Journal of Informatics and Vocational Education (JOIVE)
Vol.9, No.1, January 2026, pp. 58~66
ISSN: 2746-7813

AI Based Digital Book Indexing System Using YAKE and WORD2VEC Methods

Mohammad Alfarizi Abdullah[1], Ulla Delfana Rosiana[2], Vit Zuraida[3], Arhan Windu Rizki Putra Budianto [4], Rizki Putri Ramadhani [5]
[1,2,3,4] Information Technology Department, State Polytechnic Of Malang
[5] Civil Engineering Department, State Polytechnic Of Malang

Article Info

Article history:

Received Dec 16, 2025
Revised Jan 6, 2026
Accepted Feb 1, 2026

Corresponding Author:

Ulla Delfana Rosiana, Information Technology Department, State Polytechnic Of Malang. Affiliation address, Street name, Street number, City, State, Country, Postcode
Email: rosiani@polinema.ac.id

ABSTRACT

Polinema Press, the publishing unit of the State Polytechnic of Malang (Polinema),requires an efficient solution for automatically generating book indexes. The current manual indexing process is time-consuming and inefficient. This research aims to develop an AI-based automatic indexing system utilizing the YAKE (Yet Another Keyword Extractor) and Word2Vec methods to improve the accuracy and efficiency of index generation. The system is designed to process digital books in PDF format through several stages: (1) text preprocessing (text extraction, stopword removal, tokenization), (2) keyword extraction using YAKE based on statistical features such as word frequency and position, (3) final keyword selection by measuring semantic similarity using Word2Vec, and (4) alphabetical index compilation along with page numbers where keywords appear. The indexing results are evaluated by comparing them with manual indexes using cosine similarity to measure the degree of similarity. Experimental results on 37 digital books show that the proposed system achieves an average cosine similarity of 0.33–0.37, with precision values ranging from 0.06 to 0.11 depending on phrase length configuration. While semantic similarity scores indicate strong contextual relevance, precision and recall remain relatively low, highlighting the challenges of aligning automatic indexes with manually curated ones. These results show that the system is able to produce relevant, fast, and contextual indexes when compared to manual indexes, and is expected to reduce the manual workload at Polinema Press and become a reference for the application of natural language processing (NLP) technology for Indonesian-language documents.

Keywords: Automatic Indexing, YAKE, Word2Vec, Artificial Intelligence, Natural Language Processing.

1. INTRODUCTION

Book indexing is the arrangement of important words or terms contained in a book in alphabetical order, accompanied by the page number of the important words or terms. Indexing functions to make it easier for readers to find specific information quickly (Ganapathy & Fadziso, 2020). Along with technological developments, automatic indexing has emerged as a solution to increase efficiency and accuracy in the indexing process. This process uses software and natural language processing algorithms to automatically identify and organize important terms in the text. Research by Dimasputra Bagawan (2023) shows how a web application system can be used for automatic book indexing by utilizing natural language processing techniques. Thus, automatic indexing not only saves time but also ensures consistency and ease of access to information for readers (Fauzi et al., 2017).

Indexing faces several challenges, one of which is the need to correct words one by one, which is time-consuming. The second is manual input, which is inefficient in terms of time. These three challenges require an automated book indexing system that is expected to perform optimally in terms of automatic word selection, accuracy of important words or terms, and ease of use for users. This system utilizes two current methods to ensure optimal indexing: the YAKE and Word2Vec methods. These two methods have different functions. The YAKE model (Gupta et al., 2024) is an automated method used to extract keywords designed to work efficiently on specific books or

documents without requiring training data or additional document collections. YAKE (Campos et al., 2020) can work effectively by filtering out keywords that do not contain punctuation and that do not begin or end with stop words.

Word2Vec is a word embedding technique used to group word vectors with similar meanings. This method can accurately estimate the similarity of meaning between words when trained using a sufficiently large dataset (Ariesta et al., 2023). This study proposes an AI-based book indexing system to support efficient and practical automatic index generation at Polinema Press. This system is expected to reduce author obstacles in indexing. By utilizing two methods, YAKE and Word2vec, this system is expected to provide convenience and provide high-quality index results.

