JoEICT (Journal of Education and ICT)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/joeict
ISSN : 2987-3215
Vol. 8, No. 2, December 2024, Pp. 50-55

AN SVM-BASED APPROACH FOR DETECTING DATA DEFINITION LANGUAGE OPERATIONS IN INDONESIAN NATURAL LANGUAGE

Yayak Kartika Sari*1), Fahrur Rozi2) ,Agung Prasetya3)

1. Universitas Bhinneka PGRI, Indonesia
2. Universitas Bhinneka PGRI, Indonesia
3. Universitas Bhinneka PGRI, Indonesia

Article Info

Keywords: data definition language ; sentence; support vector machine; query ; text-to-sql

Article history:
Received 16 Oktober 2024
Revised 14 November 2024
Accepted 24 November 2024
Available online 1 December 2024

DOI :
https://doi.org/10.29100/joeict.v8i2.1649

* Corresponding author.
Corresponding Author
E-mail address:
informatikaubhi@gmail.com

ABSTRACT

Text-to-SQL is an approach that enables users to interact with databases using natural language, eliminating the need to understand SQL syntax. However, most existing approaches translate input sentences directly into final SQL queries without explicitly identifying the type of SQL operation involved. This may obscure the distinction between structural and manipulative commands and increase the risk of executing unintended or destructive queries. This study proposes separating the identification of SQL operation types—specifically Data Definition Language (DDL) commands—as a standalone classification task using the Support Vector Machine (SVM) algorithm. Indonesian-language sentences are preprocessed through tokenization, stopword removal, and stemming, then transformed into feature vectors using TF-IDF with unigram and bigram representations. Experiments were conducted on a dataset of 800 Indonesian sentences covering four DDL operations: CREATE, ALTER, DROP, and TRUNCATE. The results show that the proposed SVM model achieved an average accuracy of 93.05%, outperforming baseline models such as Naive Bayes and Random Forest. These findings indicate that early identification of SQL operation types can enhance the accuracy, efficiency, and safety of Text-to-SQL systems. This work also highlights the importance of developing NLP approaches tailored for the Indonesian language in the context of database querying.

.

I. INTRODUCTION

THE task of translating natural language into structured database queries, known as text-to-SQL, enables users to retrieve or manipulate data without requiring direct knowledge of SQL syntax [1], [2], [3]. Users simply input natural language expressions, which the system then automatically translates into executable SQL queries. This paradigm lowers the barrier to database interaction; however, it introduces substantial challenges due to the nature of natural language, which is inherently unstructured, ambiguous, and highly variable in expression. These characteristics pose significant difficulties for systems attempting to accurately interpret user intent. In contrast, SQL queries are rigidly structured, governed by strict syntactic rules, and highly sensitive to even minor interpretive errors.

To address these challenges, various approaches have been proposed. Rule-based methods utilize linguistic patterns to transform natural language into queries, offering a controllable solution that is particularly suitable for constrained domains [4], [5], [6]. Template-based methods [7], [8], [9], [10] aim to match input sentences with predefined query structures, allowing for limited variation in input without requiring query generation from scratch. More recently, sequence-to-sequence architectures [11], [12], [13], [14] and large language models (LLMs) [15], [16], [17], [18] have been employed to enhance system generalization and flexibility, enabling the handling of complex and diverse sentence structures beyond the capabilities of rule- or template-based systems.

A considerable portion of existing research on text-to-SQL focuses on the direct translation from natural language to final SQL queries, often overlooking the explicit identification of the underlying SQL operation type. This omission may obscure the distinction between structural and manipulative operations in the database. In particular, the identification of Data Definition Language (DDL) operations—such as CREATE, ALTER, DROP, and TRUNCATE—is critical, as these operations directly affect the schema structure. Failure to detect such operations may result in unintended and potentially harmful schema modifications. Early identification of DDL operations allows systems to issue appropriate warnings to users, thereby enhancing database security by preventing inadvertent structural changes.

This study frames the identification of DDL operations as a classification task. The Support Vector Machine (SVM) algorithm [19], [20] is adopted due to its effectiveness in managing high-dimensional and heterogeneous textual data. SVM is well-regarded for its strong generalization capability, even in scenarios with limited training data, as it maximizes the decision margin between classes. Moreover, SVM is notably robust against overfitting, particularly when the feature space is significantly larger than the sample size [21], [22], [23].

This research specifically targets Indonesian-language sentences, which present unique linguistic characteristics distinct from English. Evaluation is conducted across multiple domains to assess both the performance and generalization capability of the proposed model.

