International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 11, November, 2025, pp. 3003-3011
Journal homepage: http://iieta.org/journals/isi

Transformer-Based Semantic Search Engine with Morphophonemic Stemming for Low-Resource Sundanese Language

Aries Maesya[1*], Yulyani Arifin[1], Amalia Zahra[1], Widodo Budiharto[2]

[1] Computer Science Department, BINUS Graduate Program-Doctor of Computer Science Program, Bina Nusantara University, Jakarta 11480, Indonesia
[2] Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Corresponding Author Email: aries.maesya@binus.ac.id

https://doi.org/10.18280/isi.301117

Received: 9 October 2025
Revised: 19 November 2025
Accepted: 27 November 2025
Available online: 30 November 2025

Keywords:
Sundanese language, information retrieval, transformers, stemming, semantic search

ABSTRACT

An effective information retrieval system is crucial for accessing information. However, existing state-of-the-art models, particularly those based on the Transformer architecture, primarily focus on high-resource languages such as English. This creates a significant challenge for low-resource languages such as Sundanese, which remains understudied. To address this research gap, we propose a comprehensive web-based Sundanese language-processing toolbox integrated with a Transformer-based information-retrieval framework, which serves as a standardized baseline for evaluating low-resource Sundanese retrieval tasks. The web application consists of two core components: a stemming module and a semantic search engine. For the stemming module, we employed the AMStemming algorithm, which applies morphophonemic rules and is currently the state-of-the-art approach for Sundanese. The stemming module demonstrated strong computational performance, with latency below 400 ms and linear scalability with increasing numbers of concurrent users. The semantic search engine was implemented using a Transformer-based model fine-tuned from pre-trained IndoBERT weights. The model achieved an mAP@5 of 0.2883, a Precision@5 of 0.0780, and a Recall@5 of 0.3899, providing a quantitative baseline for low-resource Sundanese retrieval. A user experience (UX) evaluation involving 100 participants confirmed the system's usability and clarity. High satisfaction was reported across five evaluation dimensions, with clarity (88.83%) and relevance (88.71%) receiving the strongest ratings. These results demonstrate that the proposed system provides a practical and scalable foundation for future research on Transformer-based semantic search in low-resource languages.

1. INTRODUCTION

Recent advances in large language models (LLMs) have significantly expanded the scope of human–computer interaction. LLMs have been applied to a wide range of tasks, including conversational agents such as ChatGPT, which has played an essential role in improving user productivity [1]. Unlike traditional chatbots, modern systems can generate highly natural, context-aware responses owing to the capabilities of Transformer-based architectures [2]. These models are typically trained on large-scale, multi-domain corpora, enabling them to answer a wide variety of queries [3]. However, their knowledge is limited to the data used during training, which makes the integration of external or updated information essential for practical deployment.

Retrieval-Augmented Generation (RAG), first introduced in 2020 [4], is a widely adopted approach for addressing this limitation. In RAG-based systems, an information-retrieval pipeline is applied before text generation to incorporate relevant external knowledge into the LLM. The effectiveness of such systems strongly depends on the quality of the retrieval component, which must identify pertinent documents that can serve as contextual input for the generation process. Therefore, reliable and accurate information-retrieval methods are a critical foundation of modern RAG-based applications.

Natural Language Processing (NLP) methods for information retrieval have evolved substantially with the development of Transformer architectures. Earlier approaches relied primarily on sparse lexical representations such as Bag-of-Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF) [5]. More recent methods employ dense vector representations that capture semantic relationships, including Word2Vec [6], GloVe [7], and FastText [8]. These techniques represent text in continuous vector spaces and have become a foundation for modern semantic retrieval models [9]. In both sparse and dense approaches, text preprocessing, particularly stemming, plays a critical role. Stemming removes affixes to normalize word forms, allowing semantically related words to be mapped to a standard representation [10]. This process reduces vocabulary size and improves the quality of text representations used for retrieval.

Transformer-based models provide a robust framework for encoding semantic information and have become the backbone of modern language models [11-13]. These models are typically trained through a pre-training phase using masked language modeling, followed by fine-tuning for downstream tasks such as information retrieval [14]. In retrieval systems, the learned text representations are used to embed documents and queries into a shared vector space, enabling semantic matching beyond exact lexical overlap.

Despite their success in high-resource languages, most publicly available Transformer-based semantic search models have been trained primarily on English corpora. Although multilingual models exist, their performance for truly low-resource languages remains limited, including for Sundanese [15, 16]. Sundanese is one of Indonesia's major local languages, yet it remains significantly under-represented in large-scale linguistic datasets. The scarcity of high-quality Sundanese corpora poses a significant challenge for training and fine-tuning Transformer-based retrieval models, while simultaneously motivating the exploration of methods that can operate effectively under low-resource conditions.

In this study, we leverage recent advances in Sundanese natural language processing to develop a comprehensive web-based language-processing system designed for information-retrieval research. The proposed system focuses on two core components: a morphophonemic stemming module and a Transformer-based semantic search engine, providing an integrated experimental platform for evaluating low-resource Sundanese retrieval.

2. RELATED WORKS

Word stemming has long been recognized as a fundamental component of NLP. A wide range of techniques has been developed, including rule-based approaches, regular expression–based methods, and well-established algorithms such as the Snowball stemmer, which are primarily designed for English text processing [17]. In the context of Sundanese, several specialized stemming algorithms have been proposed, including those developed by Purwoko [18], UG18 Stemmer [19], Sutedi et al. [20], and the AMS Stemmer. These algorithms exploit distinct morphological and phonological characteristics of Sundanese and therefore exhibit varying levels of effectiveness. Among them, the AMS stemmer is currently regarded as state-of-the-art, as it employs a morphophonemic, rule-based framework that is well aligned with the linguistic structure of Sundanese.