Different from previous studies that focus primarily on keyword extraction accuracy, this research emphasizes the practical implementation of an AI-based indexing system for Indonesian-language digital books. The contribution of this study lies in (1) the integration of YAKE and FastText-based Word2Vec for book-level indexing, (2) extensive evaluation using multiple phrase-length configurations, (3) comparison with real manual indexes from a publishing institution, and (4) analysis of processing time for real-world deployment.

2. AI BASED BOOK INDEXING

2.1. BOOK INDEXING

Book Indexing is the process of compiling a list of terms or important entries from the contents of a book that are sorted alphabetically or systematically, accompanied by the page numbers where the terms appear (ONWUCHEKWA, 2024). The main purpose of an index is to make it easier for readers to find specific information in the text without having to read the entire book (Mai, 1999). According to Arifin, an index is a list of important words or terms found in a printed book (usually placed at the end of a book section), arranged alphabetically and providing information on the page where the words or terms are found in the book. The process of compiling an index includes selecting relevant terms, normalizing terms, and organizing them systematically to make information access efficient (Anderson & Pérez-Carballo, 2001).

2.2. APPLICATION OF AI IN BOOK INDEXING

Advances in artificial intelligence (AI) technology have made it possible to automate the book indexing process. AI can analyze large amounts of text, recognize important words or phrases, and determine their relevance to the context of the book's content. By utilizing Natural Language Processing (NLP), the system is able to understand the structure of natural language, thus mimicking the way humans recognize the main topic of a document. This approach offers significant advantages, such as a faster and more efficient process compared to manual methods, consistent results because the algorithm operates based on the same parameters, the ability to process large numbers of documents in a short time, and easy integration with digital systems, such as automatic search or glossary creation.

2.3. GENERAL STAGES IN THE AI-BASED BOOK INDEXING SYSTEM

1. Text Input and Extraction: The system receives document files in PDF, DOCX, or TXT format. It then performs a text extraction process to obtain pure content that can be analyzed by the machine.
2. Data Preprocessing: This stage includes cleaning the text from irrelevant elements such as punctuation, numbers, and stopwords, as well as word normalization (lowercasing, stemming, or lemmatization). The goal is to prepare the data for further analysis.
3. Keyword Extraction: This process aims to find the most representative words or phrases from the document's content. This study used the YAKE (Yet Another Keyword Extractor) method, which can extract keywords without requiring an external corpus or training data.
4. Semantic Analysis with Word Embedding: Once keywords are acquired, the system uses the Word2Vec model to calculate semantic similarity between words or between keywords and book titles. Word2Vec converts words into numeric vector representations based on the context in which they are used in the text.
5. Similarity Calculation: Using this vector representation, the similarity level (cosine similarity) between the keyword and the main topic of the book is calculated. The similarity value helps determine the keyword's relevance to the overall content.

6. *Compiling Indexing Results: Highly relevant keywords are displayed along with additional information such as relevance scores, page appearances, and surrounding sentence context. The final results can be saved in PDF or Excel format for easy documentation and searching.*

2.4. YAKE AND WORD2VEC

The selection of the YAKE (Yet Another Keyword Extractor) and Word2Vec methods in an AI-based book indexing system is based on considerations of their effectiveness, efficiency, and ability to extract and represent the semantic meaning of a text. The YAKE method was developed by Campos et al. (2020) as an unsupervised keyword extraction algorithm capable of extracting important words or phrases without requiring an external corpus or training data. YAKE works by analyzing intrinsic statistical features of the text, such as word frequency, position distribution, capitalization, and the context surrounding the words. This approach makes YAKE very efficient for use on long documents such as books, because the process does not rely on complex machine learning models. In addition, YAKE supports the extraction of multi-word keywords (e.g., two to three words), which provides results that are more representative of the concepts raised in the text.