II. Literatur Review

Various approaches have been developed to translate natural language sentences into SQL queries. The earliest solutions to the text-to-SQL task were rule-based approaches. [4], [5], [6] These methods utilized linguistic rules to map sentence structures into corresponding query forms. Rule-based systems are relatively easy to control due to their reliance on explicitly defined patterns. However, they tend to lack flexibility in handling the diverse and often unpredictable expressions found in natural language.

To address these limitations, template-based approaches [7], [8], [9], [10] were introduced. These approaches attempt to match entities, attributes, or question words within a sentence to predefined query templates. By doing so, systems can handle some variation in sentence structure without generating queries from scratch. While template-based methods are more flexible than rule-based ones, they are still constrained by the need to design new templates to accommodate different query patterns.

In an effort to improve both flexibility and generalization, encoder-decoder-based approaches [15], [16], [17], [18] were proposed. These methods treat text-to-SQL as a sequence-to-sequence learning problem, solved using neural encoder-decoder architectures. Such models are capable of learning semantic structures from training data, thereby eliminating the need for handcrafted templates. This line of research has further evolved through the adoption of large language models (LLMs) such as T5 and Codex, which are trained on large-scale corpora. LLMs offer a deeper understanding of sentence context and can generate complex SQL queries. Both encoder-decoder and LLM-based approaches were designed to overcome the limitations of template-based methods, particularly in handling complex and highly variable sentence structures.

Nevertheless, most of these approaches translate natural language directly into final SQL queries without explicitly identifying the type of SQL operation—particularly Data Definition Language (DDL) operations. In fact, identifying the operation type can help mitigate destructive queries that alter the database schema. Unfortunately, to date, there has been no dedicated approach that explicitly focuses on the identification of DDL operations, especially in the context of the Indonesian language.

III. Proposed Approach

A. *SVM Model for Identifying DDL Operations*

This study formulates the problem of identifying Data Definition Language (DDL) operations from natural language sentences as a text classification task. The approach is designed to map Indonesian-language sentences to one of the DDL operation classes: CREATE, ALTER, DROP, or TRUNCATE.

Formally, given a dataset:

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\},$$

where each x_i is a feature representation vector of the i-th sentence and $y_i \in Y = \{CREATE, ALTER, DROP, TRUNCATE\}$ is the corresponding DDL operation label, the objective is to learn a classification function:

$$f: R^d \rightarrow Y$$

which maps each input vector to the correct class label.

This classification problem is addressed using a Support Vector Machine (SVM). SVM aims to find the optimal hyperplane that maximizes the margin between classes. In the linear case, the classification function is defined as:

$$f(x) = sign(w^t x + b)$$

where w is the weight vector and b is the bias term. The optimization objective of SVM is:

$$min\ (1/2)\|w\|^2 + C \sum \xi_i$$

subject to the constraints:

$$y_i\ (w^t x_i + b) \geq 1 - \xi_i, dan\ \xi_i \geq 0$$

where $C > 0$ is a regularization parameter that balances margin maximization and classification error.

For non-linearly separable data, the kernel trick is applied. In this study, the Radial Basis Function (RBF) kernel is used, defined as:

$$K(x_i, x_j) = exp(-\gamma\|x_i - x_j\|^2)$$

This kernel projects the data into a higher-dimensional space to capture non-linear relationships between features.

Since the task involves four DDL classes, a One-vs-Rest (OvR) strategy is adopted. Under this scheme, a separate binary SVM classifier is trained for each class $k \in Y$, where each model distinguishes class k from the rest. The final prediction is determined by selecting the class with the highest decision score:

$$\hat{y} = argmax\ f_k(x), \forall k \in Y$$

SVM is chosen due to its effectiveness in handling high-dimensional feature spaces and its robustness against overfitting. It is particularly suitable for situations with limited training data, as it maximizes the margin between classes. Additionally, SVM supports non-linear classification through the use of kernel functions such as the RBF kernel, providing greater flexibility for modeling complex relationships in text data.

B. *Sentence Features for DDL Detection*

Input sentences are converted into feature vectors through a two-phase process: pre-processing and feature extraction. In the pre-processing phase, sentences are first tokenized using an Indonesian-language tokenizer to separate individual words. Each token is then stemmed using the Nazief–Adriani stemming algorithm to obtain the root form. Common stopwords such as “yang”, “dengan”, and “adalah” are removed, as they contribute little to classification performance.

The first feature representation used is the Bag-of-Words (BoW) model, which encodes the frequency of each word without considering word order. Each sentence is thus represented as a d-dimensional vector based on the vocabulary.

Next, Term Frequency–Inverse Document Frequency (TF-IDF) is employed to assign numerical weights to each word, reflecting its importance in the sentence relative to the entire corpus. The TF-IDF value for a term t in document d is calculated as:

$$TFIDF(t, d) = f(t, d)/\Sigma_k f(k, d) \times log(N/df(t))$$

where $f(t, d)$ denotes the frequency of term t in document d, $df(t)$ is the number of documents containing term t, and N is the total number of documents.

