Jurnal Teknik Informatika (JUTIF) Vol. 7, No. 1, February 2026, Page. 587-605
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5370

Optimizing Bag of Words and Word2Vec with Vocabulary Pruning and TF-IDF Weighted Embeddings for Accurate Chatbot Responses in Indonesian Treasury Services

Eko Aprianto*[1], Deni Mahdiana[2], Arief Wibowo[3]

[1,2,3]Faculty of Information Technology, Universitas Budi Luhur, Jakarta, Indonesia

Email: [1]2311600882@student.budiluhur.ac.id

Received : Oct 13, 2025; Revised : Nov 17, 2025; Accepted : Nov 17, 2025; Published : Feb 15, 2026

Abstract

The high volume of support tickets submitted to the HAI DJPb Service Desk has caused delays and inconsistent response quality in payroll-related inquiries across Indonesian treasury work units (Satker). To improve the accuracy and efficiency of public service responses, this research proposes an optimized text-vectorization framework for chatbot development using a hybrid combination of Bag of Words (BoW), Word2Vec, vocabulary pruning, and TF-IDF weighted embeddings. The dataset consists of 2024 ticket logs, curated FAQs, and questionnaire data related to the Satker Web Payroll Application. The method includes preprocessing (snippet removal, normalization, tokenization, stopword removal, stemming), vocabulary pruning based on empirical frequency thresholds (<5 and >80) while preserving domain-specific technical terms, and semantic weighting through TF-IDF. Four vectorization models—BoW, BoW with pruning, Word2Vec, and Word2Vec + TF-IDF—were evaluated using cosine similarity, response time, and accuracy. Results show that BoW achieved the highest accuracy of 88.32%, while Word2Vec produced the most stable response time with an average of 47.32 ms and a cosine similarity of 0.99. The findings demonstrate that frequency-based representations remain highly effective for structured administrative datasets, while weighted embeddings improve semantic relevance. This study contributes to the field of Informatics by providing an efficient hybrid vectorization framework tailored for Indonesian administrative language, enabling more accurate and scalable chatbot solutions for e-government services.

Keywords : *Bag Of Words, Chatbot Development, TF-IDF Weighting, Vocabulary Pruning, Word2Vec*

1. INTRODUCTION

Digital transformation in public financial administration has encouraged the Directorate General of Treasury (DJPb) to strengthen its information service ecosystem through the HAI DJPb Service Desk as the central hub for resolving system-related issues [1]. The Satker Web Payroll Application, as one of the critical components within this ecosystem, supports real-time payroll processing and requires timely and accurate assistance for users across Indonesia [2]. However, the increasing ticket volume, limited service agents, and repetitive inquiries have resulted in delays that negatively impact response efficiency and user satisfaction [3]. Chatbot-based automation has emerged as a promising solution for improving the responsiveness and consistency of public service communication [4].

Research on vectorization techniques shows that Bag of Words (BoW) remains effective for structured and repetitive datasets despite its limited semantic capability [5]. Word2Vec, on the other hand, provides dense distributed representations that better capture contextual meaning in natural language processing tasks [6]. Hybrid approaches integrating TF-IDF with semantic embeddings have demonstrated improved performance for intent classification and short-text retrieval tasks [7]. Vocabulary pruning has also been explored to reduce noise by filtering overly frequent or rare terms to improve model robustness [8]. Studies on chatbot applications highlight the importance of high-quality training data for achieving accurate automated responses [9]. Domain-specific training has been shown to significantly enhance chatbot performance in specialized environments such as academic information systems and public institutions [10].

Existing research on Indonesian NLP has focused on word embeddings, deep learning models, and hybrid vectorization approaches, but these studies are primarily developed for general conversational datasets rather than administrative texts [11]. Indonesian administrative language exhibits unique patterns characterized by formal expressions, technical terminology, and repetitive question structures that differ significantly from open-domain corpora [12]. Furthermore, prior work on chatbot development has not extensively evaluated the combined impact of BoW, pruning thresholds, Word2Vec, and TF-IDF weighting on domain-specific government service data [13]. The integration of vectorization optimization with real-world ticket logs remains underexplored, particularly for improving the accuracy of automated responses in e-government contexts [14]. Evaluations comparing multiple vectorization models for Indonesian public service chatbots are still limited and lack systematic benchmarking using real-world datasets [15].

Considering these limitations, this research develops and compares four vectorization models—Bag of Words, Bag of Words with vocabulary pruning, Word2Vec, and Word2Vec with TF-IDF weighting—to identify the most accurate model for chatbot-based response generation in Indonesian Treasury services [16]. The dataset used consists of support tickets, curated FAQs, and questionnaire data related to payroll processes, providing comprehensive coverage of user inquiries [17]. Each vectorization method is evaluated using cosine similarity, response time, and accuracy benchmarks to obtain a holistic performance assessment [18]. The pruning technique employed in this study removes words with extremely low or high frequency while retaining domain-specific terms to improve discriminative power [19]. Weighted embeddings are used to enhance semantic relevance by combining Word2Vec representations with TF-IDF-based importance scores [20].

The contribution of this research lies in delivering an optimized vectorization framework tailored for Indonesian administrative language, supported by empirical evidence from real-world government service data [21]. This work enriches the field of Informatics by demonstrating how hybrid vectorization, pruning, and weighted embeddings can significantly improve chatbot accuracy in structured public service environments [22]. The findings also provide practical insights for developing scalable and efficient NLP-based systems that support digital governance in Indonesia [23]. This study further establishes a reproducible evaluation pipeline that can be adopted for future research in e-government AI applications [24]. By addressing the unique linguistic characteristics of Indonesian administrative text, the proposed model supports more accurate information retrieval for high-volume public service systems [25]. The methodological framework presented here contributes to advancing computational models for low-resource and domain-specific languages in Southeast Asia [26]. Ultimately, this research helps accelerate the adoption of intelligent automation within government institutions through dependable and accurate chatbot technology [27].

2. METHOD

Based on Figure 1, the chatbot development process consists of six main stages, starting from problem identification and data sources, data preparation, up to modeling and evaluation. This process involves the application of vectorization techniques such as Bag of Words and a combination of Word2Vec and TF-IDF, which are optimized with vocabulary pruning and weighted embeddings. Evaluation is carried out using metrics like cosine similarity, response time, and accuracy, before the chatbot is finally integrated into the system via an API with a simple mockup.