Meanwhile, Word2Vec is used to strengthen the semantic analysis of keyword extraction results. Word2Vec is a word embedding model introduced by Mikolov et al. (2013), which maps words into a high-dimensional vector space based on their context of use in the text. Thus, words with similar meanings will be in adjacent vector positions. In the context of a book indexing system, Word2Vec is used to calculate the level of semantic similarity (cosine similarity) between extracted keywords and the title or main topic of the book. This approach ensures that the selected keywords are not only statistically relevant but also conceptually meaningful. The combination of these two methods results in a more accurate and contextual automatic indexing system. YAKE plays a role in selecting keyword candidates based on local text features, while Word2Vec assesses global semantic relevance through meaning vector analysis. Thus, the obtained index results not only represent the frequency of term occurrence but also reflect the semantic relationships between concepts in the document as a whole.

Recent studies have explored transformer-based models such as BERT for keyphrase extraction, which demonstrate superior contextual understanding. However, these models require substantial computational resources and large training data, making them less practical for institutional-scale book indexing. This study deliberately adopts Word2Vec with FastText embeddings to balance semantic representation and computational efficiency, especially for Indonesian-language documents.

2.5. FLOWCHART SYSTEM

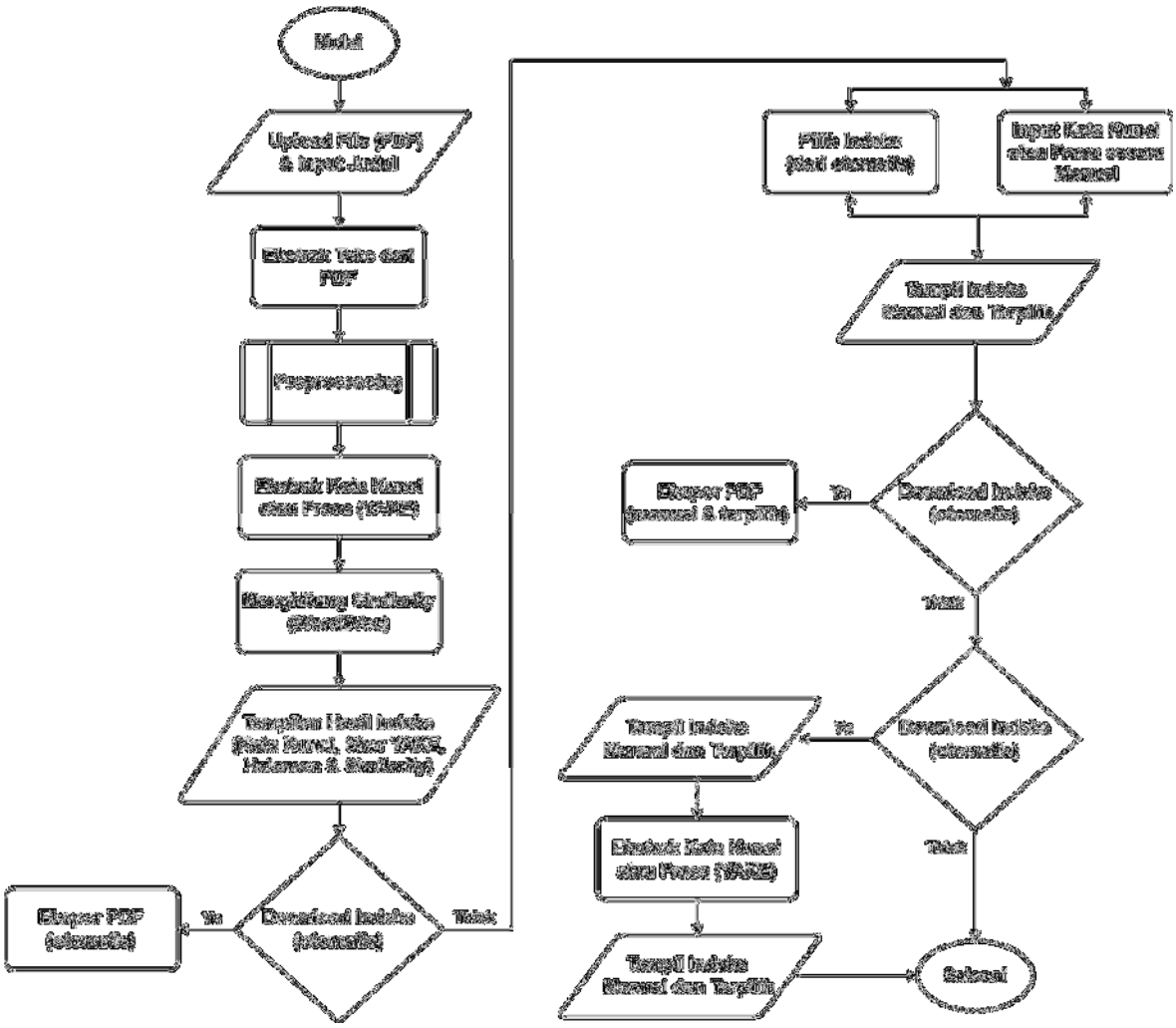

Figure 1. Flowchart Sistem

The system process can be seen in the image, starting with the user uploading the PDF file they wish to index. The system then extracts text from the uploaded PDF. After that, preprocessing is carried out, including text cleaning such as removing numbers, symbols, punctuation, and word normalization. Next, the system extracts potential keywords or phrases (n-grams) using the YAKE method. After that, the similarity calculation process between the phrase and the title is carried out using Word2Vec (FastText). If users want to see the results immediately, the system displays the automatic indexing results in the form of keywords, YAKE scores, page