In Transformer-based NLP architectures, stemming has been shown to improve downstream task performance when incorporated into the preprocessing pipeline. Previous studies have reported performance gains of up to 25% in classification and semantic tasks when effective stemming is applied [21]. However, the impact of stemming in information retrieval scenarios remains insufficiently explored, both for English and for other languages. Although prior work on information retrieval evaluation exists, stemming is rarely considered a major contributor to retrieval performance [22, 23].

Current research on information retrieval using Transformer-based models predominantly focuses on high-resource languages such as English. In contrast, existing work on Sundanese information retrieval remains limited mainly to lexical approaches [24], vector space models [25-28], and Bayesian methods. The application of state-of-the-art Transformer-based models to Sundanese information retrieval has received little attention. This gap highlights the need for more advanced neural retrieval frameworks specifically adapted to the linguistic characteristics of Sundanese.

Recent advances in neural information retrieval have demonstrated that Transformer architectures achieve state-of-the-art performance in high-resource language environments. However, adapting these models to low-resource languages poses substantial technical challenges, including severe data sparsity, high morphological complexity, and tokenization mismatches arising from suboptimal subword segmentation. Low-resource languages often exhibit rich affixation systems and morphophonemic alternations that are not adequately captured by tokenizers trained on high-resource language corpora.

To address these limitations, prior studies have explored multilingual pretraining, cross-lingual transfer learning, and domain-adaptive fine-tuning. Although these approaches have shown promise, they typically depend on lexical similarity or large-scale parallel corpora, which are rarely available for truly low-resource languages. These constraints underscore the need for linguistically informed normalization techniques to reduce morphological variation before semantic encoding.

In contrast to previous work, this study integrates a morphophonemic-based stemming mechanism with a fine-tuned Transformer model to form a hybrid retrieval architecture designed explicitly for low-resource Sundanese language processing. This approach combines linguistic normalization with deep semantic representation, enabling more accurate and contextually meaningful retrieval. By jointly addressing linguistic and architectural challenges, the proposed framework provides a novel contribution to low-resource neural information retrieval and establishes a clear methodological distinction from prior studies.

3. METHODS

This section describes the study methodology and the waterfall-style software development lifecycle adopted in this work [29]. Figure 1 presents the main stages of the Waterfall Method. The primary requirement of the proposed web application was the ability to perform Sundanese text stemming and semantic information retrieval through a dedicated search engine. To meet these requirements, the system was designed with two core components: a stemming module based on the AMS stemming algorithm and a Transformer-based search engine module. The proposed web application was evaluated from three perspectives: system performance, response latency, and user experience (UX).

3.1 System design

Web technologies enable large-scale access across multiple platforms, but reliable system performance requires adequate computing and storage resources. In addition, the database must support high-dimensional embedding vectors, which are not natively handled by most relational database management systems. The proposed system architecture follows industry best practices, as illustrated in Figure 2.

The overall architecture consists of three main components: (1) a database, (2) a web server, and (3) a reverse proxy. PostgreSQL 17 with the pgvector extension was used to store embedding vectors. The pgvector extension implements a Hierarchical Navigable Small World (HNSW) index to support efficient similarity search. The web server was implemented in Python 3.11 using Flask, with Gunicorn and

Nginx used to manage concurrent requests and worker processes. This software stack enabled the deployment of a scalable and robust semantic search system.

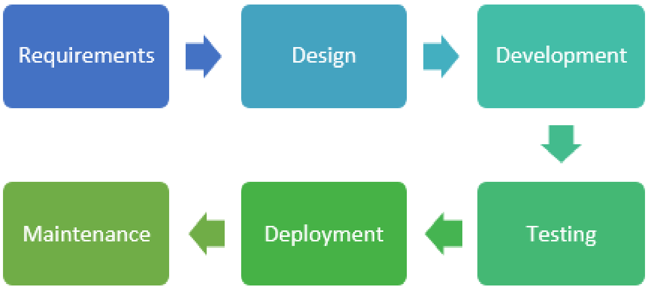


Figure 1. Research methodology

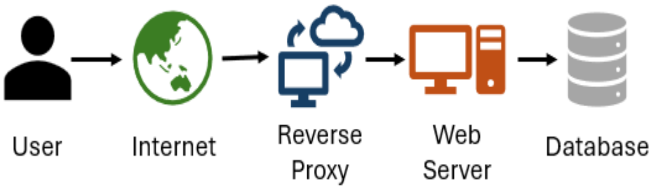


Figure 2. Proposed web app topology

3.2 Sundanese stemming module

Stemming is a fundamental NLP task that supports downstream applications such as information retrieval. An accurate and efficient stemming algorithm is therefore a critical component of the proposed web application. In this study, AMS stemming was adopted as it represents the current state of the art for the Sundanese language. The AMS algorithm is based on morphophonemic rules and achieved an F1 score of 87.52% in previous evaluations [30].

The Sundanese stemming module provided two main functions: (1) word-level stemming with word suggestion and (2) document-level stemming with accuracy analysis. Word suggestions were generated using the Levenstein distance [31] to recommend relevant dictionary entries when an input word was not found. This functionality improved usability and reduced user input errors. Document-level stemming applied the same algorithm to each unique word in a document and produced aggregate accuracy statistics, enabling efficient large-scale evaluation of stemming quality.

3.3 Sundanese search engine module

The search engine module consisted of four components: (1) AMS stemming, (2) a Transformer-based embedding model, (3) a vector store, and (4) cosine-based ranking, as shown in Figure 3. During indexing, source documents were processed using AMS stemming, converted into embeddings, and stored in the vector database. The same preprocessing pipeline was applied during querying to ensure consistency.