To capture local context and word sequences, n-gram features are also extracted. This study employs both unigrams and bigrams ($n = 2$) to account for short-term dependencies. If the resulting feature space becomes too sparse, dimensionality reduction techniques such as Principal Component Analysis (PCA) are applied to reduce redundancy while retaining informative features.

IV. Results and Discussion

A. *Dataset*

The dataset used in this study comprises 800 Indonesian-language sentences representing SQL commands under the category of Data Definition Language (DDL) operations. Each sentence is annotated with a corresponding operation label: CREATE, ALTER, DROP, or TRUNCATE. The data were collected manually from SQL documentation, technical forums, database textbooks, and paraphrased variants generated by annotators. Each class contains 200 instances, resulting in a balanced dataset. The dataset was randomly split into training and testing subsets with an 80:20 ratio. Pre-processing steps included tokenization, stopword removal, and stemming to ensure consistent feature representation.

B. *Evaluation Scenario*

The experiments were designed to evaluate the effectiveness of the Support Vector Machine (SVM)-based classification approach in identifying DDL operation types. Feature representation was performed using the TF-IDF scheme with unigram and bigram configurations. A One-vs-Rest (OvR) strategy was adopted to address the multi-class classification setting. The Radial Basis Function (RBF) kernel was used, with hyperparameters C and γ optimized via grid search. A 5-fold cross-validation scheme was employed to assess model performance stability and generalization. Evaluation metrics included accuracy, precision, recall, and F1-score for each class.

C. Experimental Results

Table 1 presents the average accuracy from the five cross-validation folds.

TABLE I
ACCURACY PER FOLD

Fold	ACCURACY (%)
1	93.5
2	92.75
3	93.25
4	92
5	93.75
Aveage	93.05

Table 2 reports the precision, recall, and F1-score for each class, measured on the test data.

TABLE II
CLASS-WISE EVALUATION

Class	PRECISION (%)	RECALL (%)	F1-SCORE (%)
CREATE	94	91.5	92.7
ALTER	92.5	93	92.7
DROP	91	92	91.5
TRUNCATE	94.5	95	94.7

D. Baseline Comparison

To further validate the proposed approach, we conducted comparative experiments with two widely used text classification algorithms: Multinomial Naive Bayes (NB) and Random Forest (RF). These models serve as benchmarks due to their prevalence and distinct learning characteristics—NB relies on term frequency distributions, while RF leverages ensemble decision trees to capture non-linear patterns.

All models were trained and evaluated on the same dataset using identical feature representations. The NB model was used with default parameters, while RF was configured with 100 decision trees. The maximum tree depth was determined using a simple grid search strategy.

TABLE III
AVERAGE ACCURACY COMPARISON

Model	ACCURACY (%)
Naive Bayes	88.4
Random Forest	90.75
SVM (Proposed)	93.05

Table 4 provides the comparative average values of precision, recall, and F1-score across the three evaluated models.

TABLE IV
AVERAGE PRECISION, RECALL, AND F1-SCORE

Model	PRECISION (%)	RECALL (%)	F1-SCORE (%)
Naive Bayes	87.9	88.2	88
Random Forest	90.2	90.5	90.3
SVM (Proposed)	93	92.9	92.9

The results show that the proposed SVM model outperforms both baseline models across all metrics. NB struggles to perform well when bigram features are included, likely due to its strong assumption of feature independence. While RF yields reasonably good performance, it does not match SVM, particularly in handling high-dimensional sparse feature vectors with complex interdependencies.

E. Discussion

The results confirm that the proposed SVM-based approach achieves superior performance compared to the NB and RF baselines. The average accuracy of the SVM model (93.05%) surpasses both RF (90.75%) and NB (88.4%). SVM also demonstrates better precision, recall, and F1-scores.

The advantage of SVM lies in its ability to construct optimal non-linear decision boundaries using the RBF kernel. In contrast, NB's assumption of feature independence makes it less suitable for complex linguistic structures. Although RF handles non-linearity through its ensemble structure, it underperforms on sparse feature vectors—such as those generated by TF-IDF—due to its reliance on threshold-based splits.

Error analysis reveals that the SVM model effectively recognizes phrases such as "buat tabel pelanggan baru", "hapus tabel sementara", or "ubah struktur kolom". This indicates that surface-level features extracted from the sentence text are sufficiently informative for optimal classification.

Nonetheless, SVM has some limitations. Its training process becomes slower as dataset size or feature dimensionality increases. Furthermore, SVM models are generally less interpretable than NB models, making it difficult to trace and explain the classification decisions in practical applications.