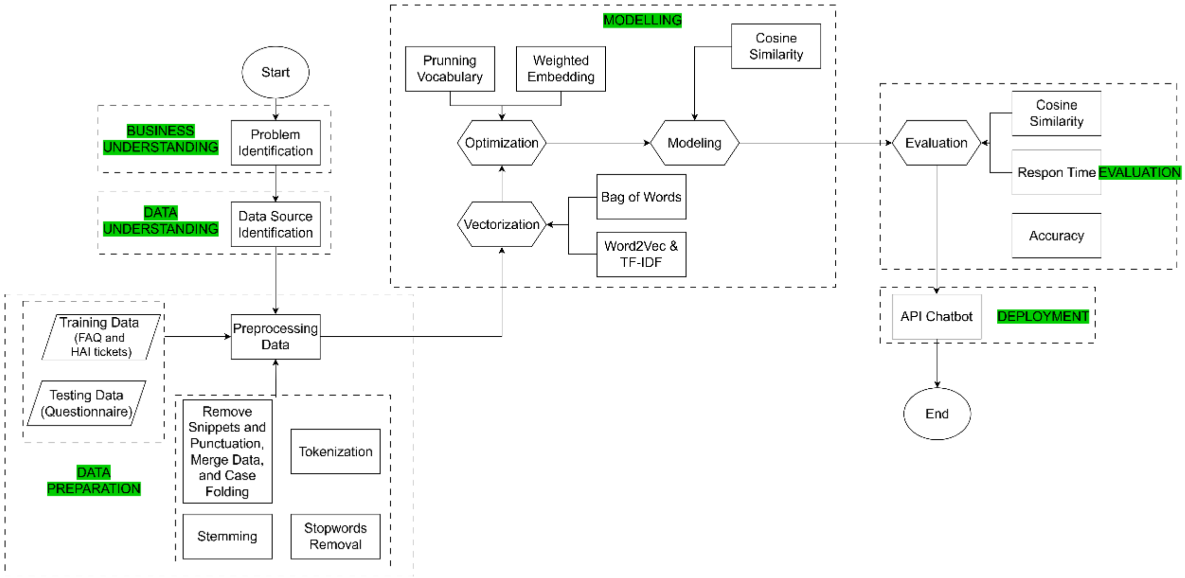


Figure 1. Research Methodology

2.1. Business Understanding

In an effort to support bureaucratic reform and improve public services, the Directorate General of Treasury (DJPb) developed the Service Desk HAI DJPb as the official help center for users of the treasury system. One of the main communication channels is via email, which is then automatically converted into tickets. However, the increasing volume of tickets, especially recurring ones like questions related to the Satker Web Payroll Application, leads to a high workload for a limited number of agents. This results in response delays and a need for escalation to developers. To address these challenges, a chatbot solution was developed that is capable of providing automatic responses to common questions, thereby accelerating service, reducing the agents' workload, and improving the service quality of HAI DJPb. Therefore, the text-vectorization optimization becomes a critical component in designing an effective chatbot model.

2.2. Data Collection

The dataset used in this research was collected from three primary sources that represent actual user interactions with the HAI DJPb Service Desk related to the Satker Web Payroll Application. These sources include: (1) historical ticket logs, (2) curated Frequently Asked Questions (FAQ), and (3) questionnaire responses from system users. The combination of these datasets ensures comprehensive coverage of both frequently occurring and newly emerging inquiries.

The first dataset is the HAI DJPb ticket log containing one year of archived conversations from January to December 2024. This dataset includes question–answer pairs submitted by work units (Satker) through the official email-based ticketing system. A total of *X* ticket entries (to be filled with actual numbers) were extracted in Excel format (.xlsx), consisting of user questions, agent responses, timestamps, and metadata. Before use, all personally identifiable information (PII) such as names, positions, and contact details was anonymized to protect user privacy.

The second dataset is the official FAQ document, compiled by DJPb's technical team, containing structured and validated question–answer pairs related to payroll access, authorization issues, form errors, data validation, and general troubleshooting. This dataset consists of *Y* FAQ entries and was provided in JSON format. The FAQ serves as a reliable knowledge base that strengthens the chatbot's capability to answer recurring and well-defined inquiries.

Table 1. Sample Data for the HAI DJPb Ticket

Ticket ID	Question	Answer
184860220240101-NE8IY4archived	*Selamat Pagi Bapak/Ibu, kami dari Polres Kepulauan Aru Polda Maluku ingin bertanya terkait SP2D gaji kami di Polres Kepulauan Aru belum keluar SP2D, mohon petunjuk dan arahan*	*Yth. Bapak/Ibu polres aruTerima kasih telah menggunakan layanan Helpdesk Terintegrasi HAI DJPb. Sehubungan dengan pertanyaan Bapak/Ibu polres aru, Dapat kami sampaikan jawaban sebagai berikut :Untuk mempermudah dalam penyelesaian masalah Anda, dapat kami sampaikan informasi sebagai berikut :1. Mohon informasi nomor SPM yang dimaksud.2. Mohon melampirkan data pendukung permasalahan Anda.3. Mohon informasi enam digit kode satker anda. Demikian informasi ini kami sampaikan semoga dapat membantu. Terima kasih.*

Table 2. Sample Data for FAQ

Question	Answer
Mengapa saat menyimpan formulir permintaan akses terdapat error?	*Pastikan besaran file yang diupload tidak lebih dari 2 Mb dengan nama file tidak terlalu panjang. Pastikan juga tidak ada isian yang kosong seperti nama satker atau kode kppn ataupun nip.*
Mengapa ada keterangan unauthorized atau tidak memiliki otorisasi saat mengakses aplikasi gaji?	*user unauthorized atau tidak memiliki otorisasi terjadi karena user belum diijinkan untuk mengakses aplikasi gaji namun sudah berhasil melakukan pendaftaran. silahkan hubungi admin KPPN untuk dapat disetujui permohonan aksesnya.*
Mengapa terdapat pesan nip tidak ditemukan di BKN pada Informasi Umum Pegawai?	*silahkan klik tanda kaca pembesar di sebelah field nip untuk mengecek apakah nip yang direkam ada di data BKN. Pastikan jenis pegawai yang digunakan adalah PNS/PNS Polri/PNS TNI.*

Table 3. Sample Data for Quistionnaire

No	Question
1	*Bagaimana cara mengatasi status error hubungi admin di aplikasi Gaji Web Satker*
2	*Bagaimana cara mengatasi gagal validasi SKPP karena data gaji terakhir tidak sama dengan database KPPN?*
3	*bagaimana cara reset user?*
4	*Kenapa cetak laporan baik gaji maupun perubahan statusnya dalam proses terus, dan membutuhkan waktu yang sangat lama?*
5	*Pada pembuatan kekurangan Tukin pada Gaji web terkadang tertolak sistem*

The third dataset comes from a user questionnaire distributed to Satker operators to identify unresolved issues not captured in the ticket logs. The questionnaire contains *Z* open-ended questions submitted in text format, without predefined answers. This dataset is used solely as testing data to evaluate the generalization ability of the vectorization models when encountering new and unseen questions.

All datasets were consolidated and standardized into a uniform JSON structure containing three main fields: "question," "answer," and "source." Since the data originates from operational government systems, strict data handling procedures were applied, including anonymization, removal of sensitive attributes, and compliance with DJPb internal data governance policies. This multi-source collection strategy ensures that the dataset reflects real user language, technical terminology, and administrative patterns relevant to the Indonesian Treasury environment.

2.3. Data Preparation

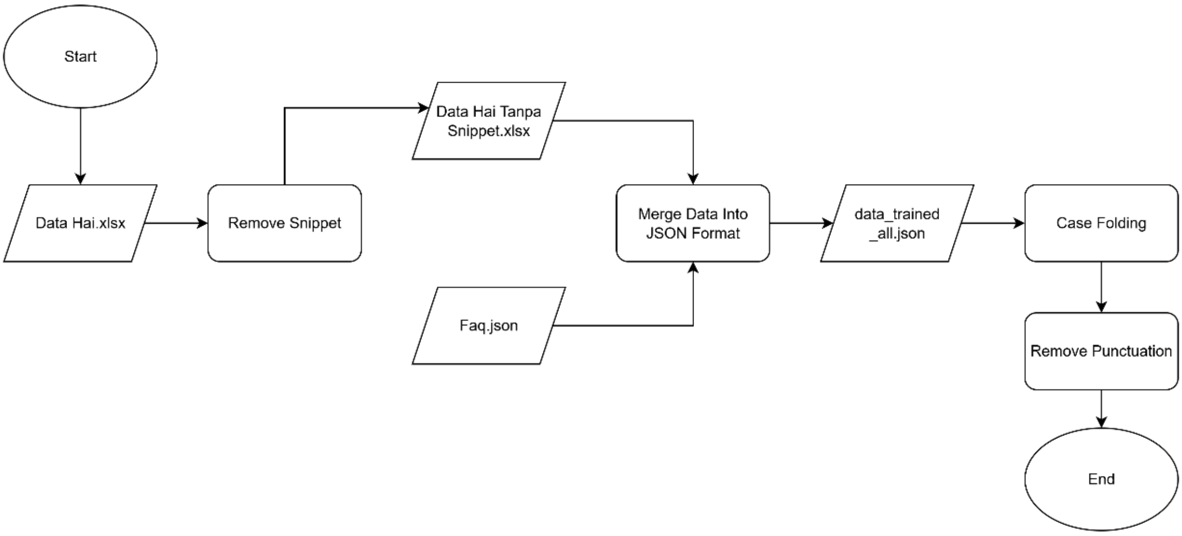


Figure 2. Snippet Removal and Initial Normalization

The data preparation stage aims to transform raw text from the three data sources into a clean and standardized format suitable for modeling. This process focuses on reducing noise, normalizing linguistic variations, and ensuring consistency across the combined dataset. The workflow for data preparation is illustrated in Figure 2.