For retrieval, the query embedding was compared with document embeddings using cosine distance [32]. Given two n-dimensional vectors A and B, the cosine distance was computed as defined in Eq. (1). Documents were ranked by their cosine distances, with smaller distances indicating greater semantic similarity.

$$\mathrm{dist}(A,B) = 1 - \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

The Transformer model was implemented using a Siamese-BERT architecture [33]. Three pre-trained models were evaluated: MPNet [34], MelayuBERT [35], and IndoBERT [36], selected because their training data include Indonesian or Malay, which are linguistically related to Sundanese. Retrieval performance was evaluated using Precision@k, Recall@k, and mAP@k, as defined in Eqs. (2)–(4):

$$P_K = \frac{\#Relevant\ items\ on\ the\ ranking \leq K}{K} \tag{2}$$

$$AP_k = \frac{1}{N_d}\sum_{k=1}^{K} P_k rel_k \tag{3}$$

$$mAP = \frac{1}{N_q}\sum_{k=1}^{N} AP_k \tag{4}$$

where, N_d is the total number of relevant items, P_k is the precision at k, rel_i is the document relevance (1 if the item at top k is relevant, otherwise 0), N_q is the total number of queries, AP_k is the average precision at k, and mAP is the mean average precision. We chose to evaluate the retrieval performance with $k = 5$.

Mean Average Precision (MAP) is a widely used evaluation metric in information retrieval, including search engines [37, 38]. Precision and recall are fundamental measures for assessing retrieval effectiveness [39-41]. Precision represents the proportion of relevant items among all retrieved results, indicating how many of the returned documents satisfy the defined relevance criteria.

We published the dataset for fine-tuning and evaluation of the retrieval system in an open source repository with DOI: https://doi.org/10.5281/zenodo.15494944 [42]. We also compared search engine retrieval performance with that of public search engines and with baseline retrieval models. We prepared a unique 100-word list of common topics, calculated their respective evaluation scores, and performed an analysis of variance (ANOVA) to assess differences in mean retrieval performance across the search engine groups [43].

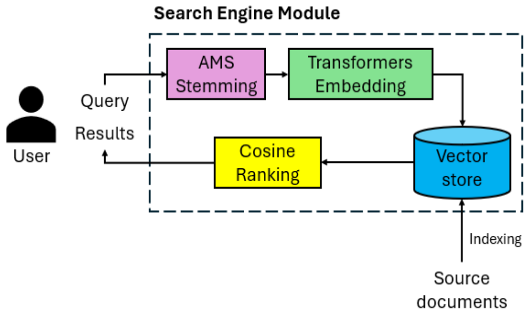


Figure 3. Search engine retrieval workflow diagram

3.4 Performance testing

System responsiveness was evaluated using the k6 load testing framework [44, 45]. Five levels of virtual users, ranging from 10 to 50 in increments of 10, were simulated over a 1-minute test duration. The total number of successful requests and the 95th percentile latency were recorded. All tests were conducted from a remote client over a 100 Mbps network connection. Lower latency and higher throughput were considered indicators of better system performance.

3.5 User experience (UX) testing

User experience is a critical factor in the adoption of information systems [46, 47]. A UX evaluation was conducted to assess the usability and effectiveness of the proposed web application. A total of 100 participants from diverse backgrounds took part in the review. The evaluation employed a structured questionnaire comprising 15 items grouped into five dimensions: clarity, relevance, completeness, delivery, and future use. Responses were recorded using a five-point Likert scale ranging from Strongly Disagree (1) to Agree (5) Strongly. The questionnaire items are provided in Appendix A to ensure transparency and reproducibility. For each dimension, the score was calculated as the mean of its three associated items. The overall UX score was obtained by averaging the five dimension scores. This aggregation method ensured internal consistency and facilitated quantitative comparison across evaluation dimensions.

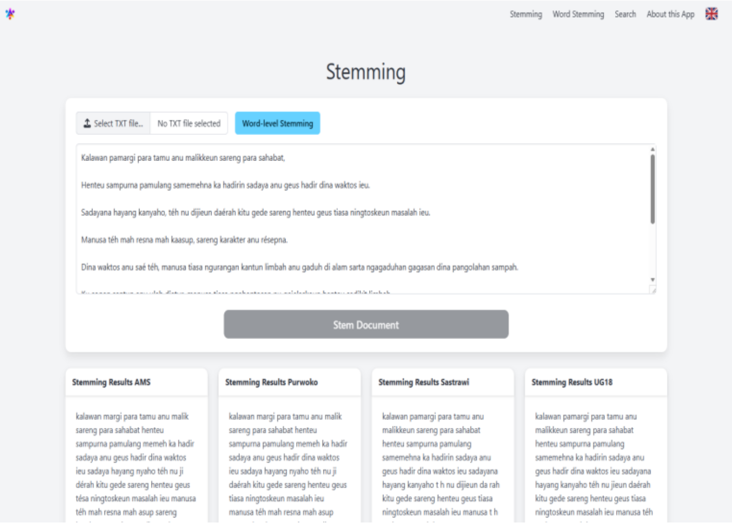
4. RESULT AND DISCUSSION

This section presents and analyzes the performance of the proposed system, including retrieval accuracy, system throughput, and user experience. The system was deployed on a virtual machine equipped with an Intel Xeon Gold 6248R CPU running Ubuntu Linux 22.04, with two virtual CPUs and 4 GB of memory, and without GPU acceleration. This configuration was selected to evaluate system behavior under cost-efficient, CPU-only conditions. The objective was to assess whether Transformer-based retrieval could be deployed effectively under limited computational resources.

