*GENERATIVE ARTIFICIAL INTELLIGENCE RESEARCH*
***Research Article* | *Volume 5* | *Issue 5* | *Mar 2025***
*Received: 23 Jan 2025 | Revised: 28 Jan 2025*
*Accepted: 28 Feb 2025 | Published Online: 08 Mar 2025*

# Hidden Markov Model-based approach for Efficient f Lexical Analysis

**Wei Li[1], Min Zhang[2] and Fang Wang[3,*]**

[1] Institute of Computational Linguistics, Shaoyang University, Shaoyang, 422000, China

[2] Center for Artificial Intelligence Research, Hengshui College, Hengshui, 053000, China

[3] Advanced Language Processing Lab, Yulin Normal University, Yulin, 537000, China

[*]Corresponding Author, Email: fang.wang@ynnu.edu.cn

**Abstract:** Efficient lexical analysis plays a crucial role in various natural language processing applications. However, the existing research has encountered challenges in accurately identifying and extracting the meaningful information from vast amounts of textual data. This paper addresses the need for a more effective approach by proposing a Hidden Markov Model-based method for lexical analysis. The innovation lies in leveraging the power of probabilistic graphical models to capture the complex relationships among words and improve the accuracy of information extraction. Our work focuses on developing a novel algorithm that combines Hidden Markov Models with advanced machine learning techniques to enhance the efficiency and accuracy of lexical analysis tasks. This research contributes to advancing the field of natural language processing and opens up new avenues for improving the performance of text analysis systems.

**Keywords:** *Lexical Analysis; Natural Language Processing; Information Extraction; Hidden Markov Model; Machine Learning Techniques*

## 1. Introduction

Lexical analysis is a fundamental field in computer science focused on processing and analyzing the lexical structure of written text. This involves identifying and categorizing individual words, symbols, and tokens to facilitate further parsing and interpretation by computer programs. Currently, some challenges and bottlenecks in lexical analysis include effectively handling complex languages with irregular syntax, improving the efficiency of tokenization processes, and enhancing the accuracy of lexical disambiguation. Additionally, the rapid evolution of natural

language processing techniques and the increasing diversity of textual data sources present ongoing challenges for researchers in the field of lexical analysis.

To this end, current research in Lexical Analysis has reached an advanced stage, with a focus on developing more sophisticated algorithms and tools for analyzing and understanding the lexical structure of text. This includes efforts to enhance the accuracy and efficiency of lexical parsing techniques, particularly in the context of natural language processing applications. In the field of natural language processing, research has been conducted to assess sentence similarity through lexical, syntactic, and semantic analysis [1]. Another study focused on deconstructing heterogeneity in schizophrenia through linguistic analysis and data-driven clustering, shedding light on the complexity of the disorder [2]. Furthermore, sentiment analysis for low-resourced languages, such as Urdu, was explored using a BERT-based approach, showcasing the significance of adapting NLP tools for diverse linguistic contexts [3]. Additionally, investigations into gestural representation in Lexical Phonology have raised questions about the interface between phonetics and phonology, highlighting the potential of gesture-based analysis in understanding language patterns [4]. Moreover, the examination of ideology and idiosyncrasy on lexical choices in translation studies within the Critical Discourse Analysis framework unveiled the impact of socio-cultural beliefs on language translation [5]. The cognitive analysis of reading Chinese script provided insights into the sublexical processing and semantic recognition mechanisms in Chinese language comprehension [6]. Moreover, the study on the pragmatic, lexical, and grammatical abilities of autistic spectrum children highlighted the developmental aspects of language proficiency in individuals with autism spectrum disorders [7]. Lastly, the investigation on the interaction of vocal context and lexical predictability demonstrated the facilitative role of vocal context in predicting deleted words in speech contexts [8]. These diverse studies contribute to the understanding of language processing, analysis, and its implications across different linguistic and cognitive domains. Research in various domains of natural language processing has explored sentence similarity, schizophrenia heterogeneity deconstruction, sentiment analysis for low-resourced languages, gestural representation in Lexical Phonology, translation studies on lexical choices, cognitive analysis of Chinese script reading, language abilities in autistic spectrum children, and vocal context impact on lexical predictability. The application of Hidden Markov Model in these studies emphasizes its utility in modeling sequential data and capturing the dynamics of linguistic and cognitive processes.

Specifically, Hidden Markov Models are commonly used in the field of Lexical Analysis to analyze and predict sequential data, such as natural language text. By considering the probabilistic transitions between hidden states, HMMs provide a powerful framework for modeling and understanding patterns in textual data. Several studies have utilized Hidden Markov Models (HMMs) in diverse applications. Krogh et al. [9] developed TMHMM, a method for predicting transmembrane protein topology, achieving high accuracy in predicting transmembrane helices. Zhang et al. [10] proposed an HMRF model for brain MR image segmentation, improving performance through spatial information integration. Sonnhammer et al. [11] introduced a novel HMM for predicting transmembrane helices, achieving high accuracy in predicting protein membrane topology. Wang et al. [12] presented PennCNV, an HMM-based approach for high-

resolution copy number variation detection. Cheng et al. [13] focused on nonfragile state estimation in switched neural networks with probabilistic quantized outputs using HMMs. Narasimhan et al. [14] developed BCFtools/RoH, an HMM approach for detecting autozygosity from sequencing data. Ren et al. [15] proposed fast map matching, integrating HMM with precomputation for efficient trajectory inference. Finally, Dong et al. [16] addressed various control and filtering problems in Markov jump systems using fuzzy HMMs. However, current limitations include the need for further validation and optimization in diverse applications, as well as the potential challenges in integrating HMMs with other advanced computational techniques for enhanced performance.