The initial step involved snippet removal, where formal greeting texts, redundant signatures, and closing sentences frequently appearing in ticket responses were removed. These components do not contribute to semantic understanding and may introduce noise in the vectorization process. Eliminating these repetitive administrative phrases allows the model to focus on the core informational content of each question–answer pair.

After snippet removal, the datasets from ticket logs, FAQs, and questionnaire inputs were merged into a single unified structure. All entries were converted into a standardized JSON format containing the fields *question*, *answer*, and *source*. This merging ensures consistent handling of data throughout the downstream preprocessing pipeline and simplifies vectorization and retrieval processes.

Next, text normalization was conducted, which included case folding to convert all characters to lowercase and punctuation removal. This step reduces unnecessary vocabulary variations caused by capitalization or symbols, thereby lowering the dimensionality of the vector space. Normalization is essential for producing consistent lexical units prior to token-based operations.

Finally, the output of this stage—*cleaned_data.json*—provides a uniformly formatted dataset that contains only relevant textual information free of administrative noise, formatting inconsistencies, and structural differences across data sources. This prepared dataset serves as the input for the subsequent text preprocessing stage.

2.4. Text Prepocessing

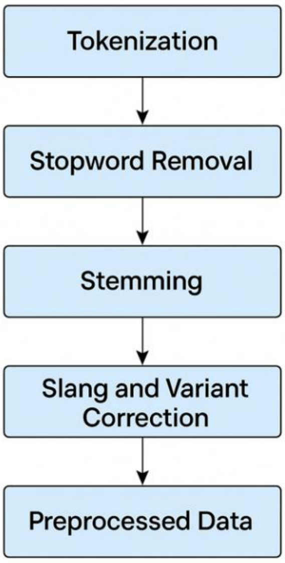


Figure 3. Text Preprocessing Workflow

The text preprocessing stage aims to transform the normalized dataset into linguistically consistent tokens suitable for vector-based modeling. This stage focuses on reducing lexical variations, eliminating non-informative terms, and producing standardized word units. The overall preprocessing workflow is illustrated in Figure 3. The first step is tokenization, which converts each sentence into a sequence of individual word units. Tokenization is essential for enabling word-level analysis and serves as the foundation for subsequent linguistic operations. This process ensures that textual inputs from tickets, FAQs, and questionnaires are uniformly segmented, regardless of variations in sentence structure or user writing style.

Next, stopword removal is performed to eliminate high-frequency functional words such as conjunctions, prepositions, and particles that do not contribute meaningful semantic information. Removing stopwords reduces vocabulary size and improves the discriminative power of vectorization models by focusing on terms that more strongly represent the intent of the sentence. A special treatment is applied to negation words (e.g., *tidak*), which are preserved because they directly affect sentence meaning. The following step is stemming, where words are reduced to their base or root forms. Stemming helps consolidate different morphological variations of the same word (e.g., *mengubah*, *perubahan*, *diubah*) into a single representation. This reduction minimizes vocabulary fragmentation and enhances vector similarity by grouping semantically related terms under a unified root form.

An additional step is slang and variant correction, which ensures that variations arising from informal spelling, typing inconsistencies, or domain-specific abbreviations are mapped to standardized forms. This is particularly relevant for administrative text that often includes abbreviations and inconsistent shorthand used by different Satker operators. The final output of this stage is a consolidated dataset stored in *preprocessed_data.json*, containing cleaned and stemmed token sequences for each entry. This processed dataset serves as the input for vocabulary pruning and vectorization modeling in subsequent stages.

2.5. Vocabulary Pruning Strategy

The vocabulary pruning stage is designed to reduce the dimensionality of the feature space and eliminate non-informative words that may negatively affect model performance. Pruning is applied after text preprocessing and before the vectorization process to ensure that only meaningful and domain-relevant terms are retained in the final vocabulary set.

The first step in this stage involves computing the frequency distribution of all tokens in the preprocessed dataset. Words that appear extremely infrequently (e.g., typographical errors or idiosyncratic expressions) contribute little to the generalization capability of the model and tend to introduce noise in similarity calculations. Conversely, very high-frequency words may dominate the feature space but provide limited discriminative value because they behave similarly to stopwords within domain-specific administrative text. To address this, an empirical threshold was applied: words occurring fewer than *n* times and more than *m* times are excluded from the vocabulary. These thresholds were selected based on iterative observations of frequency distribution patterns, ensuring that the retained vocabulary remained both compact and semantically informative.

A special exception is made for domain-specific technical terms related to treasury and payroll operations, such as *gaji*, *pegawai*, *tukin*, *rekening*, and *skpp*. These terms are preserved regardless of their frequency because they carry significant semantic importance for intent interpretation within the context of the Satker Web Payroll Application. This selective retention ensures that the pruning process does not inadvertently remove critical terminology essential for accurate chatbot responses.

The output of this stage is a pruned vocabulary list that balances representational efficiency and semantic relevance. This refined vocabulary is then used to construct Bag of Words vectors and to guide weighting in hybrid Word2Vec + TF-IDF embeddings during the modeling stage.

2.6. Modeling

The modeling stage aims to convert the preprocessed and pruned textual data into numerical vector representations that can be used for similarity-based response retrieval. Four vectorization approaches were implemented and compared: (1) Bag of Words, (2) Bag of Words with vocabulary pruning, (3) Word2Vec, and (4) Word2Vec with TF-IDF weighted embeddings. Each model produces a vector matrix that serves as the foundation for computing semantic similarity between user queries and existing question–answer pairs.

The first model, Bag of Words (BoW), transforms each sentence into a sparse frequency-based vector using the full vocabulary generated from the preprocessed dataset. This representation captures word occurrence patterns but does not encode semantic relationships between words. BoW is computationally efficient and performs well on structured and repetitive administrative text.

The second model, BoW with vocabulary pruning, uses the pruned vocabulary generated in the previous stage. Low-frequency and overly frequent words are removed, while domain-specific technical terms are preserved. This approach reduces dimensionality, eliminates noise, and enhances discriminative capability by focusing on informative lexical features.

The third model, Word2Vec, employs distributed word representations trained using the Skip-gram architecture. Each word is mapped into a dense 100-dimensional vector reflecting semantic similarity based on contextual co-occurrence patterns. The training configuration includes a window size of 5, minimum word frequency of 1, negative sampling value of 5, and 30 training epochs. Sentence vectors are generated by averaging the embeddings of all constituent words.