The web application comprised three primary functional components, as illustrated in Figure 4. The first module supported document-level and word-level stemming for text preprocessing (Figure 4 (a)). The second module provided word-level analysis and dictionary-based normalization (Figure 4 (b)). The third module implemented the semantic search interface, which returned ranked documents along with cosine similarity scores and query response times (Figure 4 (c)). Together, these components enabled interactive, real-time processing and retrieval of Sundanese text.

4.1 Retrieval performance evaluation

The fine-tuning results of the Transformer-based models are presented in Table 1. The IndoBERT-based model achieved the highest performance, with an mAP@5 of 0.2883, a Precision@5 of 0.0780, and a Recall@5 of 0.3899. This result is consistent with the fact that IndoBERT was pre-trained on a large Indonesian corpus, which is linguistically closer to Sundanese than the data used for MPNet or MelayuBERT. MPNet, which was primarily pre-trained on English data, achieved lower but comparable scores, whereas MelayuBERT exhibited the weakest performance. These differences indicate that pre-training language similarity plays a substantial role in the effectiveness of downstream fine-tuning for low-resource languages.



Kalawan pamargi para tamu anu malikkeun sareng para sahabat,

Henteu sampurna pamulang samemehna ka hadirin sadaya anu geus hadir dina waktos ieu.

Sadayana hayang kanyaho, téh nu dijieun daérah kitu gede sareng henteu geus tiasa ningtoskeun masalah ieu.

Manusa téh mah resna mah kaasup, sareng karakter anu résepna.

Dina waktos anu saé téh, manusa tiasa ngurangan kantun limbah anu gaduh di alam sarta ngagaduhan gagasan dina pangolahan sampah.

Stemming Results AMS

kalawan margi para tamu anu malik sareng para sahabat henteu sampurna pamulang memeh ka hadir sadaya anu geus hadir dina waktos ieu sadaya hayang nyaho téh nu ji dérah kitu gede sareng henteu geus tésa ningtoskeun masalah ieu manusa téh mah resna mah asup sareng

Stemming Results Purwoko

kalawan margi para tamu anu malik sareng para sahabat henteu sampurna pamulang memeh ka hadir sadaya anu geus hadir dina waktos ieu sadaya hayang nyaho téh nu ji daérah kitu gede sareng henteu geus tiasa ningtoskeun masalah ieu manusa téh mah resna mah asup

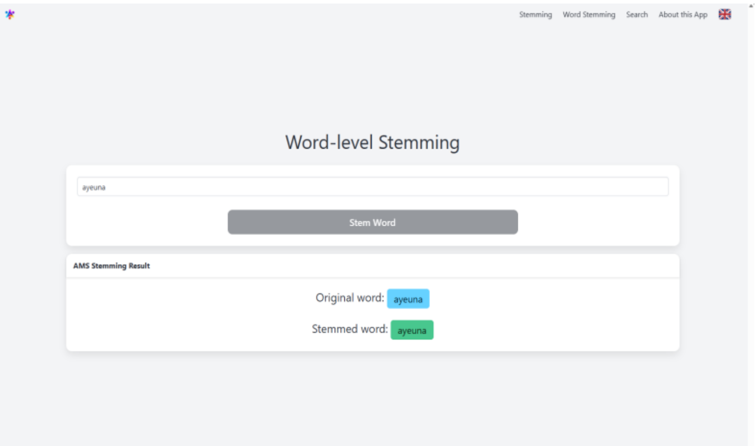
Stemming Results Sastrawi

kalawan pamargi para tamu anu malikkeun sareng para sahabat henteu sampurna pamulang samemehna ka hadirin sadaya anu geus hadir dina waktos ieu sadayana hayang kanyaho t h nu dijieun da rah kitu gede sareng henteu geus tiasa ningtoskeun masalah ieu manusa t h

Stemming Results UG18

kalawan pamargi para tamu anu malikkeun sareng para sahabat henteu sampurna pamulang samemehna ka hadirin sadaya anu geus hadir dina waktos ieu sadayana hayang kanyaho téh nu jieun daérah kitu gede sareng henteu geus tiasa ningtoskeun masalah ieu manusa téh

(a) Document-level stemming



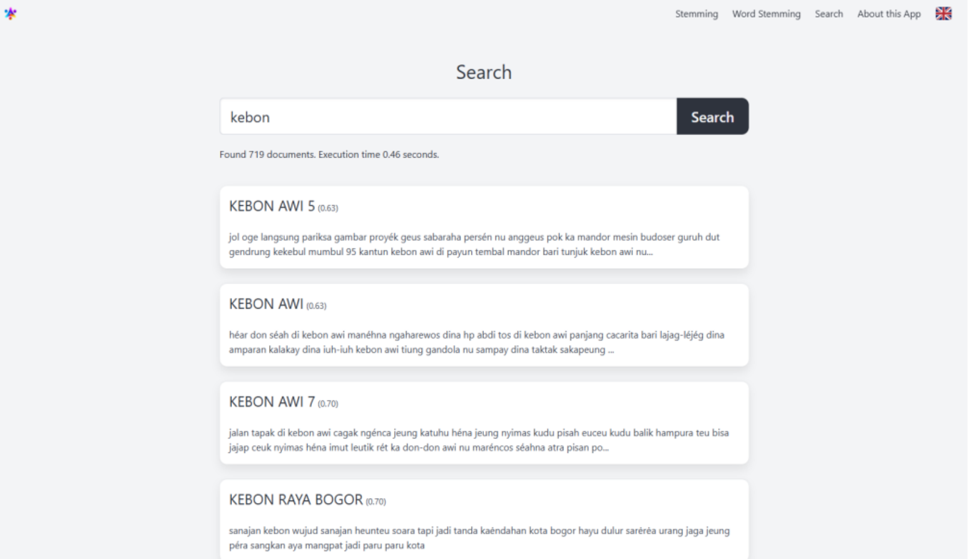
ayeuna

AMS Stemming Result

Original word: ayeuna

Stemmed word: ayeuna

(b) Word-level stemming



Found 719 documents. Execution time 0.46 seconds.

