questions	*Cronbach's Alpha* if Item Deleted	Reliability (>=0.7)
Q1	0.711	Yes
Q2	0.740	Yes
Q3	0.742	Yes
Q4	0.809	Yes
A5	0.726	Yes

Table 3 above presents the results of the reliability test of the journal recommendation quality questions using Cronbach's Alpha to assess internal consistency between items. All items (Q1 to Q5) met the reliability criteria with Cronbach's Alpha values ≥0.7 when each item was removed, indicating that the scale is reliable overall. The highest value was Q4 (0.809), indicating that this item contributed the most to the consistency of the scale, while the lowest value was Q1 (0.711)-still above the minimum required. The “Reliability (≥0.7)” column with “Yes” marked on all items confirms that no items need to be removed, as removal of items does not reduce the reliability of the scale. The high Cronbach's Alpha values on all items indicate that the questions are measured consistently, so the scale is valid for use in further research or analysis.

The assessment of the technical evaluation and accuracy of the system was carried out through a questionnaire containing 5 questions aimed at evaluating the technical aspects and accuracy of the recommendations generated by the system. This was to validate the match between the system's technical capabilities and the user's expectations. The questions covered the ability of the system to handle various text inputs, evaluation of the information presented, evaluation of the ability to handle keywords in the search, comparison with other systems, and the benefits perceived by the users towards the system.

TABEL IV
RELIABILITY TEST OF TECHNICAL EVALUATION AND SYSTEM ACCURACY

Questions	*Cronbach's Alpha* if Item Deleted	Reliabilitas (>=0.7)
Q1	0.755	Yes
Q2	0.771	Yes
Q3	0.820	Yes
Q4	0.741	Yes
Q5	0.777	Yes

The table presents the results of the technical evaluation reliability and system accuracy tests using Cronbach's Alpha to assess internal consistency between items. All items met the reliability criteria with Cronbach's Alpha values ≥0.7, even when one item was deleted. The highest value was found in Q3 (0.820), while the lowest value was Q4 (0.741), which remained above the minimum reliability threshold. This shows that all items consistently measure the same construct without needing to be removed, as item removal does not reduce the reliability of the scale. This high level of consistency confirms that Variable D has good reliability and can be relied upon for further analysis, as each item supports each other in representing the dimension being measured. Thus, this scale is valid for use in research or evaluation related to the observed construct.

IV. CONCLUSION

The research findings demonstrate the successful implementation and evaluation of a semantic-aware recommendation system leveraging the Sentence Transformer model all-MiniLM-L6-v2 and cosine similarity, trained on the arXiv dataset. By generating semantically rich text embeddings, the system transcends keyword-based matching to incorporate contextual and thematic nuances, thereby aligning recommendations with the intent and scope of research inquiries. Performance evaluations across key metrics—including Precision (0.48–0.52), Mean Average Precision (MAP: 0.66–0.69), Mean Reciprocal Rank (MRR: 0.69–0.72), and Normalized Discounted Cumulative Gain (NDCG: 0.72–0.75)—indicate robust accuracy in identifying relevant scholarly papers. These results surpass conventional approaches such as collaborative filtering and Word2Vec-based models, underscoring the efficacy of semantic embedding techniques in academic recommendation tasks. Furthermore, the deployment of a Streamlit-based web application featuring an intuitive interface yielded high user satisfaction, as evidenced by testing with 45 participants. Users rated the system favourably (average scores of 4–5/5) for usability, recommendation speed, and result relevance, validating its practical utility in real-world research workflows. Some limitations of the system:

- Domain bias: Only uses Computer Science data from arXiv.
- Only include extracted papers in the system. Not included new papers.
- ArXiv metadata dependency: PDF text structure inconsistency could affect preliminary extraction.

To further strengthen the methodology and extend the research impact, several recommendations are proposed. First, expanding the dataset beyond the current Computer Science-focused arXiv subset to include multi-domain sources (e.g., PubMed, IEEE Xplore) could enhance the generalizability of the system. Second, while the *all-MiniLM-L6-v2* model demonstrated strong performance, experimenting with alternative architectures such as MPNet or BERT-large—despite potential computational trade-offs—may yield higher accuracy. Finally, supplementing evaluation metrics like F1-Score or Diversity Score could provide deeper insights into recommendation quality, particularly in balancing relevance and diversity. Collectively, the study highlights the potential of advanced semantic modelling to enhance scholarly recommendation systems while balancing technical precision and user-centric design.