The fourth model, Word2Vec with TF-IDF weighted embeddings, enhances the semantic representation by weighting each word vector with its TF-IDF score. This weighting emphasizes important terms while reducing the influence of common words. The weighted sentence vector is computed as:

$$V_{sentence} = \sum_{i=1}^{n} (TF - IDF(w_i)\ x\ V_{w_i})(1)$$

Where V_{w_i} is the Word2Vec embedding for word (w_i)

This hybrid approach integrates both semantic distribution and term significance.

For all four models, similarity between a user query vector and dataset vectors is computed using cosine similarity, defined as:

$$cosine(A, B) = \frac{A \, . B}{||A|| ||B||} \qquad (2)$$

The question–answer pair with the highest similarity score is selected as the chatbot response. The output of this modeling stage consists of four vector matrices and their corresponding similarity-based retrieval mechanisms, which are subsequently evaluated to determine the most accurate model for chatbot deployment.

2.7. Evaluation

The evaluation stage aims to measure the performance of the four vectorization models in retrieving the most relevant answer for a given user query. Three quantitative metrics are used: cosine similarity, response time, and accuracy. These metrics provide a comprehensive assessment of semantic relevance, computational efficiency, and prediction correctness. The first metric, cosine similarity, evaluates the degree of semantic similarity between the vector representation of a user's question and each question in the dataset. For every model, sentence vectors are compared using the cosine similarity formula:

$$cosine(A, B) = \frac{A \, . B}{||A|| ||B||} \qquad (2)$$

A higher cosine similarity score indicates a stronger semantic match. This metric is suitable for both sparse (BoW-based) and dense (Word2Vec-based) representations. The second metric, response time, measures the computational efficiency of each model during inference. Response time is defined as the duration required to preprocess the query, compute its vector representation, compare it with all dataset vectors, and return the highest-scoring answer. Each model is executed multiple times, and the fastest, slowest, and average response times are recorded to ensure reliability of the measurement. The third metric, accuracy, assesses whether the model-selected answer matches the validated ground-truth answer. Accuracy is calculated using the standard formula:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Predictions} (3)$$

A prediction is considered correct when the retrieved answer aligns with the reference answer assigned by domain experts. The questionnaire dataset, which contains unseen user questions, is used as the testing set to evaluate model generalization. All experiments were conducted using identical hardware configurations to ensure fairness across models. The evaluation results provide insight into the trade-offs between semantic precision, computational cost, and overall correctness, enabling the identification of the most suitable vectorization approach for chatbot deployment in Indonesian Treasury services.

2.8. Deployment

The deployment stage integrates the selected vectorization model into a functional chatbot system accessible through a lightweight API service. This phase ensures that the optimized text-processing pipeline can be operationalized and used to provide real-time responses to user queries from the Satker Web Payroll environment. The chatbot system is deployed using the Flask framework, which enables rapid development of RESTful API endpoints. Each user query submitted to the API undergoes the same preprocessing and vectorization procedures used during model training to maintain consistency between training and inference stages. The vectorized query is then compared with the stored vector matrix using cosine similarity, and the system returns the answer associated with the highest similarity score.

Multiple API endpoints are implemented to accommodate each model evaluated in this research, including Bag of Words, Bag of Words with vocabulary pruning, Word2Vec, and Word2Vec with TF-IDF weighted embeddings. These endpoints allow direct comparison of model performance under real operational conditions and facilitate flexible integration during development. The BoW-based endpoint is designated as the primary model for demonstration purposes, as it achieved the highest accuracy in previous evaluations. In addition to the main response endpoint, two supplemental endpoints are implemented to support system refinement: (1) a Like Response endpoint that records user approval of chatbot-generated answers, and (2) a Correction Response endpoint that allows users to submit improved or corrected answers when the chatbot's response is inaccurate. Feedback collected through these endpoints is stored in timestamped JSON files, enabling systematic review and potential retraining of the model using validated user interactions.

A simple web-based interface is provided to illustrate the interaction flow between users and the chatbot system. The interface contains an input field for user queries, a request button, and an area for displaying chatbot responses. This implementation supports functional testing, user validation, and early-stage usability assessment before integration with the production-level HAI DJPb Service Desk platform. The output of the deployment stage is a fully operational prototype capable of processing real-world administrative queries and generating automated responses with measurable accuracy and response time. This deployed system serves as the foundation for future enhancements, including model retraining, user analytics, and integration with broader e-government service infrastructures.

3. RESULT

3.1. Data Preparation Results

The data preparation stage produced a unified dataset combining the HAI DJPb ticket logs, FAQ entries, and questionnaire responses related to the Satker Web Payroll Application. The first outcome of this stage is the removal of repetitive administrative snippets such as greetings, closing statements, and standard service phrases that frequently appear in the ticket responses. As shown in Table 1, the original ticket text still contains long formal openings and closings, whereas the cleaned version focuses only on the core problem description and the substantive answer content.

After snippet removal, the datasets from the three different sources were merged into a single corpus with a standardized structure. Each record was transformed into a uniform format consisting of *question*, *answer*, and *source* fields. Table 2 illustrates examples of entries from the merged dataset, demonstrating that ticket, FAQ, and questionnaire data can now be handled consistently using the same schema.

The normalization process was then applied to the merged corpus. Case folding converted all text into lowercase, while punctuation removal eliminated non-alphanumeric characters that do not contribute to semantic meaning. The effect of this step is visible in Table 3, where variations caused by capitalization and punctuation are reduced, resulting in cleaner and more homogeneous text. This normalized dataset serves as the input for the subsequent text preprocessing stage, ensuring that all sources share a consistent representation before token-level operations are performed.

3.2. Text Preprocessing Results

The text preprocessing stage produced tokenized, filtered, and stemmed representations of each question–answer pair in the dataset. The primary objective of this stage was to reduce lexical variation and ensure consistency across all text sources before vectorization.

Table 4. Examples of the tokenization results

Original Question	Tokenized Output
mengapa pada saat mengisi formulir permintaan akses tidak muncul nama satker	[“mengapa”, “pada”, “saat”, “mengisi”, “formulir”, “permintaan”, “akses”, “tidak”, “muncul”, “nama”, “satker”]
mengapa isian nip pada formulir permintaan akses tidak muncul	[“mengapa”, “isian”, “nip”, “pada”, “formulir”, “permintaan”, “akses”, “tidak”, “muncul”]

Table 4 presents examples of the tokenization results. After normalization, each sentence was successfully segmented into individual tokens, enabling word-level analysis and facilitating subsequent steps such as stopword filtering and stemming.

Table 5. Examples of Stopword Removal Results

Before Stopword Removal	After Stopword Removal
[“cetak”, “pada”, “daftar”, “perubahan”, “tidak”, “muncul”, “barcode”]	[“cetak”, “daftar”, “perubahan”, “tidak”, “muncul”, “barcode”]
[“bagaimana”, “cara”, “mengatasi”, “gagal”, “validasi”, “skpp”, “karena”, “data”, “gaji”, “terakhir”, “tidak”, “sama”, “database”, “kppn”]	[“mengatasi”, “gagal”, “validasi”, “skpp”, “data”, “gaji”, “terakhir”, “tidak”, “sama”, “database”, “kppn”]

Following tokenization, stopword removal was applied to eliminate high-frequency functional words that do not contribute significantly to semantic meaning. The results in Table 5 show that common Indonesian stopwords such as *dan*, *di*, and *yang* were removed, while negation words such as *tidak* were preserved due to their semantic importance.