KEBON AWI 5 (0.63)

jol oge langsung pariksa gambar proyék geus sabaraha persén nu anggeus pok ka mandor mesin budoser guruh dut gendrung kekebul mumbul 95 kantun kebon awi di payun tembal mandor bari tunjuk kebon awi nu...

KEBON AWI (0.63)

héar don séah di kebon awi manéhna ngaharewos dina hp abdi tos di kebon awi panjang cacarita bari lajag-léjég dina amparan kalakay dina iuh-iuh kebon awi tiung gandola nu sampay dina taktak sakapeung ...

KEBON AWI 7 (0.70)

jalan tapak di kebon awi cagak ngénca jeung katuhu héna jeung nyimas kudu pisah euceu kudu balik hampura teu bisa jajap ceuk nyimas héna imut leutik rét ka don-don awi nu maréncos séahna atra pisan po...

KEBON RAYA BOGOR (0.70)

sanajan kebon wujud sanajan heunteu soara tapi jadi tanda kaéndahan kota bogor hayu dulur saréréa urang jaga jeung péra sangkan aya mangpat jadi paru paru kota

(c) Semantic search engine

Figure 4. Intelligent web app screenshots

Table 1. Transformers model retrieval performance

Model	mAP@5	Prec.@5	Recall@5
MPNet	0.2550	0.0707	0.3536
MelayuBERT	0.1765	0.0513	0.2565
IndoBERT	**0.2883**	**0.0780**	**0.3899**

The performance difference can be attributed to the fine-tuning method used for a Transformer-based model. When fine-tuning a Transformer model, the tokenizer is usually not updated with new corpus data; only the model weights are updated [48]. This is both an advantage and a primary limitation of Transformer-based models. When fine-tuning is performed using the same language as the pre-trained model, the model can generalize and produce better, more specific models. However, when the same model is fine-tuned using a significantly different language and alphabet, it may not learn the correct semantic meaning from the corpus [49, 50].

Due to the limited number of pre-trained Indonesian-language models, this severely restricts the potential for fine-tuning to create new, task-specific Transformer models. While we can train from scratch, the Transformer model requires a large corpus to produce an effective model. However, data are scarce for low-resource languages such as Sundanese. Nevertheless, we consider these results satisfactory for integration into the web app.

Table 2. Comparison with baseline retrieval models

Model	mAP@5	Prec.@5	Recall@5
TF–IDF (VSM)	0.1421	0.0394	0.3536
IndoBERT (non-fine-tuned)	0.2015	0.0582	0.2971
Proposed AMS + IndoBERT (Fine-tuned)	**0.2883**	**0.0780**	**0.3899**

To ensure a fair and rigorous scientific evaluation, additional baseline retrieval models were incorporated into the experimental framework. An unfine-tuned IndoBERT model was employed as a zero-shot Transformer baseline, providing a neural retrieval baseline without task-specific adaptation. In contrast, a traditional TF–IDF vector space model was used to capture classical lexical retrieval behavior. The inclusion of these baselines enables a more controlled and methodologically sound comparison with the proposed fine-tuned Transformer-based retrieval model in a low-resource language setting. As demonstrated in Table 2, the comparative results reveal a clear and consistent performance improvement of the proposed model over both classical and neural baseline approaches.

To provide a rigorous evaluation, two baseline retrieval models were included: a TF–IDF vector space model and a zero-shot IndoBERT model without fine-tuning. As shown in Table 2, the TF–IDF model achieved the lowest retrieval effectiveness (mAP@5 = 0.1421), indicating that lexical matching alone was insufficient for capturing semantic relationships in Sundanese text. The zero-shot IndoBERT model showed moderate improvement (mAP@5 = 0.2015), confirming the benefit of contextual embeddings even without task-specific adaptation.

The proposed AMS-integrated, fine-tuned IndoBERT model achieved the highest performance across all metrics (mAP@5 = 0.2883, Precision@5 = 0.0780, Recall@5 = 0.3899). This represents an improvement of approximately 43.9% over TF–IDF and 43.1% over the zero-shot IndoBERT baseline in terms of mean Average Precision. These consistent gains across multiple metrics indicate that integrating morphophonemic normalization with Transformer-based semantic encoding substantially improves retrieval quality in low-resource language settings.

4.2 Performance testing

Figures 5 and 6 summarize the performance evaluation of the stemming and search modules. The stemming module maintained stable performance as the number of virtual users increased from 10 to 50. The average latency remained below 300 ms, and throughput scaled approximately linearly with the number of users. At peak load, the system processed approximately 2,740 requests, with a 95th-percentile latency of 320.34 ms, indicating high responsiveness and scalability.

In contrast, the search module exhibited a throughput limit of approximately 800 requests across all user levels. Latency increased with higher concurrency, reflecting the computational cost of Transformer-based embedding inference on CPU-only hardware. Nevertheless, the system remained responsive with up to 50 concurrent users, demonstrating that the architecture could support multi-user operation in real-time conditions.

Runtime profiling indicated that Transformer inference accounted for approximately 68–75% of total query latency, while vector similarity search using the HNSW index contributed an additional 18–22%. The remaining overhead was associated with web server routing and network communication. These results provide a reproducible baseline for future system optimization.