numbers, and similarity values. Users can download the results in PDF format. Alternatively, users can compare the index results with a manual index. In this case, users can select or manually input the index from directly typed results. Users can then upload the manual index data to the system, which then compares it with the automatic results using cosine similarity, precision, and recall.

The comparison results are then displayed, indicating the similarity level (in the form of cosine similarity, precision, and recall scores). As a final step, the system displays the final comparison results between the automatic and manual indexes, including cosine similarity, precision, and recall values.

2.6. FLOWCHART PREPROCESSING

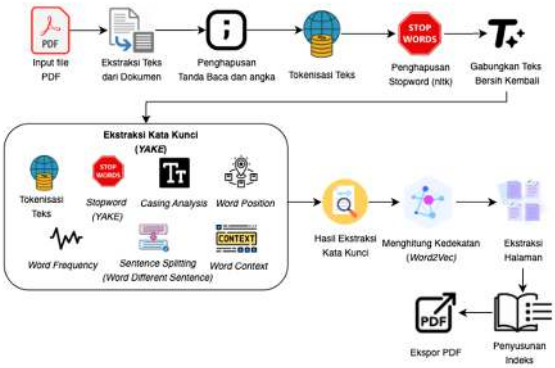

Figure 2. Flowchart Preprocessing

The data processing process in this AI-based indexing system begins with the author uploading a document that supports PDF format. The system will extract text from the document using the relevant library, PyMuPDF. After extraction, the next step is keyword extraction using the YAKE (Yet Another Keyword Extractor) algorithm. YAKE will generate a list of 1 to 3 keywords or phrases. The extraction results include the keywords or phrases found, their YAKE scores, and the pages where the keywords or phrases appear. To assess the relevance of keywords or phrases to the document context, semantic similarity calculations are performed between the keywords and the document title using FastText Bahasa Indonesia. Each keyword and title are converted into a vector representation by calculating the average embedding of the words. Then, the similarity between the keywords and the title is calculated using the Word2Vec method. The final result of this process is indexing data in the form of keywords, YAKE scores, pages where they appear, and similarity values with the title.

The data is stored in a structured manner and displayed in tabular form through the user interface. Additionally, users can download results in PDF format, manually add keywords, and use real-time search and filtering features. The system also offers a comparison function for automatic indexing results with a user-uploaded manual index list to measure the closeness of the AI results to the human-generated version.