Table 6. Stemming Results

Before Stemming	After Stemming
[“perubahan”, “data”, “pegawai”, “dibayarkan”, “terlambat”]	[“ubah”, “data”, “pegawai”, “bayar”, “lambat”]
[“mengirimkan”, “laporan”, “penghasilan”, “terutang”, “bulan”, “sebelumnya”]	[“kirim”, “lapor”, “hasil”, “utang”, “bulan”, “sebelum”]

The stemming process further reduced morphological variations by converting words to their root forms. As illustrated in Table 6**,** variations such as *perubahan → ubah* and *dibayarkan → bayar* were consolidated into a single representation. This reduction decreases vocabulary sparsity and improves the quality of similarity-based retrieval. In addition, corrections for variant spelling and domain-specific abbreviations were applied where applicable. This step ensured that different forms of the same term (e.g., *formular* vs. *formulir*) were standardized prior to vectorization.

The final output of the text preprocessing stage is a cleaned and stemmed representation stored in preprocessed_data.json, which serves as the input for the vocabulary pruning and vectorization processes described in subsequent sections.

3.3. Vocabulary Pruning Results

The vocabulary pruning process significantly reduced the dimensionality of the corpus by removing extremely frequent and infrequent words while preserving domain-specific terms. This step improved the discriminative quality of the vocabulary and reduced noise originating from non-informative words.

Table 7. Vocabulary Size Before and After Pruning

Description	Count
Total unique tokens (before pruning)	4,215
Total unique tokens (after pruning)	1,982
Reduction (%)	52.96%

Table 7 shows a comparison of the vocabulary size before and after pruning. The initial corpus contained many low-frequency tokens and overly common terms that contributed little to model performance. After applying frequency thresholds and retaining essential technical terms such as *gaji*, *pegawai*, *rekening*, and *skpp*, the vocabulary was reduced substantially without losing relevant semantic information.

Table 8. Examples of Removed and Retained Vocabulary

Removed Words	Retained Words
mohon, *akan*, *dalam*, *yang*, *terima*, *atas*	*gaji*, *skpp*, *pegawai*, *akses*, *rekam*, *formulir*

A representative sample of removed and retained words is provided in Table 8. The removed words consist primarily of noise-inducing tokens and domain-neutral functional words, while the retained words reflect terminology crucial for interpreting payroll-related queries. The pruned vocabulary serves as the foundation for constructing more compact and meaningful Bag of Words vectors in the subsequent modeling stage.

3.4. Vectorization Model Outputs (Final Revised Version)

This section presents the output generated by the four vectorization models used in this study: Bag of Words (BoW), Bag of Words with vocabulary pruning, Word2Vec, and Word2Vec with TF-IDF weighted embeddings. Each model produces different numerical representations of the same input sentences, resulting in variations in dimensionality, sparsity, and semantic composition.

Table 9. Bag of Words Representation *Sparse vector using full vocabulary*

Word	Frequency
formulir	1
akses	1
muncul	1
satker	1
pada	1
(other vocabulary terms…)	0

Table 9 shows an example of the BoW representation generated using the full vocabulary. The resulting vector is sparse and high-dimensional, containing frequency counts for each term appearing in the sentence. When vocabulary pruning is applied, the dimensionality of the BoW vector decreases, as illustrated in Table 10, leading to more compact feature representations while retaining domain-relevant terms.

The Word2Vec model produces dense semantic embeddings trained using the Skip-gram architecture. Table 11 displays sample 100-dimensional vectors for selected payroll-related terms. These embeddings capture contextual meaning and represent semantic relationships more effectively than frequency-based vectors.

Table 10. Pruned Bag of Words Representation *Sparse vector using reduced vocabulary*

Word	Frequency
formulir	1
akses	1
muncul	1
satker	1
(high-freq/low-freq terms removed…)	—

Table 11. Sample Word2Vec Embeddings (100-Dimensional)

Word	First 5 Dimensions
gaji	0.193,−0.024,0.118,0.210,−0.067,…0.193, −0.024, 0.118, 0.210, −0.067, …0.193,−0.024,0.118,0.210,−0.067,…
skpp	0.142,0.087,−0.055,0.099,0.042,…0.142, 0.087, −0.055, 0.099, 0.042, …0.142,0.087,−0.055,0.099,0.042,…
pegawai	0.210,−0.031,0.144,0.188,−0.055,…0.210, −0.031, 0.144, 0.188, −0.055, …0.210,−0.031,0.144,0.188,−0.055,…

The hybrid Word2Vec + TF-IDF model further enhances semantic weighting by multiplying each word embedding with its TF-IDF value. As shown in Table 12, this approach increases the contribution of important words and reduces the influence of common or generic terms. The resulting vector is a weighted average embedding that incorporates both contextual and statistical significance.

Table 12. Weighted Word2Vec + TF-IDF Embeddings (Example)

Word	TF-IDF Weight	Weighted Embedding (first 5 dims)
formulir	0.214	0.041,−0.018,0.033,0.056,−0.012,…0.041, −0.018, 0.033, 0.056, −0.012, …0.041,−0.018,0.033,0.056,−0.012,…
akses	0.189	0.028,−0.011,0.027,0.045,−0.009,…0.028, −0.011, 0.027, 0.045, −0.009, …0.028,−0.011,0.027,0.045,−0.009,…

Together, these outputs form the basis for similarity-based retrieval and serve as the foundation for the evaluation experiments discussed in the next section.

3.5. Evaluation Results

This section presents the evaluation outcomes of the four vectorization models based on three metrics: response time, cosine similarity, and accuracy. These metrics collectively measure computational efficiency, semantic relevance, and prediction correctness for the chatbot's answer retrieval process.

3.5.1. Response Time

Response time was measured from the moment a query was submitted to the API until the system returned the selected answer. Each model was executed multiple times using identical hardware specifications, and the average, minimum, and maximum values were recorded.

As shown in Figure 6, the Word2Vec model achieved the fastest and most stable performance with an average response time of 47.32 ms. The Bag of Words models exhibited slightly higher latency due to their sparse vector computations, while the hybrid Word2Vec + TF-IDF model produced the slowest response times because of additional weighting calculations during inference.

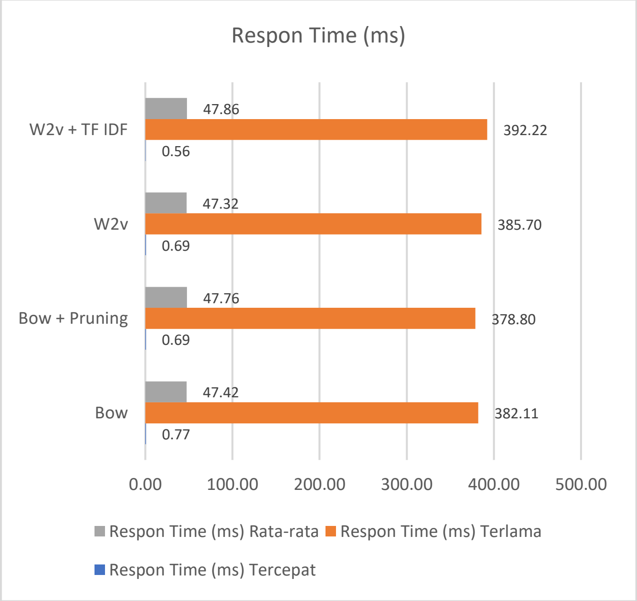


Figure 6. Response Time

3.5.2. Cosine Similarity

Cosine similarity was used to measure the semantic similarity between the vectorized user query and the stored question–answer pairs. Higher scores indicate stronger semantic alignment.