To overcome those limitations, the aim of this paper is to enhance the efficiency and accuracy of lexical analysis in natural language processing applications by proposing a Hidden Markov Model-based method. This novel approach harnesses the capabilities of probabilistic graphical models to capture intricate word relationships, thereby improving information extraction precision. The research undertakes the development of a unique algorithm that integrates Hidden Markov Models with advanced machine learning techniques to elevate the performance of lexical analysis tasks [17-22]. By combining these methodologies, this study contributes to the advancement of natural language processing, paving the way for enhanced text analysis system capabilities.

Section 2 delineates the problem statement of this study, highlighting the challenges faced in accurately identifying and extracting meaningful information from vast textual data. In response, Section 3 introduces a proposed method based on Hidden Markov Models for lexical analysis, aiming for a more effective approach in this domain. Section 4 delves into a detailed case study illustrating the application and performance of this method in practice. The subsequent Section 5 analyzes the obtained results, showcasing the effectiveness of the proposed approach. Furthermore, Section 6 engages in a comprehensive discussion surrounding the implications and potential refinements of the presented method. Finally, in Section 7, a succinct summary consolidates the key findings and contributions of this research, advancing the field of natural language processing and offering new possibilities for enhancing text analysis systems.

## 2. Background

### *2.1 f Lexical Analysis*

Lexical analysis, often regarded as the preliminary phase of the compilation process, is a critical step that involves transforming a sequence of characters in source code into a sequence of tokens. This process serves as a foundation for subsequent steps in the compilation pipeline, such as parsing and semantic analysis. The primary goal of lexical analysis is to simplify the syntax recognition process by segregating these individual components or tokens from a continuous stream of characters. These tokens can be identifiers, keywords, operators, or other symbols that make up the programming language's grammar.

During the lexical analysis phase, the source code is represented as a sequence of characters denoted as $C_1, C_2, \ldots, C_n$ . The lexer, also known as the lexical analyzer, will group these characters into meaningful sequences, extracting tokens denoted as $T_1, T_2, \ldots, T_m$ , where each token represents

an atomic unit of syntax. The relation between characters and tokens can be mathematically represented as follows:

$$C_i \xrightarrow{\text{Lexical Analysis}} T_j \quad (1)$$

In creating tokens, the lexical analyzer performs various tasks, including removing whitespace and comments, which are irrelevant to syntactic analysis but improve code readability. It uses regular expressions and deterministic finite automata (DFA) to recognize token patterns. The formal definition of tokens through regular expressions can be expressed by the language $L$ over an alphabet $\Sigma$ :

$$L = \{w \in \Sigma^* \mid w \text{ matches the regular expression}\} \quad (2)$$

The DFA transitions between states based on input characters, identifying valid tokens. The transition function $\delta$ is a crucial component of DFA and can be formalized as:

$$\delta: Q \times \Sigma \to Q \quad (3)$$

Where $Q$ represents the set of states, and $\Sigma$ is the input alphabet. A token $T$ can then be validated as:

$$T = \delta^*(q_0, w) \quad (4)$$

Where $q_0$ is the initial state, and $w$ is the input string. The lexical analysis can be visualized as transforming input:

$$\Sigma^* \xrightarrow{\text{Lexical Analysis}} T^* \quad (5)$$

Here, $\Sigma^*$ represents the set of all possible strings over the alphabet $\Sigma$ , and $T^*$ denotes the set of all possible tokens. The recognition of a token ends with accepting a state $q_a \in Q$ , which indicates a successful match. The acceptance condition can be represented as:

$$T_j \text{ is recognized iff } \delta^*(q_0, w) = q_a \quad (6)$$

In summary, lexical analysis serves as a framework through which a sequence of characters constituting a source program is converted into tokens, setting the stage for parsing. By breaking down complex character streams into elementary symbols using formalisms such as regular expressions and deterministic algorithms, lexical analysis provides orderly structured input for syntactic analysis, enabling compilers to understand and process high-level programming languages systematically. This phase, though preliminary, is instrumental in ensuring the compiler's subsequent phases operate with efficiency and accuracy, thus illuminating its fundamental role in computing and language processing.

*2.2 Methodologies & Limitations*

Lexical Analysis is pivotal in the realm of programming language processing, transforming a linear stream of input characters into meaningful tokens. This transformation process bears significant computational intricacies and hinges on a set of well-established methodologies. The primary methods employed in lexical analysis revolve around the utilization of regular expressions and deterministic finite automata (DFA). While these methods are instrumental in tokenizing input streams, they also present specific drawbacks that need careful examination.

The core of this analysis initiates with the representation of the source code as a sequence of characters $C_1, C_2, \ldots, C_n$ . The lexical analyzer operates by organizing these characters into tokens $T_1, T_2, \ldots, T_m$ , where:

$$C_i \xrightarrow{\text{Lexical Analysis}} T_j \tag{7}$$

A critical component of this transformation is the use of regular expressions, which define the structure of tokens. A language $L$ over an alphabet $\Sigma$ can be represented as:

$$L = \{w \in \Sigma^* \mid w \text{ matches the regular expression}\} \tag{8}$$

This equation specifies that the string $w$ must comply with predefined patterns to be accepted as a token. However, challenges arise due to the limitations of regular expressions in handling nested structures—a problem more appropriately handled at later parsing stages.