In this study, text extraction from PDF documents was performed using the PyMuPDF library. Text preprocessing included case folding, removal of punctuation, numbers, and special characters, tokenization, and Indonesian stopword removal to reduce textual noise before keyword extraction. Keyword extraction was conducted using the YAKE algorithm with multiple phrase-length configurations, including unigram, bigram, trigram, 1–3 words, and 2–3 words, where each extracted keyword was associated with its YAKE score and page occurrence information. Semantic similarity evaluation was carried out using a pre-trained FastText-based Word2Vec model for the Indonesian language with a vector dimension of 300. For multi-word phrases, the vector representation was obtained by averaging the embeddings of individual words, and cosine similarity was calculated between each keyword phrase and the book title. Similarity threshold values of 0.3 and 0.5 were applied to filter extracted keywords and analyze the effect of semantic strictness on indexing performance. All experiments were implemented using Python and supported by YAKE, Gensim, and Scikit-learn libraries, enabling reproducibility without requiring high computational resources.

3. RESULT AND ANALYSIS

This study tested an AI-based automatic indexing system using the YAKE method and a combination of YAKE with context similarity measurement through cosine similarity using Word2Vec. Testing was conducted on 37 digital books, then selected into 30 best books as the main dataset. The purpose of this test was to see the effectiveness of the method in generating relevant keyword indexes, reviewed by precision, recall, and cosine similarity metrics. In addition, two phrase length configurations were used, namely 1–3 words and 2–3 words, to determine the effect of n-gram variations on the quality of indexing results.

Overall, the results show that the combination of YAKE and Word2Vec yields better performance than YAKE alone. This is evident in the increased precision and recall values for most books, especially when using similarity filtering ≥ 0.5. The addition of similarity measurement to book titles helps the system filter out more contextually relevant phrases. Furthermore, with phrase lengths of 1–3 words, the system performs more consistently than with 2–

3 words. While the 2–3 word configuration has a narrower scope, it produces more focused phrases that align with the manual index format.

The highest precision was recorded in the combination of YAKE and Word2Vec with a 1–3 word configuration, reaching up to 0.09 in some books. Meanwhile, the YAKE method alone produced relatively low precision, even approaching zero in some books with complex keyword structures. The cosine similarity value was the most prominent aspect of the combined method, with some books recording very high scores such as 0.9192, indicating the system's ability to capture semantic context well. This method was also able to increase recall in some cases by up to two times compared to the pure YAKE method.

In terms of time efficiency, the system demonstrates good performance for large-scale operations. Tests across all scenarios show that the average processing time is in the range of 1 to 2 seconds for YAKE-only configurations with 1–3-word phrases. Meanwhile, for trigram or combination YAKE configurations with similarity ≥ 0.5, processing time tends to increase to 3 to 7 seconds per document. While processing time increases for more complex configurations, these values are still within reasonable limits and efficient for use in real-world applications. These results indicate that the proposed system achieves a balance between semantic relevance and computational efficiency.

3.1. RESULT

Table 1. Test Results

No.	Testing Scenario	Number of Books	*Cosine Similarity*	*Precision*	*Recall*	Processing Time (seconds)
1	YAKE (1–3 words)	37	0.2592	0.0734	0.1245	1.22
2	YAKE + Word2Vec (1–3 words), Similarity ≥ 0.5	37	0.3304	0.0697	0.1046	1.56
3	YAKE (2–3 words) – 30 Best Books	30	0.2743	0.0588	0.0973	1.49
4	YAKE + Word2Vec (2–3 words), Similarity ≥ 0.5 – 30 Best Books	30	0.3216	0.0602	0.0834	1.84
5	YAKE (Unigram)	37	0.2581	0.1089	0.1633	1.15
6	YAKE + Word2Vec (Unigram), Similarity ≥ 0.5	37	0.1598	0.1041	0.0445	1.05
7	YAKE (Bigram)	37	0.2674	0.0663	0.1068	1.56
8	YAKE + Word2Vec (Bigram), Similarity ≥ 0.5	37	0.2565	0.0325	0.0222	2.34
9	YAKE (Trigram)	37	0.2697	0.0543	0.0977	1.65
10	YAKE + Word2Vec (Trigram), Similarity ≥ 0.5	37	0.2584	0.0235	0.0253	2.98
11	YAKE + Word2Vec (Unigram, Bigram, Trigram), Similarity ≥ 0.3, Top 33	37	0.3728	0.0749	0.1303	1.57
12	YAKE + Word2Vec (Unigram, Bigram, Trigram), Similarity ≥ 0.5, Top 33	37	0.3371	0.0705	0.1106	1.82