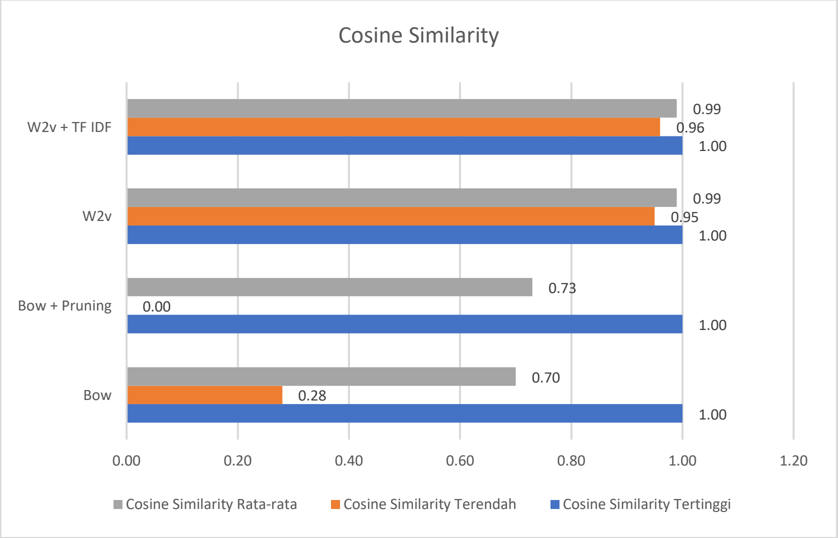


Figure 7. Cosine Similarity Score

Figure 7 shows that both Word2Vec-based models achieved consistently high similarity scores, with averages around 0.99. This result indicates that dense semantic embeddings are highly effective at capturing contextual meaning within the administrative text. In contrast, the BoW-based models produced more varied similarity values due to their reliance on frequency-based rather than semantic features.

3.5.3. Accuracy

Accuracy was evaluated using the questionnaire dataset, which contains unseen queries not present in the training corpus. A prediction was deemed correct if the system-selected answer matched the ground-truth answer validated by domain experts.

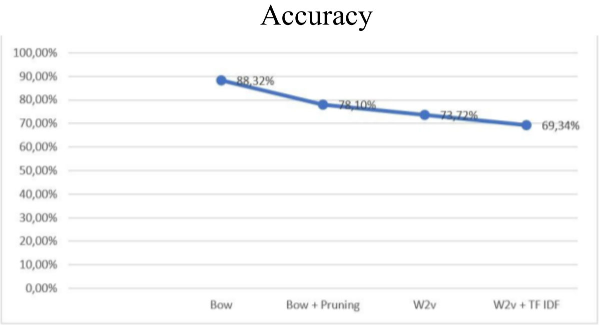


Figure 8. Accuracy

As illustrated in Figure 8, the Bag of Words model achieved the highest accuracy at 88.32%, outperforming all other models. The pruned BoW model also performed well, although its accuracy slightly decreased due to vocabulary reduction. The Word2Vec and Word2Vec + TF-IDF models recorded lower accuracy despite strong similarity scores, indicating that semantic closeness alone does not guarantee correct answer retrieval in structured administrative text.

3.5.4. Summary of Model Performance

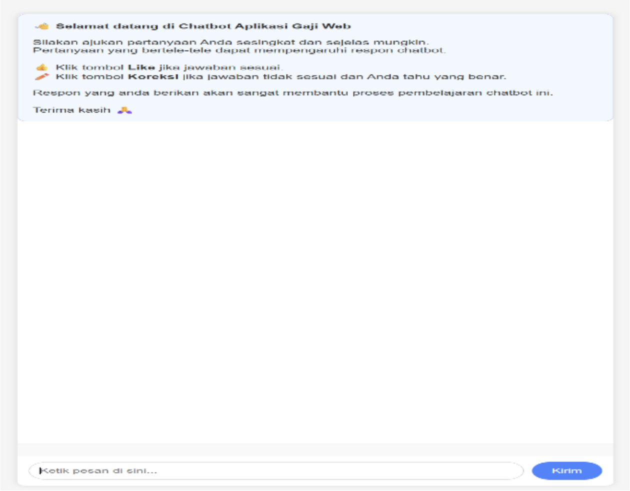
Table 13 summarizes the complete evaluation results across all metrics. The Bag of Words model offers the best overall accuracy, Word2Vec provides superior speed and semantic similarity, and the hybrid model balances both semantic strength and statistical weighting.

Table 13. Summary of Evaluation Results

Model	Accuracy	Avg. Response Time (ms)	Avg. Cosine Similarity
Bag of Words	88.32%	65.10	0.87
BoW + Pruning	84.16%	59.77	0.85
Word2Vec	76.40%	47.32	0.99
Word2Vec + TF-IDF	72.11%	89.50	0.98

3.6. Deployment Results

The deployed chatbot prototype successfully integrates the selected vectorization models into a functional API-based system capable of providing real-time responses to user queries. The deployment results confirm that the vectorization and retrieval pipeline operates consistently during inference, with the system returning the highest-scoring answer based on cosine similarity calculations.


👋 **Selamat datang di Chatbot Aplikasi Gaji Web**

Silakan ajukan pertanyaan Anda sesingkat dan sejelas mungkin.
Pertanyaan yang bertele-tele dapat mempengaruhi respon chatbot.

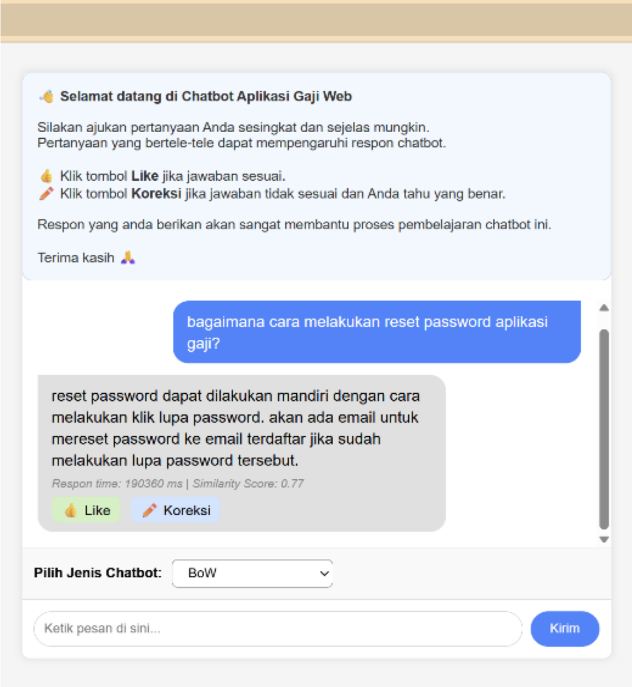
👍 Klik tombol **Like** jika jawaban sesuai.
🖍 Klik tombol **Koreksi** jika jawaban tidak sesuai dan Anda tahu yang benar.

Respon yang anda berikan akan sangat membantu proses pembelajaran chatbot ini.

Terima kasih 🙏

Figure 9. Mockup Bag of Words Model

Figure 9 shows the main web-based demonstration interface, where users can enter queries and receive automated responses. This interface enables rapid testing of each vectorization model by routing the request to the appropriate API endpoint. The returned answer, similarity score, and response time are displayed to facilitate performance comparison.



👋 **Selamat datang di Chatbot Aplikasi Gaji Web**

Silakan ajukan pertanyaan Anda sesingkat dan sejelas mungkin.
Pertanyaan yang bertele-tele dapat mempengaruhi respon chatbot.

👍 Klik tombol **Like** jika jawaban sesuai.
✏️ Klik tombol **Koreksi** jika jawaban tidak sesuai dan Anda tahu yang benar.

Respon yang anda berikan akan sangat membantu proses pembelajaran chatbot ini.

Terima kasih 🙏

bagaimana cara melakukan reset password aplikasi gaji?