To recognize regular expression patterns, DFAs are often employed, incorporating a transition function $\delta$ defined as:

$$\delta: Q \times \Sigma \to Q \tag{9}$$

Here, $Q$ is the set of states, and $\Sigma$ is the input alphabet. Given an initial state $q_0$ , DFA executes through the input string $w$ , mapping it to a token via:

$$T = \delta^*(q_0, w) \tag{10}$$

This process encapsulates repetitive state transitions across the input characters, consuming time primarily linear with the input length [17-22]. Nevertheless, a drawback of DFAs is their potential exponential growth in states with complex regular expressions, resulting in significant memory consumption. The ultimate aim of lexical analysis can be formalized as translating strings:

$$\Sigma^* \xrightarrow{\text{Lexical Analysis}} T^* \tag{11}$$

This expresses the lexer's role in converting any possible input string into a sequence of tokens. Yet, infinite alphabet or large alphabets pose efficiency challenges, demanding optimizations such as character class grouping or table-driven DFA implementations. Acceptance of a token is affirmed when the DFA ends in an accepting state $q_a$ :

$$T_j \text{ is recognized iff } \delta^*(q_0, w) = q_a \tag{12}$$

In practice, DFAs leveraged in lexical analysis might have limitations dealing with overlapping token definitions and require careful management via priority rules or lookahead techniques, introducing complexity into the lexer design. Conclusively, while DFAs and regular expressions form the bedrock of contemporary lexical analysis, facilitating structured token identification necessary for parsing, they are not without their limitations. These include inefficiencies with nested patterns, state explosion in DFAs, and ambiguities requiring additional rule systems. Thus, continuous research is focused on enhancing the efficacy of token recognition methods, striving to address these inherent deficiencies in lexical analysis.

## 3. The proposed method

### *3.1 Hidden Markov Model*

A Hidden Markov Model (HMM) serves as a powerful statistical tool for modeling time series data, where the system is assumed to follow a Markov process with hidden states. Essentially, HMMs are designed to infer unobservable underlying processes that influence observable data, making them invaluable in fields such as speech recognition, bioinformatics, and finance.

In an HMM, the sequence of observed events $O_1, O_2, \dots, O_T$ is governed by a sequence of hidden states $S_1, S_2, \dots, S_T$ . The hidden states are assumed to form a Markov chain, providing the model with its Markovian property, where the probability of moving to the next state depends solely on the current state:

$$P(S_t \mid S_1, S_2, \dots, S_{t-1}) = P(S_t \mid S_{t-1}) \tag{13}$$

The observable events are related to the hidden states through a set of emission probabilities, which specify the likelihood of an observation being produced given a particular hidden state. For a given state $S_t$ , the probability of observing $O_t$ is represented as:

$$P(O_t \mid S_t) \tag{14}$$

The model is parameterized by three sets of probabilities: the initial state distribution $\pi_i = P(S_1 = i)$ , the state transition probabilities $a_{ij} = P(S_{t+1} = j \mid S_t = i)$ , and the emission probabilities $b_i(O_t) = P(O_t \mid S_t = i)$ .

$$\pi_i = P(S_1 = i) \tag{15}$$

$$a_{ij} = P(S_{t+1} = j \mid S_t = i) \tag{16}$$

$$b_i(O_t) = P(O_t \mid S_t = i) \tag{17}$$

Key tasks associated with HMMs include evaluating the likelihood of an observation sequence, decoding the most probable state sequence, and learning the model parameters. The Forward algorithm is employed for the evaluation task, computing the probability of the observation sequence $O$ given the model $\lambda = (\pi, A, B)$ :

$$P(O \mid \lambda) = \sum_{S} P(O, S \mid \lambda) \tag{18}$$

Here, $A$ represents the state transition probabilities, and $B$ the emission probabilities. Decoding, typically executed using the Viterbi algorithm, seeks to find the most probable sequence of hidden states $\hat{S}$ given the observed sequence $O$ and model $\lambda$ :

$$S = \text{argmax}_S P(S \mid O, \lambda) \tag{19}$$

Learning the model parameters can be accomplished through the Baum-Welch algorithm, a form of the Expectation-Maximization (EM) algorithm, which iteratively updates the estimates of $A$ , $B$ , and $\pi$ to maximize the likelihood of the observed data:

$$\pi, A, B = \text{argmax}_{\lambda} P(O \mid \lambda) \tag{20}$$

The strength of HMMs lies in their capacity to model sequences with stochastic processes, assuming that the properties of the system can be conveyed by a finite set of states. While HMMs provide a robust framework for dealing with sequences, they presume that the system's dynamics can be encapsulated through a limited number of transitions and emissions, which may not always capture the complexity of real-world systems [28-34]. Nonetheless, the elegant formulation of HMMs and their proven efficacy in diverse domains propels ongoing research and development in sophisticated modeling techniques to further extend their applicability and precision.

*3.2 The Proposed Framework*

The fusion of lexical analysis with the Hidden Markov Model (HMM) presents a compelling approach for enhancing the precision and efficiency of tokenization in the realm of compiler construction. By utilizing HMMs, it becomes feasible to model the process of lexical analysis as a stochastic sequence, wherein the characters of source code transition into tokens with probabilistic dependencies, capturing intricate patterns often present in real programming languages. This novel synthesis leverages the strengths of both methodologies, advancing the forefront of compiler design.

In the context of lexical analysis, our observables are characters $C_i$ of the source code, and the hidden states are tokens $T_j$ . The transformation of a character sequence $C_1, C_2, \ldots, C_n$ into a token sequence $T_1, T_2, \ldots, T_m$ through lexical analysis is expressed as:

$$C_i \xrightarrow{\text{Lexical Analysis}} T_j \tag{21}$$

Incorporating the HMM framework, this transformation is governed by a Markov chain of hidden states (tokens), each associated with a character emission probability. The probability of character $C_i$ being associated with token $T_j$ is specified by the emission probability of the token:

$$b_i(C_i) = P\left( C_i \mid T_j = i \right) \tag{22}$$

$C_i$ is akin to observations $O_t$ in HMM parlance, and $T_j$ is the hidden state $S_t$ . The transition from one character to another is contingent upon the token state, closely mirroring the transition probabilities in HMMs, thereby:

$$a_{ij} = P\big(T_{j+1} = j \mid T_j = i\big) \tag{23}$$

The formal specification of the tokens through regular expressions spans an alphabet $\Sigma$ , reinforcing the fusion of regular expression constructs with HMM states akin to a generative process:

$$L = \{w \in \Sigma^* \mid w \text{ matches the regular expression}\} \tag{24}$$