The analysis results in Table 6.1 of the 12 scenarios show that the combination of YAKE and Word2Vec with a similarity threshold ≥ 0.3 and the retrieval of the top 33 unigram, bigram, and trigram phrases (scenario 11) provided the best results in terms of cosine similarity, namely 0.3728, and a fairly high recall value of 0.1303. This indicates that the system is able to capture key words or phrases contextually and relevant to the book's content. This scenario also showed a stable precision value (0.0749), making it one of the most balanced configurations. On the other hand, the YAKE scenario with unigrams without similarity (scenario 5) recorded the highest precision of 0.1089 and recall of 0.1633, indicating that explicit single-word retrieval is quite effective despite paying less attention to semantic context. However, this method produced a lower cosine similarity value than the Word2Vec-based approach. Scenarios with high similarity (≥ 0.5) such as trigrams and bigrams tend to produce low precision and recall (e.g., a precision of 0.0235 and a recall of 0.0253 for trigrams). This is due to overly stringent similarity filters, which eliminate many relevant phrases even though they are semantically sound.

In terms of efficiency, the average processing time ranges from 1.05 to 2.98 seconds, depending on the phrase length and similarity complexity. Trigram configurations with similarity ≥ 0.5 tend to take the longest time, indicating a higher computational load on long phrases. Overall, it can be concluded that the YAKE + Word2Vec configuration with a moderate similarity threshold (≥ 0.3) and varying phrase lengths (unigrams to trigrams) provides the most optimal and balanced results, both in terms of semantic similarity, precision, recall, and system execution time.

3.2. ANALYSIS

Table 2. Table Analysis

Criteria	The Highest Score	Scenario
Cosine Similarity	**0.3728**	Scenario 11 (YAKE + Word2Vec, Top 33, Similarity ≥ 0.3, Unigram–Trigram)
Precision	**0.1089**	Scenario 5 (YAKE Unigram)
Recall	**0.1633**	Scenario 5 (YAKE Unigram)
Fastest Time	**1.05 seconds**	Scenario 6 (YAKE + Word2Vec, Unigram, Similarity ≥ 0.5)

Based on the test results of 12 scenarios, it can be concluded that the combination of the YAKE and Word2Vec methods with a configuration of top 33 phrases and a similarity threshold of ≥ 0.3 (unigram–trigram) produces the highest cosine similarity value of 0.3728, indicating the system's ability to capture the most relevant semantic meaning to the title. On the other hand, the highest precision and recall were obtained from the YAKE scenario with unigrams (without Word2Vec), namely 0.1089 for precision and 0.1633 for recall, which indicates that the pure statistical method is still superior in reaching explicit words that match the manual index. In terms of time efficiency, the YAKE + Word2Vec scenario on unigrams with a similarity threshold of ≥ 0.5 is the fastest, requiring only 1.05 seconds per document. This shows that the system is not only capable of producing contextually relevant indexes, but is also fast and efficient enough to be applied in academic and digital publishing environments. Although precision values are relatively low, this phenomenon is consistent with previous studies on automatic indexing, where manual indexes often reflect subjective editorial preferences rather than purely textual relevance.

4. CONCLUSION

The results of this study indicate that the integration of the YAKE and Word2Vec methods significantly impacts the quality of the system's automatic indexing. YAKE acts as a statistically efficient keyword extraction tool that recognizes important phrases based on local features such as frequency and position within the text. However, this approach is still limited in capturing the context or full meaning of words. Therefore, the addition of the Word2Vec model, in this case FastText Bahasa Indonesia, serves as an additional semantic layer that helps evaluate the proximity of phrases to chapter titles or topics. FastText has proven effective in handling morphological variations and technical terms often found in academic or vocational books. This makes the system more adaptive to the domain language, allowing semantically relevant phrases to be identified even if they are not lexically identical. Some books recorded high cosine similarity scores despite low precision, indicating that the system is capable of capturing meaning but does not fully match the literal manual index.

