Received December 12th, 2024; accepted December 25th, 2024. Date of publication December 30th 2024
Digital Object Identifier: https://doi.org/10/25047/jtit.v11i2.438

Comparative Analysis of Vectorization Methods for Academic Supervisor Recommendations

QOTRUNNADA NABILA[1], ARDYTHA LUTHFIARTA[2], MUTIARA SYABILLA[3], AZIZU AHMAD ROZAKI RIYANTO[4]

[1,2,3,4]Dian Nuswantoro University, Jl. Imam Bonjol No. 207, Pendirikan Kidul, Central Semarang District, Semarang City, Central Java 50131, Indonesia

CORESPONDING AUTHOR: ARDYTHA LUTHFIARTA (email:ardytha.luthfiarta@dsn.dinus.ac.id)

ABSTRACT Selecting final project supervisors often poses challenges for students due to limited lecturer quotas and difficulties in finding suitable expertise matches. This study proposes using the Cosine Similarity method with vectorization approaches such as Bidirectional Encoder Representations from Transformers (BERT), FastText, Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word2Vec to enhance the accuracy of recommendation systems. Data sourced from Google Scholar underwent scraping, preprocessing, and vectorization to evaluate the most effective method for understanding context and recommending relevant supervisors. The analysis revealed that BERT and Word2Vec based approaches achieved superior performance, delivering a perfect hit ratio (1.00) and overcoming the limitations of TF-IDF and BoW in capturing technical language. This recommendation system is expected to streamline the supervisor selection process, minimize mismatches, and effectively support academic advisory processes across educational institutions.

KEYWORDS: cosine similarity; BERT; Vectorization Approaches; Hit Ratio; FastText

I.INTRODUCTION

The final project is one of the academic requirements that must be fulfilled by students to complete higher education. In its preparation, students need guidance from lecturers who have expertise in accordance with the topic proposed in the final project proposal. The supervisor acts as a place of consultation for students, especially in facing various obstacles in the process of working on the final project [1].

Currently, selecting a final project supervisor is typically done by directly contacting lecturers known for their expertise and alignment with the student research interests. This process is possible because the study program allows students to choose a supervisor suitable for their research topic. However, students sometimes face challenges when selecting an alternative supervisor if their preferred lecturer quota is full. This issue arises from limited information about the expertise of all lecturers within the study program. As a result, some students may choose supervisors whose expertise is less relevant to their research area, leading to suboptimal guidance and potential delays in completing their final projects [1]. The alignment between a supervisor's expertise and a student research topic is a critical factor in ensuring effective guidance and the timely completion of final projects [1]. A solution is needed that can help the selection of supervisors effectively and ensure the suitability between the lecturer's expertise and the student research theme.

One potential solution is to apply the Cosine Similarity method to recommendation systems. This method calculates the similarity between two documents to determine their relevance [2]. Previous research demonstrated that Cosine Similarity could recommend supervisors based on the similarity value obtained from the thesis query and the supervisor query. However, a study by

Ashwini Tangade et al. showed that the text rank-based Cosine Similarity approach an f-measure of only about 0.39 of the entire rouge in determining similarity [3]. In contrast, other studies have highlighted the effectiveness of Cosine Similarity when combined with different vectorization techniques. Reswara's research concluded that using various vectorization methods resulted in text similarity levels exceeding 80% [4]. Similarly, Dingding Cao demonstrated that incorporating FastText-Base vectorization with the Cosine Similarity method achieving 57% - 69% accuracy on the average cosine similarity of 4 documents. This study also recommended utilizing BERT for document similarity assessments and parameter optimization [5]. These findings suggest that incorporating advanced vectorization techniques before modeling has significant potential to enhance the accuracy of text similarity measurements. This conclusion is further supported by research from Mohamed and El-Behaidy, which found that text representation techniques are widely used and significantly improve the performance of natural language processing (NLP) tasks, including text classification [6].

BERT (Bidirectional Encoder Representations from Transformers) is proposed in this research as a more accurate alternative for text representation, as it considers the context between words in a sentence bidirectionally [7] [8]. A study conducted by Reswara (2023) demonstrated that the BERT method combined with Cosine Similarity achieved higher accuracy in providing recommendations that align with the context of the input text [4].

All of the studies revealed that various combinations of text vectorization methods with Cosine Similarity, such as TF-IDF, Bag of Words (BoW), Word2Vec (W2V), FastText, and BERT, have been widely applied in various NLP tasks. However, the effectiveness of each method for specific tasks, such as recommending supervisors based on research abstracts, is still not identified.