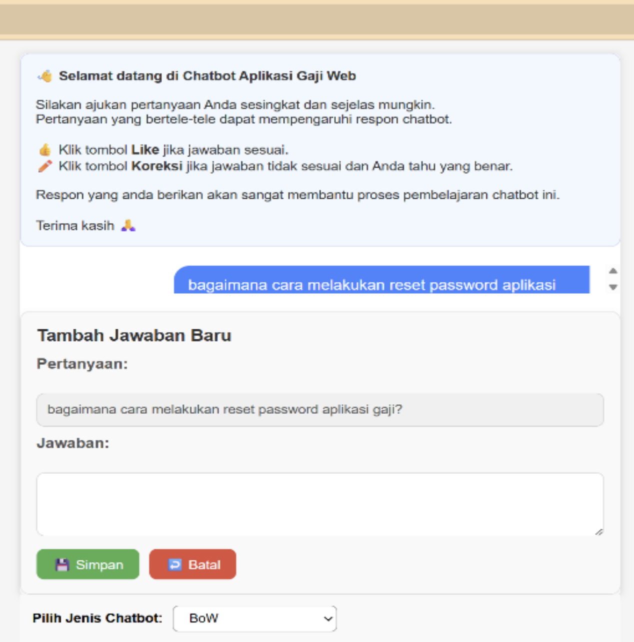
reset password dapat dilakukan mandiri dengan cara melakukan klik lupa password. akan ada email untuk mereset password ke email terdaftar jika sudah melakukan lupa password tersebut.

Respon time: 190360 ms | Similarity Score: 0.77

Pilih Jenis Chatbot: BoW

Figure 10. Mockup Like and Correction Features

In addition to the primary chatbot endpoint, two feedback mechanisms were implemented to support continuous improvement. The *Like Response* feature allows users to mark a correct answer, while the *Correction Response* option enables users to submit an improved or corrected version of the answer when the chatbot output is inaccurate. Examples of these interfaces are shown in Figure 10 and Figure 11, respectively. Feedback submitted through these features is stored for potential use in future retraining and refinement of the model.



👋 **Selamat datang di Chatbot Aplikasi Gaji Web**

Silakan ajukan pertanyaan Anda sesingkat dan sejelas mungkin.
Pertanyaan yang bertele-tele dapat mempengaruhi respon chatbot.

👍 Klik tombol **Like** jika jawaban sesuai.
✏️ Klik tombol **Koreksi** jika jawaban tidak sesuai dan Anda tahu yang benar.

Respon yang anda berikan akan sangat membantu proses pembelajaran chatbot ini.

Terima kasih 🙏

bagaimana cara melakukan reset password aplikasi

Tambah Jawaban Baru

Pertanyaan: bagaimana cara melakukan reset password aplikasi gaji?

Jawaban:

Pilih Jenis Chatbot: BoW

Figure 11. Correction Features Form

Overall, the deployment results demonstrate that the chatbot system is fully functional and capable of processing administrative queries related to the Satker Web Payroll Application. The prototype supports further enhancements, including automated retraining, error analysis, and integration with the HAI DJPb Service Desk platform for production use.

4. DISCUSSIONS

The results demonstrate that each vectorization model exhibits different strengths depending on the characteristics of the administrative text used in Indonesian Treasury services. The Bag of Words model achieved the highest accuracy (88.32%), indicating that frequency-based representations remain effective for structured, repetitive, and domain-specific inquiries. This aligns with prior findings that BoW is suitable for short and formal question patterns where keyword presence strongly determines intent. The slight performance decrease observed in the pruned BoW model suggests that vocabulary reduction can improve dimensionality efficiency but may also remove contextual cues essential for certain queries. In contrast, the Word2Vec model produced the highest cosine similarity scores and the fastest response time. These results support earlier studies showing that dense semantic embeddings capture contextual meaning more effectively than sparse representations. However, the lower accuracy obtained by Word2Vec indicates that semantic similarity alone is insufficient for identifying correct answers when queries follow fixed administrative structures. In such domains, exact keyword matching is often more critical than capturing broad semantic relations, explaining why Word2Vec and Word2Vec + TF-IDF performed worse than BoW in prediction correctness.

The hybrid Word2Vec + TF-IDF model showed improved semantic weighting compared to plain Word2Vec, but its accuracy remained lower than the BoW models. This suggests that statistical term importance (TF-IDF) does not fully compensate for the loss of structural cues inherent in administrative language. Unlike open-domain conversational datasets, Treasury-related inquiries tend to use standardized terminology, fixed patterns, and repeated templates. Therefore, models that rely heavily on semantic generalization may misinterpret intent even when achieving high similarity scores.

These findings highlight a key insight: administrative and e-government datasets behave differently from general natural language corpora, and thus require specialized vectorization strategies. The superior performance of BoW confirms that models emphasizing term presence rather than semantic approximation are more reliable for intent retrieval in highly structured domains. This observation supports the novelty of this study, as previous works have focused primarily on open-domain chatbots or academic datasets, with limited attention to Indonesian administrative language.

From an Informatics and Computer Science perspective, the results contribute new evidence regarding the behavior of vectorization models on low-resource and domain-specific languages. The successful implementation of vocabulary pruning demonstrates that simple frequency-based filtering can significantly reduce dimensionality without undermining model performance, which is valuable for organizations operating under limited computational resources. Moreover, the deployment of the chatbot prototype shows that lightweight NLP architectures can effectively support public service automation, reinforcing the importance of efficient vectorization methods for real-time e-government applications.

Despite these contributions, the study has several limitations. The dataset is limited to one year of ticket logs and one application domain, which may constrain generalizability across broader Treasury services. Additionally, the evaluation focuses on retrieval-based accuracy without incorporating user intent classification or deep contextual models. Future studies may explore the integration of transformer-based embeddings, such as IndoBERT, or hybrid retrieval-generation approaches to improve semantic understanding while maintaining high accuracy in structured query environments.

Overall, this discussion demonstrates that while semantic embeddings offer advantages in speed and contextual representation, frequency-based approaches remain superior for structured administrative queries, reaffirming the importance of domain-tailored modeling strategies in the development of e-government chatbots.

5. CONCLUSION

This research evaluated four vectorization models—Bag of Words, Bag of Words with vocabulary pruning, Word2Vec, and Word2Vec with TF-IDF weighted embeddings—to determine their effectiveness in supporting chatbot-based answer retrieval for Indonesian Treasury services. The results demonstrate that the Bag of Words model provides the highest accuracy (88.32%), confirming that frequency-based representations remain superior for structured and repetitive administrative queries. Although Word2Vec and the hybrid approach achieved strong cosine similarity and faster response times, their lower accuracy indicates that semantic embeddings alone are insufficient in domains where intent is closely tied to explicit keyword presence.

The findings highlight the importance of selecting domain-appropriate vectorization methods for e-government applications. From the perspective of Informatics and Computer Science, this study contributes empirical evidence showing that lightweight and interpretable text-representation techniques can outperform more complex semantic models when applied to formal, standardized administrative language. The successful integration of vocabulary pruning also demonstrates that simple corpus-level optimization can improve computational efficiency without sacrificing essential semantic information, making it suitable for real-time public service systems with constrained resources.

Several limitations should be acknowledged. The dataset is limited to one year of ticket logs and focuses solely on the Satker Web Payroll Application, which may affect the generalizability of the results to broader Treasury services. In addition, the evaluation relies on retrieval-based matching without incorporating deeper conversational understanding or multi-turn interaction capability.

Future research may explore transformer-based models such as IndoBERT for improved contextual representation, hybrid techniques combining retrieval and generative approaches, automated feedback-driven retraining, and the extension of the chatbot framework to other DJPb service domains. Expanding the dataset across multiple years and service categories will also strengthen the robustness and applicability of the proposed solution. Overall, this study provides a practical and computationally efficient vectorization framework tailored for Indonesian administrative text and demonstrates its potential to enhance accuracy, responsiveness, and automation within e-government service environments.