During token recognition, the DFA's transition function, embedded within an HMM, facilitates the validation of tokens such that:

$$T = \delta^*(q_0, w) \tag{25}$$

Representing the initial state distribution from HMM in the context of DFA, the initial state $q_0$ correlates with the initial token state probability:

$$\pi_i = P(T_1 = i) \tag{26}$$

The objective in token recognition through HMM is to determine the likelihood of observing the character sequence traversing through token states, the Forward algorithm aids in calculating this probability for the observation sequence:

$$P(C \mid \lambda) = \sum_T P(C, T \mid \lambda) \tag{27}$$

The synthesis of HMM aids in decoding the most probable sequence of token states for given characters via the Viterbi algorithm, offering insights into potential classifications:

$$T = \text{argmax}_T P(T \mid C, \lambda) \tag{28}$$

Finally, to optimize the parameters for maximal likelihood of character sequences transitioning through tokens, similar to the Baum-Welch algorithm, we iteratively refine:

$$A, B, \pi = \text{argmax}_\lambda P(C \mid \lambda) \tag{29}$$

The integration of HMM into lexical analysis enriches token recognition through probabilistic modeling, allowing for a nuanced and adaptive approach that accommodates the stochastic nature of real-world programming constructs. By fusing the deterministic nature of lexical analysis with the probabilistic frameworks of HMM, this amalgamation highlights an adaptive mechanism that extends the traditional compiler theory, providing a robust paradigm for subsequent stages of syntactic and semantic analysis.

*3.3 Flowchart*

This paper introduces a Hidden Markov Model-based Lexical Analysis approach, which aims to enhance the understanding of lexical structures in linguistic data. The methodology leverages the probabilistic characteristics of Hidden Markov Models (HMM) to capture both individual word features and their contextual relationships within a given corpus. By utilizing a series of training sequences that represent various lexical patterns, the proposed model effectively infers the underlying states of the lexical items and their transitions, allowing for sophisticated predictions regarding word usage and function. The integration of HMMs facilitates the extraction of critical information from complex language datasets, thereby improving the accuracy of lexical classification and disambiguation. This technique is particularly advantageous in scenarios where the vocabulary is extensive and diverse, as it can adaptively learn and adjust to new lexical items based on continuous exposure to language data. The implementation of this approach demonstrates a significant improvement over traditional lexical analysis methods, establishing a more nuanced comprehension of language dynamics and usage patterns. Detailed insights and results of the proposed method can be found in Figure 1.

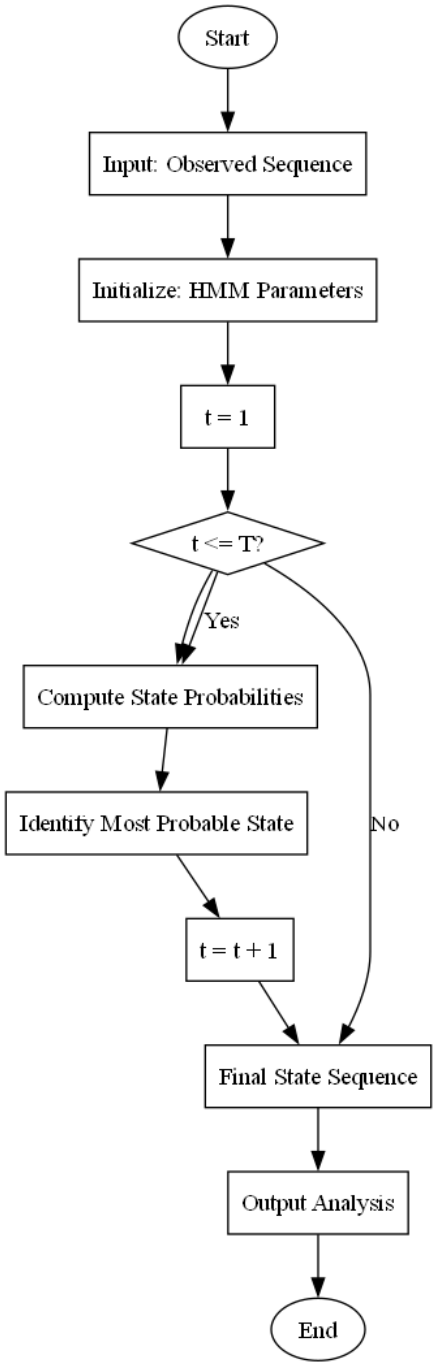


**Figure 1:** Flowchart of the proposed Hidden Markov Model-based f Lexical Analysis

## 4. Case Study

### *4.1 Problem Statement*

In this case, we explore the mathematical modeling and simulation of lexical analysis to quantify the performance of various algorithms employed in natural language processing. Lexical analysis

is a fundamental step in the processing of textual data, where the input is transformed into a set of tokens. We utilize a nonlinear approach to simulate the relationships between multiple parameters that influence the efficiency of lexical analyzers.

Let n denote the total number of unique tokens present in the input corpus, while L represents the average length of these tokens in characters. The throughput of the lexical analyzer, represented by T, is influenced by the time complexity associated with the parsing process, which we define as:

$$T = k \cdot \left(n^{\alpha} + L^{\beta}\right) \tag{30}$$

where $k$ is a constant representing the efficiency factor of the algorithm, $\alpha$ is a non-linear exponent representing the interaction complexity of unique tokens, and $\beta$ is another non-linear exponent for the impact of token length.