This study aims to compare various vectorization methods, including TF-IDF, Bag of Words (BoW), Word2Vec (W2V), FastText, and BERT for text representation using the titles and abstracts of all lecturer research available on Google Scholar. These vectorizations are then compared with the topics of students' final projects using the Cosine Similarity and hit ratio method. This research aslo to identify which the vectorization method with the highest hit ratio for aligning lecturers' research expertise with students' thesis topics. The findings aim to support the development of a supervisor selection system that improves the efficiency of the selection process. Additionally, this research seeks to contribute to developing a system that simplifies the process for students to find supervisors with relevant expertise, thereby fostering more effective academic guidance.

II.METHOD

The methodology in this case study research employs a text similarity approach using supervisors' research data, utilizing the Kaggle platform and Python programming language. The notebook specifications Kaggle used are TPU VM v3-8 with 330 GB CPU capacity and 40 GB disk memory. The data used in this study consists of the supervisors' research history sourced from Google Scholar, filtered by first and last name indices within a span of five years.

The research process begins with scraping research data from each lecturer's Google Scholar profile. The scraped data is then preprocessed through several steps, including case folding, character removal, tokenization, language detection, stopword removal, and stemming. This is followed by the embedding and text representation stages using various vectorization methods, namely TF-IDF, Bag of Words (BoW), Word2Vec (W2V), FastText, and BERT. These methods convert text into numerical representations through encoding and embedding processes, enabling analysis to determine which method best understands context and meaning learned by the model [9].

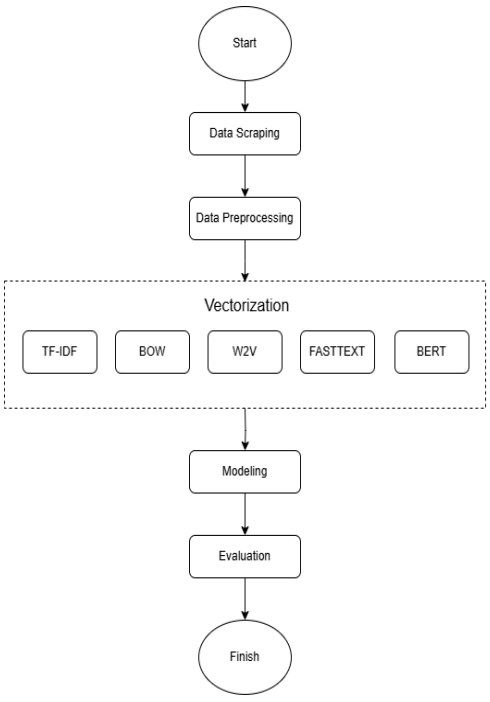

FIGURE 1. Research Methodology Diagram

Figure 1 illustrates the research methodology process in the form of a flow diagram. The final stage involves cosine similarity modeling,

where the similarity between input documents and the data documents is calculated.

A.DATA SCRAPING

Research data containing titles and abstracts of studies by supervisors on Google Scholar were scraped using Visual Studio Code with Python programming language and assisted using the beautiful soup library to automate data collection. The data collected includes the complete history of new research conducted between 2019 and 2024. The data scraping process generated information consisting of research titles, research abstracts, publication years, and author names from 71 supervisors at Dian Nuswantoro University and will be saved in csv format. The supervisor data was collected from active lecturers or supervisor in Departement Computer Science at Dian Nuswantoro University with a recorded history of serving as supervisors in the Dinus Library.

B.DATA PREPROCESSING

In Natural Language Processing (NLP), data preprocessing is performed to clean the data and prepare it for processing, ensuring better results [4]. This stage begins with language detection using the langdetect library to apply the appropriate techniques based on the language identified in the text. The process continues with case folding, where uppercase letters are converted to lowercase, followed by the removal of unnecessary elements such as extra lines, punctuation, numbers, special characters, and excessive spaces.

Subsequently, stemming and tokenization are carried out according to the detected language. Tokenization involves breaking down text into smaller units, such as words, phrases, or characters, known as tokens [10]. This step enables the model to better understand and process the text [11]. The tokenization and stemming processes vary depending on the detected language. For Indonesian texts, stopword removal and stemming are performed using the Sastrawi library. For English texts, stopword removal and lemmatization are applied using spaCy library.

The use of different approaches for stemming and lemmatization is due to the distinct structures of words and sentences in Indonesian and English language [12]. In Indonesian, root words are often concealed by affixes such as prefixes, infixes, or suffixes, requiring a word truncation process to return the word to its root form [13]. The Sastrawi Python library is specifically designed to handle the complexities of affixed Indonesian words and reduce them to their root form [14].

In contrast, English tends to have a word structure that relies more on inflectional changes based on tense and number rather than affixes [15]. The lemmatization process takes the grammatical context of a word into account to produce its base form [16]. The spaCy library, with its context-based lemmatization support, is more suitable for English as it provides higher accuracy and produces valid base forms, in contrast to stemming, which can improperly truncate words in English [17].