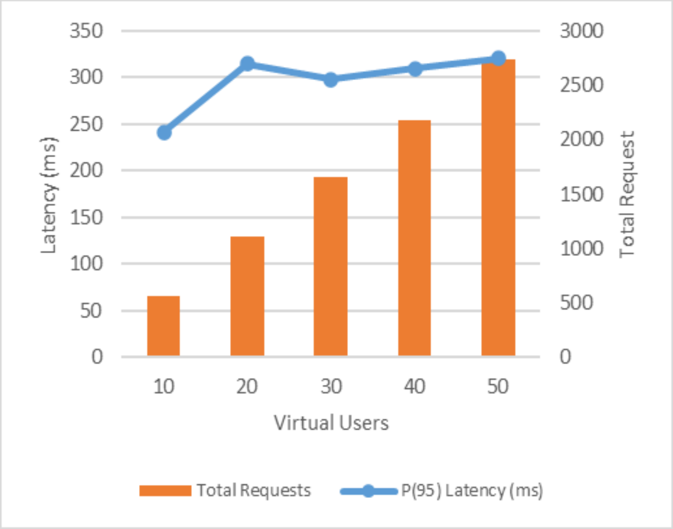


Figure 5. Performance test results for stemming capability

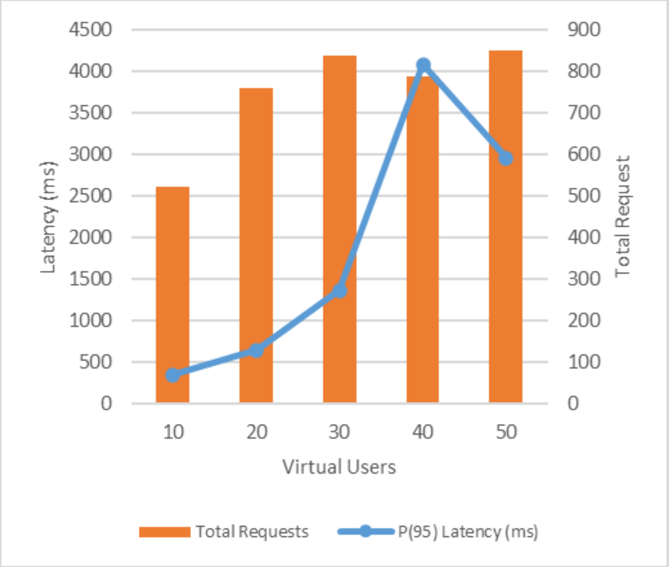


Figure 6. Performance test results for the searching capability

4.3 Error analysis and performance limitations

Although the proposed system outperformed baseline and reference models, the absolute values of Precision@5 and mAP@5 remained moderate. This outcome reflects the inherent challenges of low-resource language retrieval. Error analysis identified three primary sources of performance degradation: limited vocabulary coverage, suboptimal tokenization inherited from IndoBERT, and semantic ambiguity arising from dialectal variation in Sundanese.

In addition, the size of the available corpus constrained model generalization. Unlike high-resource languages, Sundanese lacks large-scale annotated datasets and pre-trained Transformer models, which limits the achievable retrieval accuracy. Therefore, the reported results should be interpreted as a realistic baseline under low-resource conditions rather than as an upper bound on achievable performance.

4.4 User experience (UX) testing

Figure 7 presents the results of the UX evaluation. Participants reported positive perceptions across all five dimensions. Clarity achieved the highest score (88.83%), followed by relevance (88.71%), future use (87.83%), delivery (87.67%), and completeness (86.50%). These results indicate that users found the interface intuitive, the retrieval results meaningful, and the system reliable.

The consistently high scores across dimensions suggest that the proposed system offered a balance between usability and retrieval effectiveness. Minor differences among dimensions indicate opportunities for further refinement, particularly in expanding the coverage of results and handling edge-case queries.

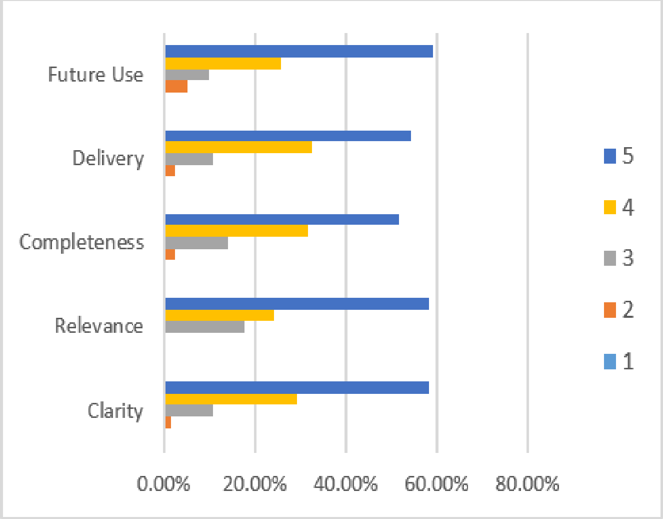


Figure 7. User experience testing questionnaire results

5. CONCLUSIONS

This study developed an integrated web-based Sundanese Natural Language Processing system that combines morphophonemic stemming and Transformer-based semantic search. The AMS stemming algorithm demonstrated strong computational efficiency, achieving a latency below 400 ms with linear scalability as the number of concurrent users increased. These results indicate that the stemming module performs reliably and efficiently, making it suitable for large-scale linguistic processing in low-resource language environments such as Sundanese. The semantic search engine, implemented using a fine-tuned IndoBERT Transformer model, achieved an mAP@5 of 0.2883, a Precision@5 of 0.0780, and a Recall@5 of 0.3899. These results confirm that the system can capture user intent and retrieve semantically relevant documents despite operating under CPU-only and low-resource data constraints. The user experience evaluation further supported the system's effectiveness, with high ratings in clarity and relevance, indicating that the interface and retrieval outputs were both intuitive and functional.