On the other hand, the recall results for some books still showed limitations. This was especially true for documents with highly specific manual indexes, rare terms, or inconsistent writing structures. This suggests that the system needs improvements in terms of synonym mapping, term normalization, and complex phrase extraction capabilities to capture a wider variety of terms. Overall, the combined YAKE and Word2Vec approach successfully improved the quality of the index contextually. The system also proved responsive with efficient processing times, supported by an interactive interface and results export feature, thus having great potential for implementation in academic and professional environments that require fast and relevant indexing.

Despite the promising results, this study has several limitations. The dataset used in the experiments was limited to 30 selected digital books, which may not fully represent the diversity of indexing patterns across different domains and publication styles. In addition, the manual indexes used as ground truth may contain subjective variations depending on editorial preferences, which can affect precision and recall measurements. This study also did not include comparisons with more advanced embedding models such as transformer-based approaches, which could potentially provide richer contextual representations. Future research may address these limitations by expanding the dataset, incorporating synonym normalization and semantic enrichment, and conducting comparative evaluations with recent transformer-based models to further improve the quality and robustness of automatic book indexing systems.

ACKNOWLEDGEMENTS

The authors would like to express their deepest gratitude to Politeknik Negeri Malang, especially the Department of Information Technology, for providing academic support and research facilities throughout the completion of this study. Special appreciation is also extended to the supervising lecturer for their valuable guidance, advice, and constructive feedback during the research process.

The authors also wish to thank fellow students and research peers for their encouragement, moral support, and insightful ideas during the development of the system and the writing of this article. Their collaboration and shared experiences have contributed greatly to the success of this research.

This research was carried out as part of an academic project to fulfill the requirements of the applied undergraduate program. All opinions, findings, and conclusions presented in this paper are solely the responsibility of the authors and do not necessarily reflect the views of the institution.

REFERENCES

Af'idah, D. I., Dairoh, D., Handayani, S. F., & Pratiwi, R. W. (2021). Pengaruh parameter Word2Vec terhadap performa deep learning pada klasifikasi sentimen. *Jurnal Informatika: Jurnal Pengembangan IT, 6*(3), 156–161. https://doi.org/10.30591/jpit.v6i3.3016

Ahmad Aliero, A., Adebayo, B. S., Aliyu, H. O., Tafida, A. G., Kangiwa, B. U., & Dankolo, N. M. (2023). Systematic review on text normalization techniques and approaches to non-standard words. *International Journal of Computer Applications, 185*(33), 1–10.

Anderson, J. D., & Pérez-Carballo, J. (2001). The nature of indexing: How humans and machines analyze messages and texts for retrieval. *Information Processing & Management, 37*(2), 231–254. https://doi.org/10.1016/S0306-4573(00)00026-1

Ariesta, B. T., Romadhony, A., & Hasmawati. (2023). Ekspansi query menggunakan Word2Vec pada pencarian artikel ilmiah. *E-Proceedings of Engineering, 10*(5), 4910–4916.

Baffy, B. (2023). *Automatic keyword extraction for a partial search engine index* (Technical report).

Caballero, A., Centeno, R., & Rodrigo, Á. (2024). LLM-based multi-agent models for multiclass classification of strategic narratives. In *CEUR Workshop Proceedings* (Vol. 3756).

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences, 509*, 257–289. https://doi.org/10.1016/j.ins.2019.09.013

Chellu, R. (2025). End-to-end documentation automation using AI and REST APIs in enterprise collaboration platforms. *Zenodo*. https://doi.org/10.5281/zenodo.15707647

Chiusano, S. A., Troncy, R., Candidato, V. B., & Spano, E. (2018). *Keyword extraction and classification* (Technical report).