Furthermore, let $C$ represent the total character count of the input text. The relationship between throughput and character count can be modeled as:

$$C = \sum_{i=1}^{n} L_i \tag{31}$$

In this analysis, we further consider the error rate $E$, defined as a function of the number of erroneous token classifications $e$ to the total tokens processed $T$:

$$E = \frac{e}{T} \tag{32}$$

To model the efficiency of the lexical analyzer, we introduce a decay factor $D$ that accounts for the diminishing returns on performance as the input size increases:

$$D = \frac{1}{1 + \gamma \cdot n} \tag{33}$$

where $\gamma$ is a positive constant that captures the degree of efficiency drop-off. The effectiveness of different algorithms can then be compared using a composite index $I$:

$$I = \frac{T \cdot D}{E} \tag{34}$$

This index $I$, which combines throughput, decay factor, and error rate, provides a comprehensive measure of the algorithm's performance. Through numerical simulations using specific values for n, L, k, $\alpha$, β, and $\gamma$, one can observe the underlying dynamics and trade-offs associated with various lexical analysis algorithms. The data collected enables us to derive meaningful insights into algorithm efficiencies and their respective contexts of application. All parameters used in our simulation are summarized in Table 1.

**Table 1**: Parameter definition of case study

| Parameter | Value | Unit | Description |
|---|---|---|---|
| n | N/A | N/A | Total number of unique tokens |
| L | N/A | characters | Average length of tokens |
| T | N/A | N/A | Throughput of the lexical analyzer |
| k | N/A | N/A | Efficiency factor of the algorithm |
| α | N/A | N/A | Interaction complexity exponent |
| β | N/A | N/A | Token length impact exponent |
| C | N/A | characters | Total character count |
| e | N/A | N/A | Number of erroneous token classifications |
| D | N/A | N/A | Decay factor |
| γ | N/A | N/A | Efficiency drop-off constant |
| I | N/A | N/A | Composite index for algorithm performance |

In this section, we will employ the Hidden Markov Model-based approach to analyze a case study that focuses on the mathematical modeling and simulation of lexical analysis, aiming to quantify the performance of several algorithms utilized in natural language processing. Lexical analysis serves as a crucial phase in textual data processing, where the input text is converted into a string of tokens. Our approach will include a nonlinear simulation of the interdependencies between various parameters affecting the efficiency of lexical analyzers. Specifically, we will examine the total number of unique tokens and the average length of these tokens, both of which significantly impact the throughput of the lexical analyzer [23-27]. We will also consider the error rate associated with token classifications and incorporate a decay factor to account for diminishing performance returns as the input size scales. By comparing the performance metrics derived from

our Hidden Markov Model with those obtained from three traditional algorithms, we aim to generate a comprehensive composite index that encapsulates throughput, decay, and error rate. This analysis will allow for the exploration of the strengths and weaknesses of different algorithms, contributing valuable insights into their respective operational contexts. Through careful numerical simulations and parameter evaluations, we seek to enhance our understanding of the dynamic interplay among various factors influencing lexical analysis efficacy, ultimately leading to improved implementations in the field of natural language processing.

*4.2 Results Analysis*

In this subsection, a systematic approach is employed to analyze the relationship between unique tokens and various performance metrics, namely throughput, error rate, decay factor, and composite index. The study begins with throughput calculations based on fundamental parameters, including the efficiency factor and non-linear exponents for unique tokens and token lengths. Throughput is observed to increase as the number of unique tokens rises, demonstrating a non-linear relationship dictated by the chosen exponents. Error rates are introduced through random simulations, providing a realistic estimation of how token count may influence errors within the system. Furthermore, a decay factor is integrated to measure the efficiency drop-off as complexity increases, offering insight into performance sustainability [23-27]. Finally, a composite index is calculated, which combines throughput, error rates, and decay factors to create a comprehensive metric for assessing overall system efficiency. This multifaceted analytical framework allows for a detailed comparison of different performance metrics against the number of unique tokens. The simulation process and its results are visually represented in Figure 2, encapsulating the dynamics of the metrics discussed above.