Future work may focus on improving scalability and retrieval accuracy through GPU-based inference, distributed deployment, and expansion of the Sundanese text corpus. In addition, extending the system to support cross-lingual retrieval and Retrieval-Augmented Generation (RAG) would enable more advanced applications, such as contextual question answering and semantic summarization. Overall, this study provides a robust and reproducible foundation for advancing Transformer-based information retrieval in low-resource languages.

AUTHOR'S CONTRIBUTIONS

Aries Maesya contributed to the conceptualization, methodology, preparation of the original draft, and visualization of the study. **Yulyani Arifin** contributed to supervision, validation, and investigation. **Amalia Zahra** contributed to the supervision, validation, review, and editing of the manuscript. **Widodo Budiharto** contributed to supervision, verification, and investigation. All authors read and approved the manuscript.

REFERENCES

[1] Al Naqbi, H., Bahroun, Z., Ahmed, V. (2024). Enhancing work productivity through generative artificial intelligence: A comprehensive literature review. Sustainability, 16(3): 1166. https://doi.org/10.3390/su16031166

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al. (2017). Attention is all you need. arXiv preprint arXiv: 1706.03762. https://doi.org/10.48550/arXiv.1706.03762

[3] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., et al. (2023). GPT-4 technical report. arXiv preprint arXiv: 2303.08774. https://doi.org/10.48550/arXiv.2303.08774

[4] Lewis, P., Perez, E., Piktus, A., Petroni, F., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33: 9459-9474.

[5] Manning, C.D., Raghavan, P., Schütze, H. (2008). Introduction to Information Retrieval. Cambridge: Cambridge University Press. http://informationretrieval.org.

[6] Asqolani, I.A., Setiawan, E.B. (2023). Hybrid deep learning approach and Word2Vec feature expansion for cyberbullying detection on Indonesian twitter. Ingénierie des Systèmes d'Information, 28(4): 887-895. https://doi.org/10.18280/isi.280410

[7] Imaduddin, H., Kusumaningtias, L.A., A'la, F.Y. (2023). Application of LSTM and GloVe word embedding for hate speech detection in Indonesian twitter data. Ingénierie des Systèmes d'Information, 28(4): 1107-1112. https://doi.org/10.18280/isi.280430

[8] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5: 135-146. https://doi.org/10.1162/tacl_a_00051

[9] Yusrandi, Muladi, Rosyid, H.A., Mahamad, A.K. (2021). Document search in information retrieval System using vector space model. In 2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Malang, Indonesia, pp. 604-608. https://doi.org/10.1109/ICEEIE52663.2021.9616735

[10] Bird, S., Klein, E., Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc.

[11] Yilmaz, Z.A., Wang, S., Yang, W., Zhang, H., Lin, J. (2019). Applying BERT to document retrieval with birch. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, Hong Kong, China, pp. 19-24. https://doi.org/10.18653/v1/D19-3004

[12] Qiao, Y., Xiong, C., Liu, Z., Liu, Z. (2019). Understanding the behaviors of BERT in ranking. arXiv preprint arXiv: 1904.07531. https://doi.org/10.48550/arXiv.1904.07531

[13] Yang, Y., Qiao, Y., Shao, J., Yan, X., Yang, T. (2022). Lightweight composite re-ranking for efficient keyword search with BERT. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event, AZ, USA, pp. 1234-1244. https://doi.org/10.1145/3488560.3498495

[14] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805. https://doi.org/10.48550/arXiv.1810.04805

[15] Alhammad, R. (2023). The phonology morphology and syntax of Sundanese. In Forum for Linguistic Studies (Transferred), pp. 1945-1945. https://doi.org/10.59400/fls.v5i3.1945

[16] Maesya, A., Arifin, Y., Zahra, A., Budiharto, W. (2023). Development of Sundanese Stemmer Based on Morphophonemics. In 2023 10th International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, pp. 1-6. https://doi.org/10.1109/ICISS59129.2023.10291840

[17] Porter, M.F. (2001). Snowball: A language for stemming algorithms. http://snowball.tartarus.org/texts/introduction.

[18] Purwoko, A. (2011). Dictionary-based stemming model for documents in Sundanese language. http://repository.ipb.ac.id/handle/123456789/56568.

[19] Mulyana, I., Suhendra, A., Ernastuti, W, B.A. (2019). Development of Indonesian stemming algorithms through modification of grouping, sequencing, and removing of affixes based on morphophonemic. International Journal of Recent Technology and Engineering (IJRTE), 8(2S7): 179-184. https://doi.org/10.35940/ijrte.B1044.0782S719

[20] Sutedi, A., Elsen, R., Nasrulloh, M.R. (2021). Sundanese stemming using syllable pattern. Jurnal Online Informatika, 6(2): 218-224. https://doi.org/10.15575/join.v6i2.812

[21] Siino, M., Tinnirello, I., La Cascia, M. (2024). Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers. Information Systems, 121: 102342. https://doi.org/10.1016/j.is.2023.102342

[22] Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I. (2021). Beir: A heterogenous benchmark

for zero-shot evaluation of information retrieval models. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract-round2.html.