Duari, S., & Bhatnagar, V. (2020). Complex network-based supervised keyword extractor. *Expert Systems with Applications, 140*, 112876. https://doi.org/10.1016/j.eswa.2019.112876

Fauzi, M. A., Arifin, A. Z., & Yuniarti, A. (2017). Arabic book retrieval using class and book index-based term weighting. *International Journal of Electrical and Computer Engineering, 7*(6), 3705–3710. https://doi.org/10.11591/ijece.v7i6.pp3705-3711

Ganapathy, A., & Fadziso, T. (2020). Intelligent indexing and sorting management system. *Engineering International, 8*(2), 101–110. https://doi.org/10.18034/ei.v8i2.554

Ghifari, M. F. (2024). *Ekstraksi kata kunci menggunakan pre-trained language model* (Undergraduate thesis).

Gupta, A., Chadha, A., & Tewari, V. (2024). A natural language processing model based on BERT and YAKE for keyword extraction on sustainability reports. *IEEE Access, 12*, 7942–7951. https://doi.org/10.1109/ACCESS.2024.3352742

Hosabettu, A. (2014). *Transformer-based extraction of statutory definitions from the U.S. Code* (Technical report).

Ilhamsyah, M. R., Subroto, I. M. I., & Haviana, S. F. C. (2023). Identifikasi kepakaran dosen menggunakan YAKE. *TRANSISTOR Elektro dan Informatika, 5*(3), 110–121.

Johnson, E., Holt, X., & Wilson, N. (2025). Improving the accuracy and efficiency of legal document tagging with large language models. *arXiv Preprint*. http://arxiv.org/abs/2504.09309

Liu, Q., Hui, Y., Liu, S., & Ji, Y. (2024). Y-Rank: A multi-feature-based keyphrase extraction method for short text. *Applied Sciences, 14*(6), 2510. https://doi.org/10.3390/app14062510

Mai, J.-E. (1999). Deconstructing the indexing process. *Knowledge Organization, 26*(3), 269–298. https://doi.org/10.1108/S0065-2830(1999)0000023013

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv Preprint*. http://arxiv.org/abs/1301.3781

Nadim, M., Akopian, D., & Matamoros, A. (2023). A comparative assessment of unsupervised keyword extraction tools. *IEEE Access, 11*, 144778–144798. https://doi.org/10.1109/ACCESS.2023.3344032

Nurdin, A., Aji, B. A. S., Bustamin, A., & Abidin, Z. (2020). Perbandingan Word2Vec, GloVe, dan FastText pada klasifikasi teks. *Jurnal Tekno Kompak, 14*(2), 74–80. https://doi.org/10.33365/jtk.v14i2.732

ONWUCHEKWA, E. O. (2024). Indexing and abstracting services. In *Fashion and Costume in American Popular Culture* (pp. 139–158). https://doi.org/10.5040/9798400650086.ch-008

Papagiannopoulou, E. (2021). *Keyphrase extraction techniques* (Doctoral dissertation).

Santosh, T. Y. S. S., Sanyal, D. K., Bhowmick, P. K., & Das, P. P. (2020). DAKE: Document-level attention for keyphrase extraction. In *Lecture Notes in Computer Science* (Vol. 12036, pp. 553–564). https://doi.org/10.1007/978-3-030-45442-5_49

Sarwar, T. B., Noor, N. M., & Miah, M. S. U. (2022). Evaluating keyphrase extraction algorithms. *PeerJ Computer Science, 8*, e1024. https://doi.org/10.7717/peerj-cs.1024

Sun, C., Hu, L., Li, S., Li, T., Li, H., & Chi, L. (2020). A review of unsupervised keyphrase extraction methods. *Symmetry, 12*(11), 1864. https://doi.org/10.3390/sym12111864

Yunmar, R. A., Setiawan, A., & Tantriawan, H. (2020). The combination of YAKE and language processing for unsupervised term extraction. *IOP Conference Series: Earth and Environmental Science, 537*(1), 012023. https://doi.org/10.1088/1755-1315/537/1/012023

Zhang, Y., Milios, E., & Zincir-Heywood, N. (2004). A comparison of keyword- and key term-based methods for automatic website summarization. *AAAI Workshop on Text and Categorization* (Technical Report WS-04-01, pp. 15–20).

Zhou, D., & Tang, X. (2020). Cross-domain keyword extraction with keyness patterns. *arXiv Preprint*. http://arxiv.org/abs/2009.04136