C.VECTORIZATION

In this research, the dataset that has been preprocessed will be vectorized using various techniques proposed in the research. Vectorization is the process of converting text data into numerical representations that can be processed by machine learning algorithms. This numerical representation is a vector consisting of a series of real or integer numbers, allowing machines to analyze relationships between words, sentences, and documents. In the context of sentiment analysis, vectorization converts text from training data into a numerical format suitable for analysis and modeling [18].

There are two main approaches to text vectorization, namely encoding, and embedding, which can facilitate the analysis and modeling process. Encoding is the initial step in converting text into numerical formats, typically producing vector representations without necessarily capturing deeper semantic meanings. In contrast, embedding provides richer and more informative representations of words in vector form, capturing the meaning and relationships between words to help models better understand their context and interconnections [19].

The vectorization methods employed include TF-IDF and Bag of Words for encoding and embedding models such as Word2Vec, FastText, and BERT for more sophisticated text representation. Each vectorization technique will process the dataset and the results between vectorization techniques will be compared using cosine and evaluation metrics.

1) TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a document vectorization method used to measure the importance of a word within a document or a corpus [20]. This method combines two principles: the frequency of a term's occurrence in a specific document (Term Frequency) and the rarity or uniqueness of the term across the entire corpus (Inverse Document Frequency) [20]. In this context, a "document" refers to a single paragraph or line of text.

TF-IDF works by multiplying the frequency of a term's occurrence within a document by the

measure of its uniqueness across the entire corpus, calculated using the formula (1).

$$TF - IDF = TF \times IDF. \quad (1)$$

With,

$$TF = \frac{Frequency\ of\ a\ term\ in\ a\ document.}{Document\ word\ count}. \quad (2)$$

$$IDF = \log\left(\frac{Dokument\ count}{Documents\ frequency}\right). \quad (3)$$

2) BAG OF WORDS

The Bag of Words (BoW) method is a text representation technique that converts a document into a set of words represented by their frequency of occurrence within the document. This approach represents text as vectors based solely on word frequency, without accounting for the relationships or context between words, treating each word as independent from the others [21].

The primary drawback of this method is its inability to capture semantic, structural, or contextual information surrounding the words. This limitation can lead to sparse vector representations and potentially result in poor model performance or overfitting, especially when the vocabulary size in the corpus is large, but word frequencies in individual documents are very low or even zero [22].

3) WORD2VEC

Word2Vec is a method introduced by Mikolov in 2013 to convert each unique word in a corpus into a vector. This technique is capable of capturing the contextual similarity between two words based on their resulting vectors [21]. Word2Vec has two primary approaches: Continuous Bag-of-Words (CBOW), which predicts the target word based on the surrounding context, and Skip-gram, which predicts the context based on the target word. Word2Vec relies on local word information within a sentence, enabling it to capture the semantic relationships between words in vector space [23].

4) FASTTEXT

FastText is a word embedding method that evolved from Word2Vec. It learns word representations by incorporating subword information, where each word is represented as a set of n-gram characters. This enables FastText to capture the meaning of short words and understand suffixes and prefixes. However, this approach has limitations in representing words from languages with extensive vocabularies and many rare words. FastText excels in several areas, such as its ability to efficiently train models on large datasets and generate representations for words not present in the training data by breaking them into n-grams to create embedding vectors [24].

5) BERT

BERT (Bidirectional Encoder Representations from Transformers) is a language model based on a bidirectional transformer architecture, trained through two main tasks: Masked Language Modeling, where some words in a sentence are masked, and the model must predict the missing words, and Next Sentence Prediction, which involves determining whether two sentences are consecutive. Through this training, BERT can understand the complex interactions between words in a sentence, resulting in contextualized and more accurate word embeddings. As shown in Figure 2, the model consists of multiple layers: 12 layers for BERT BASE and 24 layers for BERT LARGE, with embeddings that can be extracted from these layers for various applications in natural language processing [25].

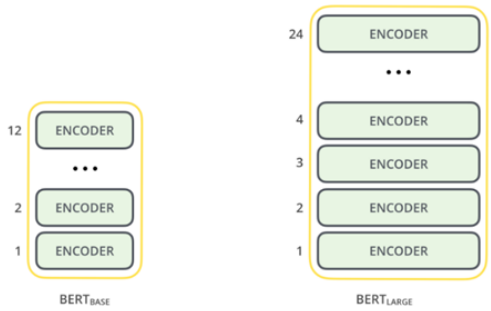

FIGURE 2. Layers in BERT

D. MODELING

The dataset will be modeled using cosine similarity to determine the similarity between the student's final project and the dataset in the research. Cosine similarity is a measure used to determine the similarity between two documents or vectors based on the angle between them in vector space. Text is converted into a numerical representation in the form of a vector through the encoding process. Cosine similarity then measures the similarity of two vectors based on the angle between them, using the Cosine Similarity formula [26].