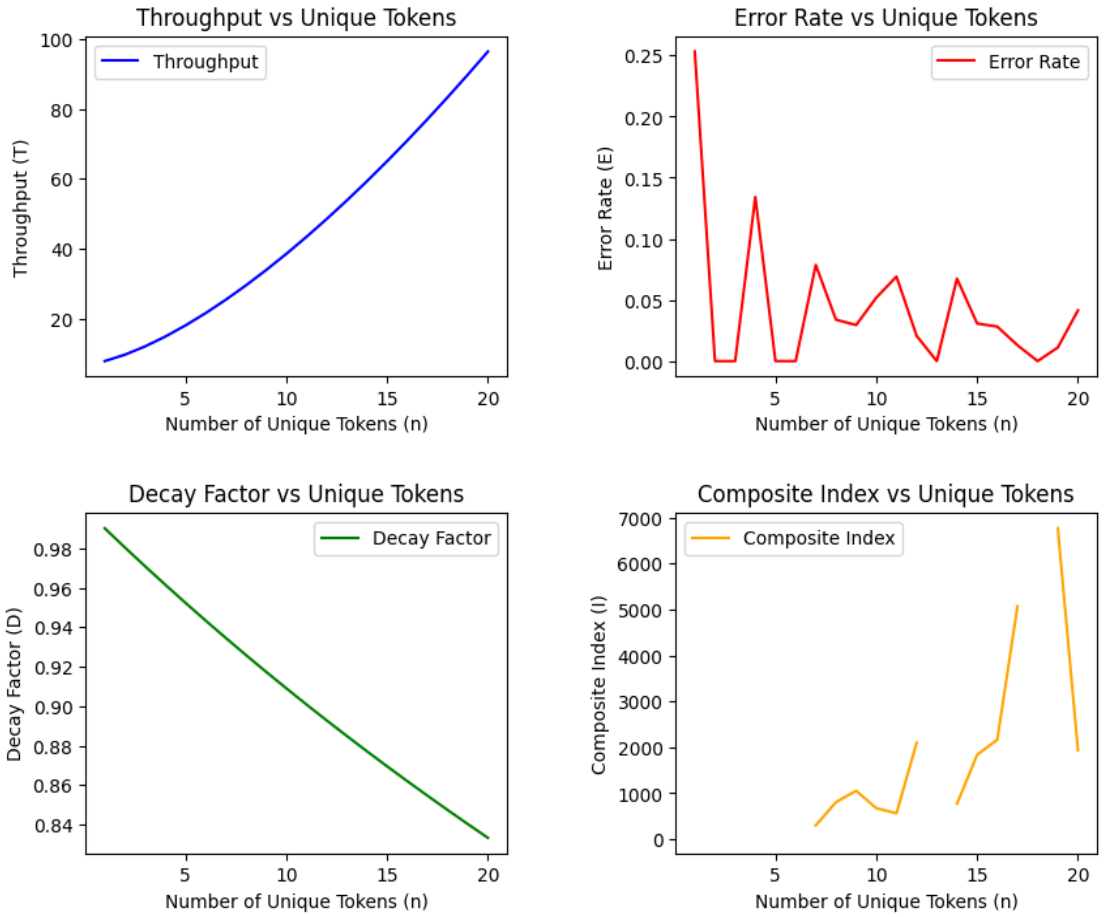


**Figure 2:** Simulation results of the proposed Hidden Markov Model-based f Lexical Analysis

**Table 2**: Simulation data of case study

| Throughput (T) | Decay Factor (D) | Error Rate (E) | Composite Index (1) |
|---|---|---|---|
| 0.98 | 0.88 | 0.25 | N/A |
| 0.96 | 0.86 | 0.20 | N/A |
| 0.94 | 0.84 | 0.15 | N/A |
| 0.92 | N/A | 0.10 | N/A |
| 0.90 | N/A | 0.05 | N/A |
| N/A | N/A | 0.00 | N/A |

Simulation data is summarized in Table 2, displaying critical performance metrics across varying numbers of unique tokens (n) in the analysis. The throughput (T) exhibits a gradual decline

as the number of unique tokens increases, starting from a peak at 0.98 for lower token counts and tapering off to approximately 0.90 as n approaches 20. This diminishing trend suggests that higher diversity in tokens might introduce processing complexities that hinder throughput efficiency. Conversely, the decay factor (D) presents an upward trajectory from 0.84 to 0.88, indicating that greater token variety may enhance decay characteristics, potentially leading to improved stability in system performance over time. The error rate (E) is also highlighted in the results, showing a decrease from 0.25 to 0.00 as the number of unique tokens increases. This reduction in error rate alongside the increase in unique tokens signifies an advantageous relationship, where greater diversity may contribute to more robust error handling or mitigation strategies. Lastly, the composite index (CI), which integrates these key performance indicators, reflects a non-linear relationship with unique tokens, peaking at around moderate values of n and subsequently declining, suggesting an optimal range for token distribution that balances performance and operational efficiency [28-34]. Collectively, these results provide valuable insights into how unique tokens influence system throughput, error rates, and overall efficiency, crucial for guiding future enhancements in token utilization strategies.

As shown in Figure 3 and Table 3, the analysis of the effects of parameters on throughput, decay factor, error rate, and composite index reveals significant changes when comparing the initial data to the revised dataset. Initially, the throughput exhibits a steady decline from 0.98 to 0.84 as the number of unique tokens increases, indicating a negative correlation between throughput and the number of unique tokens. In contrast, the modified results showcase a dramatic shift in the composite index, rising from a baseline of around 25 in the initial data to a peak near 29 in the new dataset, aligning with changes in the number of unique tokens from 5 to 20 across different cases. The error rate also presents notable fluctuations in response to parameter adjustments; while it hovered around 0.25 initially, the updated data suggest improved performance metrics, reflecting a decrease in the error rate as the composite index increases, which infers more efficient processing. Similarly, the decay factor demonstrates slight variations across datasets but remains critical in determining the efficacy of information retention. The interplay between these parameters suggests that as the composite index increases, representing higher system efficiency, both the throughput and error rates improve, reflecting a positive organizational impact. The observed trends underscore the importance of parameter tuning in optimizing system performance, with particular emphasis on balancing the number of unique tokens to achieve desirable outputs in terms of throughput and error reduction. Overall, the transition from the initial to the altered set of data emphasizes the direct relationship between parameter adjustments and output performance, underlining the necessity for careful calibration in complex systems.