V. Conclusion

This study proposes a classification-based approach for identifying Data Definition Language (DDL) operations from Indonesian-language sentences using Support Vector Machine (SVM). The model explicitly detects DDL operation types (CREATE, ALTER, DROP, TRUNCATE) and achieves an average accuracy of 93.05% based on 5-fold cross-validation. Comparative analysis with two baseline models—Naive Bayes and Random Forest—demonstrates the superiority of the SVM model across all evaluation metrics.

Future work may consider leveraging deep learning architectures such as BiLSTM or transformer-based classifiers. Expanding the dataset in both size and linguistic variety is also recommended. Additionally, integrating the proposed SVM approach into real-world applications such as query assistants or ERP-based chatbot systems may provide practical value.

References

[1] T. Yu et al., 'Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task', in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3911–3921.

[2] A.-M. Popescu, O. Etzioni, and H. Kautz, 'Towards a theory of natural language interfaces to databases', in Proceedings of the 8th international conference on Intelligent user interfaces, 2003, pp. 149–157.

[3] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, 'Natural language interfaces to databases—an introduction', Natural Language Engineering, vol. 1, no. 1, pp. 29–81, 1995.

[4] A.-M. Popescu, O. Etzioni, and H. Kautz, 'Towards a theory of natural language interfaces to databases', in Proceedings of the 8th international conference on Intelligent user interfaces, 2003, pp. 149–157.

[5] E. Goldberg and M. Safran, 'Using natural-language processing to produce relational queries', in Proceedings of the 13th annual meeting on Association for Computational Linguistics, ACL, 1985, pp. 183–189.

[6] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, 'Natural language interfaces to databases—an introduction', Natural Language Engineering, vol. 1, no. 1, pp. 29–81, 1995.

[7] A. Setzer and R. Gaizauskas, 'A template-based approach to information extraction', in Proceedings of the IJCAI Workshop on Text Mining and Link Analysis, 2001, pp. 65–72.

[8] R. Rajkumar and M. White, 'Template-based natural language generation for semantic querying', in Proceedings of the 13th European Workshop on Natural Language Generation, 2011, pp. 65–75.

[9] H. Liu, Y. Zhang, and C. Zhai, 'A template-based approach for user query presentation in E-commerce', in Proceedings of the 15th international conference on World Wide Web, 2006, pp. 1053–1054.

[10] A. Andrenucci and E. Sneiders, 'A survey on natural language question answering systems', Tech. Rep. 2005: 06, Institute of Computer Science, University of Latvia, vol. 14, no. 1, pp. 1–25, 2005.

[11] V. Zhong, C. Xiong, and R. Socher, 'Seq2SQL: Generating structured queries from natural language using reinforcement learning', in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 959–970.

[12] P. Yin and G. Neubig, 'TranX: A transition-based neural abstract syntax parser for semantic parsing and code generation', in EMNLP, 2018, pp. 5–14.

[13] R. Jia and P. Liang, 'Data recombination for neural semantic parsing', in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 12–22.

[14] L. Dong and M. Lapata, 'Language to logical form with neural attention', in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 33–43.

[15] B. Wang, Y. Xie, P. Liu, M. Gardner, and N. A. Smith, 'Text-to-SQL in the wild: A naturally-distributed dataset for multilingual text-to-SQL evaluation', Transactions of the Association for Computational Linguistics, vol. 11, pp. 128–145, 2023.

[16] T. Scholak, H. Schütze, and D. Bahdanau, 'Picard: Parsing incrementally for constrained auto-regressive decoding from language models', in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 10322–10328.

[17] B. Li, Y. Li, L. Wang, Y. Zhang, and X. Li, 'Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql parsers', in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), 2023.

[18] M. Chen et al., 'Evaluating Large Language Models Trained on Code'. 2021.

[19] D. Zhang and W. S. Lee, 'Text classification using support vector machine with multi-word features', in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003, pp. 366–367.

[20] C. Cortes and V. Vapnik, 'Support-vector networks', Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[21] Y. Wahba, N. Madhavji, and J. Steinbacher, ‘A Comparison of SVM against Pre-trained Language Models (PLMs) for Text Classification Tasks’, arXiv preprint arXiv:2211.02563, 2022.

[22] T. Hai, J. Zhou, S. Abolfath Zadeh, and others, ‘Evaluation of Text Classification Using Support Vector Machine Compared with Naive Bayes, Random Forest Decision Tree and K-NN’, in Proceedings of ICACTCE’23, Lecture Notes in Networks and Systems, 2023, pp. 321–331.

[23] B. Clavié and M. Alphonsus, ‘The Unreasonable Effectiveness of the Baseline: Discussing SVMs in Legal Text Classification’, arXiv preprint arXiv:2109.07234, 2021.