$$Cosine\ Similarity(A, B) = \frac{A \cdot B}{\|A\|\|B\|}. \quad (4)$$

Where A·B is the dot product, and ‖A‖ and ‖B‖ are the lengths (norms) of the vectors. The result of the calculation is a value between -1 and 1, with higher values indicating greater similarity. In many applications, particularly those involving text, cosine similarity values are typically calculated only for vectors that do not contain negative elements (e.g., TF-IDF results or embeddings), which restricts the range of values from 0 to 1 [27]. The primary advantage of the cosine similarity method is that it is unaffected by the length of a document. This means that two documents can be considered similar, even if they do not share the same terms, as long as the vectors representing the documents have

the same or similar direction [28]. In text document analysis, the text is generally converted into vectors using a vectorization model before calculating similarity. Afterward, cosine similarity is applied to measure the similarity between the two vectors. Research has demonstrated that cosine similarity often outperforms other methods, such as the Euclidean distance [29].

E. EVALUATION

The hit ratio evaluation method will be used in this study to measure the optimality of vectorization and cosine similarity techniques on the dataset. The hit ratio is an evaluation metric used to assess a recommendation system. It calculates the ratio of matches between the recommended items and those actually present in the test data. The hit ratio ranges from 0 to 1, where 0 indicates no matches (completely incorrect), and 1 indicates that all possible matches are predicted (completely correct). Values between 0 and 1 represent the proportion of partially correct matches [30]. This metric is simple yet effective, as it focuses on the presence of relevant items without considering the order or quantity of irrelevant items in the recommendation.

In this study, the hit ratio is modified with two additional evaluation criteria to provide a more comprehensive assessment of recommendation quality. The experiment is considered successful if at least three relevant items appear in the Top-5 recommendation list or if the cosine similarity between the recommendation and the ground truth exceeds a threshold of 0.90. The hit ratio (HR) is calculated as the ratio of successful trials to the total number of trials, as shown in the following formula:

$$HR = \frac{Successful\ trial}{Total\ trials}. \quad (5)$$

In this context, a successful trial is defined as one that contains at least three relevant items in the Top-5 list or has a cosine similarity that exceeds a predefined threshold value. This level of accuracy serves as a key indicator of the system's success in providing relevant recommendations.

III.RESULT AND DISCUSSION

The data scraping process successfully gathered 1,070 research records from supervisors, consisting of titles and abstracts, and saved them on CSV format. These records include 638 in English, 430 in Indonesian, and 2 in other languages. The distribution of this data language can be seen in Figure 3. The classification of some records as "other languages" may stem from the extensive use of foreign terms or technical jargon within the content.

To address the disparity in data volume across languages, all text was translated into Indonesian and English using DeepTranslator with Google Translator. This process resulted in a total of 2,140 entries comprising titles and abstracts in both languages. Standardizing the input language ensures the system can provide recommendations more effectively without being limited by language variations. This translation step is essential for improving the system's ability to filter information and deliver accurate recommendations.

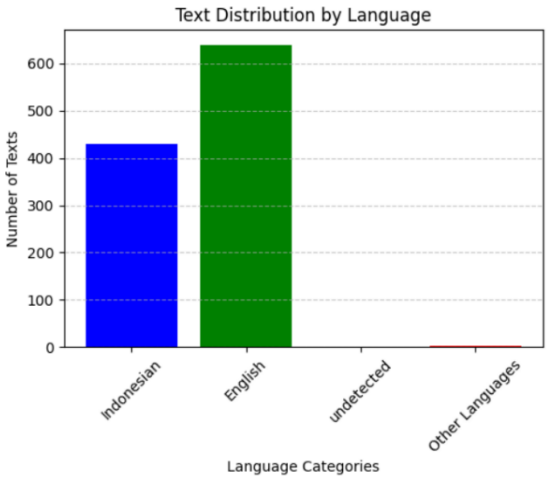

FIGURE 3. Distribution of Texts by Language

The next step involves data preprocessing, which includes text cleaning tasks such as removing symbols, numbers, and irrelevant words based on the detected language. The results from each stage of this preprocessing, as summarized in Table 1, then serve as the input for the modeling process.

TABLE 1. Preprocessing Data

Raw Data			
Mengembangkan Konsep & Strategi Smart Regional: Cara Meningkatkan Pariwisata & Investor (Smart City 4.0)			
Step	Preprocessing Data in English	Step	Preprocessing Data in Indonesian
case folding	developing the concepts & strategy of smart regional: how to increase tourism & investors (smart city 4.0)	case folding	mengembangkan konsep & strategi smart regional: cara meningkatkan pariwisata & investor (smart city 4.0)
removing character	developing the concepts strategy of smart regional how to increase tourism investors smart city	removing character	mengembangkan konsep strategi smart regional cara meningkatkan pariwisata investor smart city
tokenizer	['developing', 'the', 'concepts', 'strategy', 'of', 'smart', 'regional', 'how', 'to', 'increase', 'tourism', 'investor', 'smart', 'city']	tokenizer	['mengembangkan', 'konsep', 'strategi', 'smart', 'regional', 'cara', 'meningkatkan', 'pariwisata', 'investor', 'smart', 'city']

lematization and stopword	['develop', 'concept', 'strategy', 'smart', 'regional', 'increase', 'tourism', 'investor', 'smart', 'city']	steaming and stopword	['kembang', 'konsep', 'strategi', 'smart', 'regional', 'tingkat', 'pariwisata', 'investor', 'smart', 'city']

results of english data	results of indonesian data
develop concept strategy smart regional increase tourism investor smart city	kembang konsep strategi smart regional tingkat pariwisata investor smart city