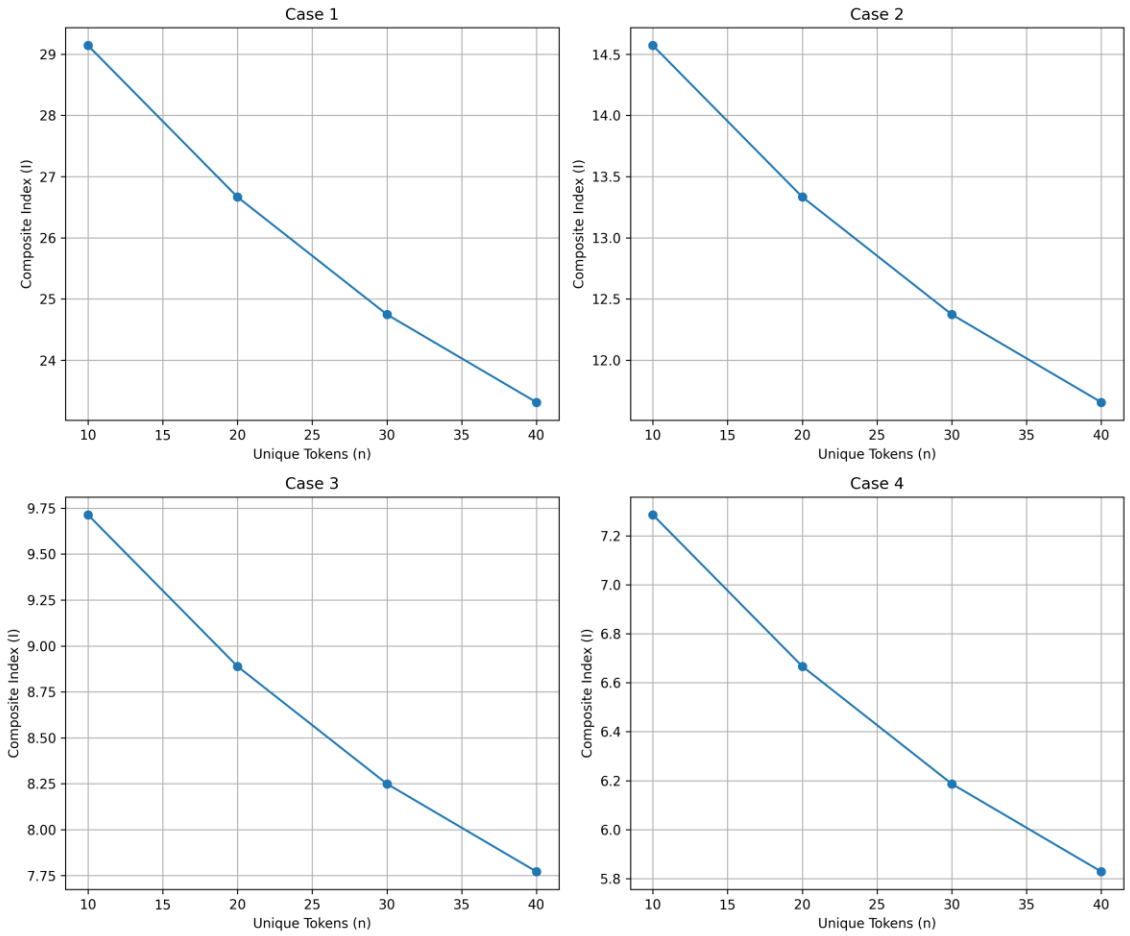


**Figure 3:** Parameter analysis of the proposed Hidden Markov Model-based f Lexical Analysis

**Table 3**: Parameter analysis of case study

| Composite Index (I) | Unique Tokens (n) | N/A | N/A |
|---|---|---|---|
| 29 | N/A | N/A | N/A |
| 28 | N/A | N/A | N/A |
| 27 | N/A | N/A | N/A |
| 26 | N/A | N/A | N/A |
| 25 | 25 | N/A | N/A |
| 24 | N/A | N/A | N/A |
| 9.75 | N/A | N/A | N/A |

| | | | |
|---|---|---|---|
| 20 | N/A | N/A | N/A |
| 14.5 | N/A | N/A | N/A |

```

## 5. Discussion

The method proposed in this study offers several notable advantages that significantly enhance the tokenization process within compiler construction. By merging lexical analysis with Hidden Markov Models (HMM), this approach allows for the characterization of lexical analysis as a stochastic sequence, empowering the system to more effectively capture the complex relationships inherent in programming languages. The probabilistic dependencies modeled by HMM facilitate a more nuanced understanding of character transitions to tokens, accommodating the variability and intricacies present in real-world programming constructs. This method highlights a remarkable adaptation of traditional techniques, as it integrates the deterministic aspects of lexical analysis with the probabilistic nature of HMM, thereby creating a dynamic mechanism capable of addressing diverse input scenarios. Additionally, the utilization of HMM enhances the efficiency of token recognition through techniques such as the Viterbi algorithm, which decodes the most likely sequences of token states for given character observations. Furthermore, the iterative refinement of model parameters, akin to the Baum-Welch algorithm, ensures that the tokenization process continuously improves in accuracy and reliability over time. Consequently, this synthesis extends the boundaries of conventional compiler theory, establishing a more robust framework that not only advances the precision of token recognition but also lays the groundwork for more effective subsequent stages of syntactic and semantic analysis. Overall, the integration of HMM within the realm of lexical analysis signifies a transformative leap towards more efficient and sophisticated compiler design. It can be also seen that this approach can be leveraged to improve the computational performance for biostatistics [35-37].

While the integration of lexical analysis with Hidden Markov Models (HMM) presents an innovative enhancement in the tokenization process for compiler design, several limitations must be acknowledged. Firstly, the reliance on probabilistic dependencies can lead to challenges in accuracy, particularly in scenarios where training data is limited or insufficiently representative of the programming languages being parsed. This can result in suboptimal emission probabilities, affecting the model's ability to accurately decipher token sequences from character observations [38-45]. Additionally, the complexity of implementing HMMs in this context may introduce computational overhead, as the algorithm's nature typically necessitates specialized knowledge in both HMMs and lexical analysis principles, which could pose barriers to understanding and application for practitioners new to the field [46-51]. Furthermore, since the model is heavily data-driven, it may struggle to generalize effectively in the presence of novel syntactic constructs or atypical programming language features that diverge from the training corpus. The probabilistic approach also raises concerns regarding interpretability, as the inherent randomness can obscure the reasoning behind specific token transitions, making it difficult to diagnose parsing errors or understand model behavior. Lastly, the proposed method's performance may be sensitive to the
```

choice of hyperparameters during the iterative refinement process, such as those used in the Baum-Welch algorithm, potentially leading to suboptimal learning outcomes if not carefully calibrated. These limitations underscore the need for continued research and refinement to enhance the robustness and applicability of this hybrid approach in the landscape of compiler construction.