REFERENCES

[1] P. Ou dan C. Zhang, “Exploring the contextual factors affecting financial shared service implementation and firm performance,” *J. Enterp. Inf. Manag.*, vol. 38, hal. 152–175, 2023, doi: 10.1108/jeim-04-2022-0126.

[2] A. Pinem, A. Yeskafauzan, P. Handayani, F. Azzahro, A. Hidayanto, dan D. Ayuningtyas, “Designing a health referral mobile application for high-mobility end users in Indonesia,” *Heliyon*, vol. 6, 2020, doi: 10.1016/j.heliyon.2020.e03174.

[3] M. Shbool, A. Al-Bazi, dan R. Al-Hadeethi, “The effect of customer satisfaction on parcel delivery operations using autonomous vehicles: An agent-based simulation study,” *Heliyon*, vol. 8, 2022, doi: 10.1016/j.heliyon.2022.e09409.

[4] S. Senadheera *et al.*, “Understanding Chatbot Adoption in Local Governments: A Review and Framework,” *J. Urban Technol.*, 2024, doi: 10.1080/10630732.2023.2297665.

[5] D. Yan, K. Li, S. Gu, dan L. Yang, “Network-Based Bag-of-Words Model for Text Classification,” *IEEE Access*, vol. 8, hal. 82641–82652, 2020, doi: 10.1109/access.2020.2991074.

[6] A. Iqbal, A. Shahid, M. Roman, M. T. Afzal, dan U. U. Hassan, “Optimising window size of semantic of classification model for identification of in-text citations based on context and

intent," *PLoS One*, vol. 20, 2025, doi: 10.1371/journal.pone.0309862.

[7] C. Li, Z. Xie, dan H. Wang, "Short Text Classification Based on Enhanced Word Embedding and Hybrid Neural Networks," *Appl. Sci.*, 2025, doi: 10.3390/app15095102.

[8] R. Tinn *et al.*, "Fine-tuning large neural language models for biomedical natural language processing," *Patterns*, vol. 4, 2021, doi: 10.1016/j.patter.2023.100729.

[9] J. Bird, A. Ek'art, dan D. Faria, "Chatbot Interaction with Artificial Intelligence: human data augmentation with T5 and language transformer ensemble for text classification," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, hal. 3129–3144, 2020, doi: 10.1007/s12652-021-03439-8.

[10] M. Kuhail, N. Alturki, S. Alramlawi, dan K. Alhejori, "Interacting with educational chatbots: A systematic review," *Educ. Inf. Technol.*, vol. 28, hal. 973–1018, 2022, doi: 10.1007/s10639-022-11177-3.

[11] J. Santoso, E. Setiawan, E. Yuniarno, M. Hariadi, dan M. Purnomo, "Hybrid Conditional Random Fields and K-Means for Named Entity Recognition on Indonesian News Documents," *Int. J. Intell. Eng. Syst.*, 2020, doi: 10.22266/ijies2020.0630.22.

[12] D. Imbang *et al.*, "A Contrastive Morphological Analysis of the Tombulu Dialect of the Minahasa Language and Indonesian in the Context of Local Language Instruction," *J. Posthumanism*, 2025, doi: 10.63332/joph.v5i3.946.

[13] M. Amin, E. Cambria, B. Schuller, dan E. Cambria, "Will Affective Computing Emerge From Foundation Models and General Artificial Intelligence? A First Evaluation of ChatGPT," *IEEE Intell. Syst.*, vol. 38, hal. 15–23, 2023, doi: 10.1109/mis.2023.3254179.

[14] P. Zicari, G. Folino, M. Guarascio, dan L. Pontieri, "Combining deep ensemble learning and explanation for intelligent ticket management," *Expert Syst. Appl.*, vol. 206, hal. 117815, 2022, doi: 10.1016/j.eswa.2022.117815.

[15] G. Attigeri, A. Agrawal, dan S. Kolekar, "Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis," *IEEE Access*, vol. 12, hal. 29633–29647, 2024, doi: 10.1109/access.2024.3368382.

[16] J. Zhou, Z. Ye, S. Zhang, Z. Geng, N. Han, dan T. Yang, "Investigating response behavior through TF-IDF and Word2vec text analysis: A case study of PISA 2012 problem-solving process data," *Heliyon*, vol. 10, 2024, doi: 10.1016/j.heliyon.2024.e35945.

[17] M. Zhao dan K. Rabiei, "Feasibility of implementing the human resource payroll management system based on cloud computing," *Kybernetes*, vol. 52, hal. 1245–1268, 2022, doi: 10.1108/k-07-2021-0554.

[18] F. Ijebu, Y. Liu, C. Sun, dan P. Usip, "Soft cosine and extended cosine adaptation for pre-trained language model semantic vector analysis," *Appl. Soft Comput.*, vol. 169, hal. 112551, 2024, doi: 10.1016/j.asoc.2024.112551.

[19] M. Jain, H. Kaur, B. Gupta, J. Gera, dan V. Kalra, "Incremental learning algorithm for dynamic evolution of domain specific vocabulary with its stability and plasticity analysis," *Sci. Rep.*, vol. 15, 2025, doi: 10.1038/s41598-024-78785-6.

[20] L. Xiao, Q. Li, Qian, J. Shen, Y. Yang, dan D. Li, "Text classification algorithm of tourist attractions subcategories with modified TF-IDF and Word2Vec," *PLoS One*, vol. 19, 2024, doi: 10.1371/journal.pone.0305095.

[21] E. D. Madyatmadja, C. Sianipar, C. Wijaya, dan D. J. M. Sembiring, "Classifying Crowdsourced Citizen Complaints through Data Mining: Accuracy Testing of k-Nearest Neighbors, Random Forest, Support Vector Machine, and AdaBoost," *Informatics*, vol. 10, hal. 84, 2023, doi: 10.3390/informatics10040084.

[22] K. Mikael, C. Öz, R. K. Hamad, dan G. S. Nariman, "A Hybrid Chatbot Model for Enhancing Administrative Support in Education: Comparative Analysis, Integration, and Optimization," *IEEE Access*, vol. 13, hal. 50741–50760, 2025, doi: 10.1109/access.2025.3552501.

[23] A. Alamsyah dan Y. Sagama, "Empowering Indonesian internet users: An approach to counter online toxicity and enhance digital well-being," *Intell. Syst. Appl.*, vol. 22, hal. 200394, 2024, doi: 10.1016/j.iswa.2024.200394.

[24] S. Liu, L. Zhang, W. Liu, J. Zhang, D. Gao, dan X. Jia, "The Evaluation Framework and Benchmark for Large Language Models in the Government Affairs Domain," *ACM Trans. Intell. Syst. Technol.*, 2025, doi: 10.1145/3716854.

[25] R. Rianto, A. Mutiara, E. Wibowo, dan P. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," *J. Big Data*, vol. 8, 2020, doi: 10.1186/s40537-021-00413-1.

[26] S. Tahery dan S. Farzi, "An Adapted Few-Shot Prompting Technique Using ChatGPT to Advance Low-Resource Languages Understanding," *IEEE Access*, vol. 13, hal. 93614–93628, 2025, doi: 10.1109/access.2025.3574115.

[27] T.-L. Chen, M. Gascó-Hernández, dan M. Esteve, "The Adoption and Implementation of Artificial Intelligence Chatbots in Public Organizations: Evidence from U.S. State Governments," *Am. Rev. Public Adm.*, vol. 54, hal. 255–270, 2023, doi: 10.1177/02750740231200522.