Table 1 shows that after the dataset goes through the preprocessing stage, it becomes more structured and neat, whereas previously, it was still unstructured, with many symbols, uppercase, and others. After the preprocessing stage, this study tested several vectorization methods, namely TF-IDF, BOW, W2V, fastText, and BERT, to measure the accuracy of prediction using cosine similarity and relevance of supervisor names based on test data. The test results from 10 experiments are presented in Tables 2, 3, and 4, with additional criteria labels as a reference for the success of hit ratio calculation.

TABLE 2. TF-IDF and BOW Vectorization Results

Trial	Vectorization			
	TF-IDF		BOW	
	Similarity	Relevant Supervisor	Similarity	Relevant Supervisor
1	0.332 (Fail)	3 (Relevan)	0.356 (Fail)	3 (Relevan)
2	0.422 (Fail)	2 (Unrelevant)	0.426 (Fail)	2 (Unrelevant)
3	0.353 (Fail)	2 (Unrelevant)	0.384 (Fail)	3 (Relevan)
4	0.430 (Fail)	3 (Relevan)	0.426 (Fail)	2 (Unrelevant)
5	0.287 (Fail)	2 (Unrelevant)	0.294 (Fail)	2 (Unrelevant)
6	0.199 (Fail)	3 (Relevan)	0.201 (Fail)	3 (Relevan)
7	0.242 (Fail)	1 (Unrelevant)	0.235 (Fail)	2 (Unrelevant)
8	0.256 (Fail)	2 (Unrelevant)	0.278 (Fail)	3 (Relevan)
9	0.315 (Fail)	3 (Relevan)	0.320 (Fail)	2 (Unrelevant)
10	0.564 (Fail)	1 (Unrelevant)	0.575 (Fail)	0 (Unrelevant)

TABLE 3. W2V and FastText Vectorization Results

Trial	Vectorization			
	W2V		FastText	
	Similarity	Relevant Supervisor	Similarity	Relevant Supervisor
1	0.988 (Succes)	2 (Unrelevant)	0.934 (Succes)	2 (Unrelevant)
2	0.986 (Succes)	2 (Unrelevant)	0.953 (Succes)	2 (Unrelevant)
3	0.981 (Succes)	1 (Unrelevant)	0.921 (Succes)	1 (Unrelevant)
4	0.986 (Succes)	2 (Unrelevant)	0.886 (Fail)	3 (Relevant)
5	0.987 (Succes)	2 (Unrelevant)	0.93 (Succes)	2 (Unrelevant)
6	0.988 (Succes)	3 (Relevan)	0.838 (Fail)	1 (Unrelevant)
7	0.989 (Succes)	1 (Unrelevant)	0.919 (Succes)	3 (Relevant)
8	0.98 (Succes)	3 (Relevan)	0.928 (Succes)	1 (Unrelevant)
9	0.989 (Succes)	4 (Relevan)	0.924 (Succes)	3 (Relevant)
10	0.992 (Succes)	0 (Unrelevant)	0.881 (Fail)	2 (Unrelevant)

TABLE 4. BERT Vectorization Results

Trial	Vectorization	
	BERT	
	Similarity	Relevant Supervisor
1	0.961 (Succes)	4 (Relevant)
2	0.977 (Succes)	3 (Relevant)
3	0.965 (Succes)	3 (Relevant)
4	0.969 (Succes)	3 (Relevant)
5	0.901 (Succes)	2 (Unrelevant)
6	0.972 (Succes)	4 (Relevant)
7	0.970 (Succes)	3 (Relevant)
8	0.968 (Succes)	4 (Relevant)
9	0.968 (Succes)	3 (Relevant)
10	0.966 (Succes)	3 (Relevant)

In the experiments conducted which can be seen in Tables 2, 3, and 4, the TF-IDF and BOW models did not achieve similarity accuracy of more than 0.90 in 10 experiments, with only four experiments resulting in relevant lecturers. Instead, the W2V model showed perfect similarity accuracy even though it only identified three experiments of relevant lecturers. The fastText model did slightly better, with two experiments having a similarity of less than 0.90 and identifying less than three relevant lecturer experiments. Finally, the BERT model produced near-perfect similarity accuracy, despite there being only one irrelevant lecturer experiment.