[23] Muennighoff, N., Tazi, N., Magne, L., Reimers, N. (2022). MTEB: Massive text embedding benchmark. arXiv e-prints arXiv: 2210.07316. https://doi.org/10.48550/arXiv.2210.07316

[24] Sugiri, D., Hikmayanti, H., Suharso, A. (2019). Rancang bangun aplikasi kamus sunda-indonesia dengan metode binary search berbasis android. Techno Xplore: Jurnal Ilmu Komputer dan Teknologi Informasi, 4(1): 1-14. https://doi.org/10.36805/technoxplore.v4i1.537

[25] Heristian, S., Al Kautsar, H.A., Sayfulloh, A. (2019). Rancang bangun information retrieval system (IRS) kamus bahasa-sunda. com dengan metode vector space model (VSM). JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer), 5(1): 65-72. https://doi.org/10.33480/jitk.v5i1.677

[26] Suryani, M., Hadi, S., Paulus, E., Yulita, I.N., Supriatna, A.K. (2017). Sundanese ancient manuscripts search engine using probability approach. Journal of Physics: Conference Series, 893(1): 012064. https://doi.org/10.1088/1742-6596/893/1/012052

[27] Suryani, M., Hadi, S., Nasuha, A.M.I. (2016). Sundanese ancient manuscript retrieval system comparison of two probability approaches. In 2016 International Conference on Informatics and Computing (ICIC), Mataram, Indonesia, pp. 105-110. https://doi.org/10.1109/IAC.2016.7905698

[28] Lestari, D.P., Furui, S. (2010). Adaptation to pronunciation variations in Indonesian spoken query-based information retrieval. IEICE Transactions on Information and Systems, 93(9): 2388-2396. https://doi.org/10.1587/transinf.E93.D.2388

[29] Petersen, K., Wohlin, C., Baca, D. (2009). The waterfall model in large-scale development. In International Conference on Product-Focused Software Process Improvement, pp. 386-400. https://doi.org/10.1007/978-3-642-02152-7_29

[30] Maesya, A., Ramadhan, A., Abdurachman, E., Trisetyarso, A., Zarlis, M. (2022). Stemming algorithm for the Indonesian language: A scientometric view. In 2022 IEEE Creative Communication and Innovative Technology (ICCIT), Tangerang, Indonesia, pp. 1-6. https://doi.org/10.1109/ICCIT55355.2022.10119050

[31] Navarro, G. (2001). A guided tour to approximate string matching. ACM Computing Surveys (CSUR), 33(1): 31-88. https://doi.org/10.1145/375360.375365

[32] Singhal, A. (2001). Modern information retrieval: A brief overview. IEEE Data Eng. Bull., 24(4): 35-43. https://d1wqtxts1xzle7.cloudfront.net/42414681/A01DEC-CD-libre.pdf.

[33] Reimers, N., Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv: 1908.10084. https://doi.org/10.48550/arXiv.1908.10084

[34] Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y. (2020). MPNet: Masked and permuted pre-training for language understanding. arXiv preprint arXiv:2004.09297. https://doi.org/10.48550/arXiv.2004.09297

[35] Limcorn, S., Wongso, W. (2025). StevenLimcorn/MelayuBERT. https://huggingface.co/StevenLimcorn/MelayuBERT, accessed on July 21, 2025.

[36] Koto, F., Rahimi, A., Lau, J.H., Baldwin, T. (2020). IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP. In COLING 2020-28th International Conference on Computational Linguistics, Proceedings of the Conference, pp. 757-770. https://doi.org/10.18653/v1/2020.coling-main.66

[37] Cormack, G.V., Lynam, T.R. (2006). Statistical precision of information retrieval evaluation. In Proceedings of the 29th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, pp. 533-540. https://doi.org/10.1145/1148170.11482

[38] Shah, D. (2022). Mean average precision (mAP) explained: Everything you need to know. V7 Labs. https://www.v7labs.com/blog/mean-average-precision.

[39] Brucato, M., Montesi, D. (2014). Metric spaces for temporal information retrieval. European Conference on Information Retrieval, pp. 385-397. https://doi.org/10.1007/978-3-319-06028-6_32

[40] Carnevali, L. (2023). Evaluation measures in information retrieval. Pinecone. https://www.pinecone.io/learn/offline-evaluation/.

[41] Tan, R.J. (2024). Mean average precision (mAP) explained. Built In. https://builtin.com/articles/mean-average-precision.

[42] Maesya, A., Arifin, Y., Zahra, A., Budiharto, W. (2025). AMSunda: A novel dataset for Sundanese information retrieval. Data in Brief, 61: 111796. https://doi.org/10.1016/j.dib.2025.111796

[43] Mann, P.S. (2010). Introductory Statistics. John Wiley & Sons, New Jersey, USA.

[44] Kao, C.H., Lin, C.C., Chen, J.N. (2013). Performance testing framework for rest-based web applications. In 2013 13th International Conference on Quality Software, Najing, China, pp. 349-354. https://doi.org/10.1109/QSIC.2013.32

[45] Khan, R., Amjad, M. (2016). Performance testing (load) of web applications based on test case management. Perspectives in Science, 8: 355-357. https://doi.org/10.1016/j.pisc.2016.04.073

[46] Yin, R., Zhang, B., Kang, M., Li, T., Chen, K., Kang, Y. (2015). Basic principles of information system UI design. In International Conference on Man-Machine-Environment System Engineering, pp. 419-423. https://doi.org/10.1007/978-3-662-48224-7_50

[47] Hu, P.J.H., Ma, P.C., Chau, P.Y. (1999). Evaluation of user interface designs for information retrieval systems: a computer-based experiment. Decision Support Systems, 27(1-2): 125-143. https://doi.org/10.1016/S0167-9236(99)00040-8

[48] Dagan, G., Synnaeve, G., Roziere, B. (2024). Getting the most out of your tokenizer for pre-training and domain adaptation. arXiv preprint arXiv: 2402.01035. https://doi.org/10.48550/arXiv.2402.01035

[49] Pires, T., Schlinger, E., Garrette, D. (2019). How multilingual is multilingual BERT?. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4996-5001. https://doi.org/10.18653/v1/P19-1493

[50] Shuvo, M.M.H., Islam, S.K., Cheng, J., Morshed, B.I. (2022). Efficient acceleration of deep learning inference

on resource-constrained edge devices: A review. Proceedings of the IEEE, 111(1): 42-91. https://doi.org/10.1109/JPROC.2022.3226481