REFERENCES

[1] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, Nov. 2016, doi: 10.1007/s00799-015-0156-0.

[2] Z. Ali, G. Qi, K. Muhammad, B. Ali, and W. A. Abro, "Paper recommendation based on heterogeneous network embedding," *Knowl Based Syst*, vol. 210, Dec. 2020, doi: 10.1016/j.knosys.2020.106438.

[3] C. K. Kreutz and R. Schenkel, "Scientific paper recommendation systems: a literature review of recent publications," *International Journal on Digital Libraries*, vol. 23, no. 4, pp. 335–369, Dec. 2022, doi: 10.1007/s00799-022-00339-w.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04805

[5] X. Kong, M. Mao, W. Wang, J. Liu, and B. Xu, "VOPRec: Vector Representation Learning of Papers with Text Information and Structural Identity for Recommendation," *IEEE Trans Emerg Top Comput*, vol. 9, no. 1, pp. 226–237, Jan. 2021, doi: 10.1109/TETC.2018.2830698.

[6] C. B. Clement, M. Bierbaum, K. P. O'Keeffe, and A. A. Alemi, "On the Use of ArXiv as a Dataset," Apr. 2019, [Online]. Available: http://arxiv.org/abs/1905.00075

[7] S. J. Mielke *et al.*, "Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP," Dec. 2021, [Online]. Available: http://arxiv.org/abs/2112.10508

[8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," 2019. [Online]. Available: https://github.com/UKPLab/

[9] F. A. Nugroho, F. Septian, D. A. Pungkastyo, and J. Riyanto, "Penerapan Algoritma Cosine Similarity untuk Deteksi Kesamaan Konten pada Sistem Informasi Penelitian dan Pengabdian Kepada Masyarakat," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 4, p. 529, Dec. 2021, doi: 10.32493/informatika.v5i4.7126.

[10] Douglas. Steinley, "K-means clustering: A half-century synthesis," *British Journal of Mathematical and Statistical Psychology*, vol. 59, no. 1, pp. 1–34, May 2006, doi: https://doi.org/10.1348/000711005X48266.

[11] Y. Gulzar, A. A. Alwan, R. M. Abdullah, A. Z. Abualkishik, and M. Oumrani, "OCA: Ordered Clustering-Based Algorithm for E-Commerce Recommendation System," *Sustainability (Switzerland)*, vol. 15, no. 4, Feb. 2023, doi: 10.3390/su15042947.

[12] S. M. Miraftabzadeh, C. G. Colombo, M. Longo, and F. Foiadelli, "K-Means and Alternative Clustering Methods in Modern Power Systems," 2023, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2023.3327640.

[13] C. Wang, "Frontiers in Computing and Intelligent Systems Pattern Classification of Stock Price Moving," 2022.

[14] J.-W. Z. Jun-Wu Zhai, Y.-C. T. Jun-Wu Zhai, W.-T. L. Yu-Chen Tian, and K. L. Wen-Tao Li, "Canopy-MMD Text Clustering Algorithm Based on Simulated Annealing and Canopy Optimization," 電腦學刊, vol. 34, no. 1, pp. 075–086, Feb. 2023, doi: 10.53106/199115992023023401006.

[15] H. Humaira and R. Rasyidah, "Determining The Appropiate Cluster Number Using Elbow Method for K-Means Algorithm," European Alliance for Innovation n.o., Mar. 2020. doi: 10.4108/eai.24-1-2018.2292388.

[16] H. Zhao, "Design and Implementation of an Improved K-Means Clustering Algorithm," *Mobile Information Systems*, vol. 2022, 2022, doi: 10.1155/2022/6041484.

[17] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Applied Sciences (Switzerland)*, vol. 10, no. 21, pp. 1–20, Nov. 2020, doi: 10.3390/app10217748.

[18] O. Jeunen, I. Potapov, and A. Ustimenko, "On (Normalised) Discounted Cumulative Gain as an Off-Policy Evaluation Metric for Top-n Recommendation," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2024, pp. 1222–1233. doi: 10.1145/3637528.3671687.

[19] L. J. Cronbach, "COEFFICIENT ALPHA AND THE INTERNAL STRUCTURE OF TESTS*," 1951.

[20] M. Tavakol and R. Dennick, "Making sense of Cronbach's alpha," Jun. 27, 2011. doi: 10.5116/ijme.4dfb.8dfd.

[21] M. Bereczki and S. Girdzijauskas, "Graph Neural Networks for Article Recommendation based on Implicit User Feedback and Content," 2021.