To show the performance of each method in providing relevant and accurate recommendations. Evaluation is carried out using a hit ratio that is adjusted to the requirements of the successful similarity results and the number of relevant name that have been described in the previous table so that, it can be seen the ratio results of each method in Table 5.

TABLE 5. Hit Ratio Results

Hit Ratio				
TF-IDF	BOW	W2V	FastText	BERT
0.4	0.4	1.00	0.8	1.00

The results shown in Table 5 indicate that the W2V and BERT methods perform best with a hit ratio of 1.00, followed by FastText with 0.8, and BoW and TF-IDF with a hit ratio of 0.4. Although W2V shows a high score in cosine similarity in Table 3, BERT is still superior due to its ability to understand the overall context of the sentence. This makes BERT a better choice for providing relevant and accurate recommendations, as evidenced by the superior relevance in all BERT experiments, compared to some trials with W2V.

The BoW and TF-IDF methods proved less effective in handling research abstracts containing technical language and unfamiliar terms, which are better handled by context-based methods such as BERT. The poor performance of TF-IDF and BoW is due to their limitations in understanding semantic context, where TF-IDF only calculates word frequency weights, and BoW does not take into consideration word order, making it less able to handle relationships between concepts or technical terms in the text.

IV.CONCLUSION

Overall, the results demonstrated that, with the aid of a combination of preprocessing methods such as data balancing, case folding, and lemmatization/stemming, the BERT method proved to be the most effective for the supervisor recommendation system, achieving a hit ratio of 1.00. This method successfully predicted more than two supervisor names relevant to the selected research topic from ten studies, yielding an average accuracy of 0.961 using the cosine similarity model. W2V and FastText also yielded efficient results, with cosine similarity accuracies of 0.986 and 0.911, respectively. While W2V offers better recommendation quality, it is less accurate in providing precise recommendations, as demonstrated by only three correct trials out of ten. In contrast, TF-IDF and BoW are more suitable for simpler tasks that do not require complex semantic analysis.

BERT excels in capturing deep semantic context, providing a substantial advantage over traditional vectorization methods. These findings have practical implications for real-world academic settings, particularly in integrating the system into university platforms to streamline supervisor selection processes.

However, its large-scale implementation faces challenges, including high computational demands and scalability constraints. Utilizing BERT requires advanced hardware such as GPUs or TPUs, which can be a significant limitation for institutions with restricted resources, especially when working with real-time systems or large datasets.

Future work could explore BERT's ability to capture deep semantic context, which may significantly improve the system's accuracy. Optimizing the model through pruning or quantization and leveraging distributed computing should be prioritized to ensure scalability. Expanding the dataset by incorporating additional data, such as student thesis results, to boost the model's generalization capability. Moreover, evaluating the quality of recommendations through user feedback would offer valuable insights into their relevance and effectiveness. Lastly, experimenting with topic clustering or integrating other key variables could significantly enhance the accuracy of predicting supervisor names, making the recommendation system more reliable and precise.

REFERENCE

[1] G. I. Sampurno, A. Annisa, and S. H. Wijaya, "Sistem Rekomendasi Dua Arah untuk Pemilihan Dosen Pembimbing Menggunakan Data Histori dan Skyline View Queries," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 5, p. 1055, Oct. 2022, doi: 10.25126/JTIIK.2022955458.

[2] C. Totla, T. Shah, K. Shah, and P. Tawde, "A Proposed Approach to Check Project Idea Similarity Using Topic Modelling," *2021 7th IEEE International Conference on Advances in Computing, Communication and Control, ICAC3 2021*, 2021, doi: 10.1109/ICAC353642.2021.9697169.

[3] A. Tangade *et al.*, "The Power of Pre-trained Transformers for Extractive Text Summarization: An Innovative Approach," *2023 11th International Symposium on Electronic Systems Devices and Computing, ESDC 2023*, 2023, doi: 10.1109/ESDC56251.2023.10149858.

[4] C. G. Reswara, J. Nicolas, M. Ananta, and F. I. Kurniadi, "Anime Recommendation System Using Bert and Cosine Similarity," in *2023 4th International Conference on Artificial Intelligence and Data Sciences: Discovering Technological Advancement in Artificial Intelligence and Data Science, AiDAS 2023 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 109–113. doi: 10.1109/AiDAS60501.2023.10284693.

[5] D. Cao and M. K. Chan, "Enhancing Chemical Synthesis Research with NLP: Word Embeddings for Chemical Reagent Identification - A Case Study on nano-FeCu," *iScience*, vol. 27, no. 10, p. 110780, Oct. 2024, doi: 10.1016/j.isci.2024.110780.

[6] E. H. Mohamed and W. H. El-Behaidy, "An Ensemble Multi-label Themes-Based Classification for Holy Qur'an Verses Using Word2Vec Embedding," *Arab J Sci Eng*, vol. 46, no. 4, pp. 3519–3529, Apr. 2021, doi: 10.1007/s13369-020-05184-0.

[7] A. D. P. Ariyanto, Chastine fatichah, and Agus Zainal Arifin, "Analisis Metode Representasi Teks Untuk Deteksi Interelasi Kitab Hadis: Systematic Literature Review," *Jurnal RESTI (Rekayasa Sistem dan*

Teknologi Informasi), vol. 5, no. 5, pp. 992–1000, Oct. 2021, doi: 10.29207/resti.v5i5.3499.

[8] B. Sunarko, U. Hasanah, and S. Hidayat, "Enhancing Restaurant Customer Review Analysis: Multi-Class Text Classification with BERT," *6th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2023 - Proceeding*, pp. 501–506, 2023, doi: 10.1109/ISRITI60336.2023.10467438.

[9] D. Rani, R. Kumar, and N. Chauhan, "Study and Comparision of Vectorization Techniques Used in Text Classification," *2022 13th International Conference on Computing Communication and Networking Technologies, ICCCNT 2022*, 2022, doi: 10.1109/ICCCNT54827.2022.9984608.

[10] A. Rahali and M. A. Akhloufi, "End-to-End Transformer-Based Models in Textual-Based NLP," *AI 2023, Vol. 4, Pages 54-110*, vol. 4, no. 1, pp. 54–110, Jan. 2023, doi: 10.3390/AI4010004.

[11] B. Kumar, Sheetal, V. S. Badiger, and A. Ds. Jacintha, "Sentiment Analysis for Products Review based on NLP using Lexicon-Based Approach and Roberta," *Proceedings of the 2nd International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics, ICIITCEE 2024*, 2024, doi: 10.1109/IITCEE59897.2024.10468039.

[12] K. Divya, B. Siddhartha, N. Niveditha, and Y. Manu, "An Interpretation of Lemmatization and Stemming in Natural Language Processing," *Journal of University of Shanghai for Science and Technology*, vol. 22, no. 10, p. 350, Oct. 2021, [Online]. Available: https://www.researchgate.net/publication/348306833

[13] M. Maulana, F. Ramadhan, and T. Z. A. Khan, "Identification of the Affixation Process for Advertisements as a Medium for Learning Indonesian for Class IV SD/MI," in *Proceedings of the 3rd International Conference on Education for All (ICEDUALL 2023)*, H. J. Prayitno, Ed., 2024, pp. 95–106. doi: 10.2991/978-2-38476-226-2_10.

[14] R. D. Himawan and Eliyani, "Perbandingan Akurasi Analisis Sentimen Tweet terhadap Pemerintah Provinsi DKI Jakarta di Masa Pandemi," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 7, no. 1, 2021.

[15] L. F. Donnelly, R. Grzeszczuk, and C. V. Guimaraes, "Use of Natural Language Processing (NLP) in Evaluation of Radiology Reports: An Update on Applications and Technology Advances," *Seminars in Ultrasound, CT and MRI*, vol. 43, no. 2, pp. 176–181, Apr. 2022, doi: 10.1053/J.SULT.2022.02.007.

[16] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, "A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, Jun. 2021, doi: 10.1145/3434237.

[17] R. Widagsa, S. Wiyanah, and P. Wahyuni, "THE INFLUENCE OF INDONESIAN PROSODIC FEATURES ON ENGLISH WORD STRESS PRODUCTION," *English Review: Journal of English Education*, vol. 7, no. 2, p. 77, Jun. 2019, doi: 10.25134/erjee.v7i2.1647.

[18] "Study of the Application of Logistic Regression and Naïve Bayes Algorithms for Automatic Classification of User Reviews in Bulgarian | IEEE Conference Publication | IEEE Xplore." Accessed: Dec. 16, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10757274

[19] D. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of word embedding models on text analytics in deep learning environment: a review," *Artificial Intelligence Review 2023 56:9*, vol. 56, no. 9, pp. 10345–10425, Feb. 2023, doi: 10.1007/S10462-023-10419-1.

[20] G. Mitrov, B. Stanoev, S. Gievska, G. Mirceva, and E. Zdravevski, "Combining Semantic Matching, Word Embeddings, Transformers, and LLMs for Enhanced Document Ranking: Application in Systematic Reviews," *Big Data and Cognitive Computing*, vol. 8, no. 9, p. 110, Sep. 2024, doi: 10.3390/bdcc8090110.

[21] M. Parmar and A. Tiwari, "Enhancing Text Classification Performance using Stacking Ensemble Method with TF-IDF Feature Extraction," *Proceedings - 2024 5th International Conference on Mobile Computing and Sustainable Informatics, ICMCSI 2024*, pp. 166–174, 2024, doi: 10.1109/ICMCSI61536.2024.00031.

[22] N. Chayangkoon and A. Srivihok, "Text classification model for methamphetamine-related tweets in Southeast Asia using dual data preprocessing techniques," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 4, pp. 3617–3628, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3617-3628.

[23] H. Zhang, A. M. Kassim, N. H. Samsudin, L. Teng, and C. Y. Tang, "A Hybrid Deep Learning Framework for Hotel Rating Systems: Integrating Word2Vec, TF-IDF, and Bi-LSTM With Attention Mechanism," *IEEE Trans Comput Soc Syst*, 2024, doi: 10.1109/TCSS.2024.3461796.

[24] E. S. Qorina, A. Zamhari, H. Hasan, K. Hulliyah, and D. Saepudin, "Comparative Analysis of the Performance of the Fasttext and Word2vec Methods on the Semantic Similarity Query of Sirah Nabawiyah Information Retrieval System: A systematic literature review," *2020 8th International Conference on Cyber and IT Service Management, CITSM 2020*, Oct. 2020, doi: 10.1109/CITSM50537.2020.9268827.

[25] J. Wang *et al.*, "Utilizing BERT for Information Retrieval: Survey, Applications, Resources, and Challenges," *ACM Comput Surv*, vol. 56, no. 7, Jul. 2024, doi: 10.1145/3648471/ASSET/ADB95A7D-9DA3-4F24-A3DF-2A396CED6ADD/ASSETS/GRAPHIC/CSUR-2021-0456-F12.JPG.

[26] C. Totla, T. Shah, K. Shah, and P. Tawde, "A Proposed Approach to Check Project Idea Similarity Using Topic Modelling," *2021 7th IEEE International Conference on Advances in Computing, Communication and Control, ICAC3 2021*, 2021, doi: 10.1109/ICAC353642.2021.9697169.

[27] P. Chowdhury and B. B. Sinha, "Evaluating the Effectiveness of Collaborative Filtering Similarity Measures: A Comprehensive Review," *Procedia Comput Sci*, vol. 235, pp. 2641–2650, Jan. 2024, doi: 10.1016/J.PROCS.2024.04.249.

[28] S. P. Faransyah *et al.*, "IMPLEMENTASI CASE BASE REASONING MENGGUNAKAN METODE COSINE SIMILARITY UNTUK MENDIAGNOSA PENYAKIT PADA SAPI," *J-ICON*, vol. 6, no. 2, pp. 47–52, 2018.

[29] J. Gaura and E. Sojka, "Normalised diffusion cosine similarity and its use for image segmentation," in *ICPRAM 2015 - 4th International Conference on Pattern Recognition Applications and Methods, Proceedings*, SciTePress, 2015, pp. 121–129. doi: 10.5220/0005220601210129.

[30] X. He, T. Chen, M. Y. Kan, and X. Chen, "TriRank: Review-aware explainable recommendation by modeling aspects," *International Conference on Information and Knowledge Management, Proceedings*, vol. 19-23-Oct-2015, pp. 1661–1670, Oct. 2015, doi: 10.1145/2806416.2806504.

QOTRUNNADA NABILA was born in Batang. She is a last-year student of Computer Science at Dian Nuswantoro University (UDINUS) in Semarang.
She is actively contributing as a Research Assistant in "Bengkel Koding", a program under the Departement of Computer Science at UDINUS. Her research interests primarily focus on Machine Learning and Natural Language Processing (NLP).

ARDYTHA LUTHFIARTA was born in Semarang. He earned a Master's degree in Software Engineering and Intelligent Systems from Universiti Teknikal Malaysia Melaka in Malaysia. His major field of study is artificial intelligence and intelligent systems.
He is currently a Lecture at the Informatics Engineering Program, Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia.
He has been actively engaged in research and education. His expertise spans Artificial Intelligence, Data Mining, Natural Language Processing, and Deep Learning. Additionally, he has contributed to academic publications in these fields. Through impactful publications and collaborative projects, he continues to shape advancements that redefine the role of technology in society.

MUTIARA SYABILLA was born in Jepara. She is a last-year student of Computers Science at Universitas Dian Nuswantoro (UDINUS) in Semarang.
She is actively contributing as a Research Assistant in "Bengkel Koding", a program under the Departement of Computer Science at UDINUS. Her research interests primarily focus on Image Processing and Natural Language Processing (NLP).

AZIZU AHMAD ROZAKI R. was born in Semarang. He last year student of the bachelor of computer science at Dian Nuswantoro University Semarang Indonesia.
He is currently an exchange student at 1Universiti Teknikal Malaysia Melaka for one semester doing research project about artificial intelligence. He is also a research assistant of Dinus Research Group for AI in Medical Science (DREAMS) and Society, Agriculture, and Food Advancement Research (SAFAR) at Dian Nuswantoro University.