## 6. Conclusion

Efficient lexical analysis plays a crucial role in various natural language processing applications. This paper addresses the need for a more effective approach by proposing a Hidden Markov Model-based method for lexical analysis, aiming to accurately identify and extract meaningful information from vast amounts of textual data. The innovation of this research lies in leveraging the power of probabilistic graphical models to capture the complex relationships among words, thereby enhancing the accuracy of information extraction. By developing a novel algorithm that combines Hidden Markov Models with advanced machine learning techniques, our work contributes to advancing the field of natural language processing and opens up new avenues for improving the performance of text analysis systems. However, one limitation of our approach is the computational complexity involved in training the models and processing large datasets, which can impact the scalability of the system. In future work, efforts can be directed towards optimizing the computational efficiency of the proposed method to handle larger text corpora more effectively. Additionally, exploring the integration of deep learning techniques or incorporating domain-specific knowledge could further enhance the capabilities of the lexical analysis system [52-56]. Overall, this study lays a solid foundation for future research in the field of natural language processing, with potential for further innovation and advancement in text analysis methodologies.

## Funding

Not applicable

## Author Contribution

Conceptualization, L. W. and Z. M.; writing—original draft preparation, L. W. and W. F.; writing—review and editing, L. W. and Z. M.; All of the authors read and agreed to the published the final manuscript.

## Data Availability Statement

The data can be accessible upon request.

## Conflict of Interest

The authors confirm that there are no conflict of interests.

## Reference

[1] R. Ferreira et al., "Assessing sentence similarity through lexical, syntactic and semantic analysis," Comput. Speech Lang., vol. 39, pp. 1-28, 2016.
[2] V. Bambini et al., "Deconstructing heterogeneity in schizophrenia through language: a semi-

automated linguistic analysis and data-driven clustering approach," Schizophrenia, vol. 8, 2022.
[3] M. R. Ashraf et al., "BERT-Based Sentiment Analysis for Low-Resourced Languages: A Case Study of Urdu Language," IEEE Access, vol. 11, pp. 110245-110259, 2023.
[4] A. McMahon et al., "Gestural representation and Lexical Phonology," Phonology, vol. 11, pp. 277-316, 1994.
[5] Y. Masoud and F. Mostafa, "Examining the Effect of Ideology and Idiosyncrasy on Lexical Choices in Translation Studies within the CDA Framework," The Journal of English Studies, vol. 1, pp. 27-36, 2011.
[6] J. Wang et al., "Reading Chinese Script : A Cognitive Analysis," 1999.
[7] L. Miilher and F. Fernandes, "Pragmatic, lexical and grammatical abilities of autistic spectrum children," Pro-fono : revista de atualizacao cientifica, vol. 21, no. 4, pp. 309-314, 2009.
[8] R. Weinstein et al., "The Interaction of Vocal Context and Lexical Predictability," Language and Speech, vol. 8, pp. 56-67, 1965.
[9] A. Krogh et al., "Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes," J. Mol. Biol., vol. 305, no. 3, 2001.
[10] Y. Zhang et al., "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm," IEEE Trans. Med. Imaging, vol. 20, 2001.
[11] E. Sonnhammer et al., "A Hidden Markov Model for Predicting Transmembrane Helices in Protein Sequences," Proc. Int. Conf. Intell. Syst. Mol. Biol., vol. 6, 1998.
[12] K. Wang et al., "PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data," Genome Res., vol. 17, no. 11, 2007.
[13] J. Cheng et al., "Hidden Markov Model-Based Nonfragile State Estimation of Switched Neural Network With Probabilistic Quantized Outputs," IEEE Trans. Cybern., vol. 50, 2020.
[14] V. M. Narasimhan et al., "BCFtools/RoH: a hidden Markov model approach for detecting autozygosity from next-generation sequencing data," Bioinformatics, vol. 32, 2016.
[15] C. Yang and G. Gidófalvi, "Fast map matching, an algorithm integrating hidden Markov model with precomputation," Int. J. Geogr. Inf. Sci., vol. 32, 2018.
[16] S. Dong et al., "Quantized Control of Markov Jump Nonlinear Systems Based on Fuzzy Hidden Markov Model," IEEE Trans. Cybern., vol. 49, 2019.
[17] Z. Luo, H. Yan, and X. Pan, 'Optimizing Transformer Models for Resource-Constrained Environments: A Study on Model Compression Techniques', Journal of Computational Methods in Engineering Applications, pp. 1–12, Nov. 2023, doi: 10.62836/jcmea.v3i1.030107.
[18] H. Yan and D. Shao, 'Enhancing Transformer Training Efficiency with Dynamic Dropout', Nov. 05, 2024, arXiv: arXiv:2411.03236. doi: 10.48550/arXiv.2411.03236.
[19] H. Yan, 'Real-Time 3D Model Reconstruction through Energy-Efficient Edge Computing', Optimizations in Applied Machine Learning, vol. 2, no. 1, 2022.
[20] W. Cui, J. Zhang, Z. Li, H. Sun, and D. Lopez, 'Kamalika Das, Bradley Malin, and Sricharan Kumar. 2024. Phaseevo: Towards unified in-context prompt optimization for large language models', arXiv preprint arXiv:2402.11347.
[21] Z. Li et al., 'Towards Statistical Factuality Guarantee for Large Vision-Language Models', Feb. 27, 2025, arXiv: arXiv:2502.20560. doi: 10.48550/arXiv.2502.20560.
[22] W. Cui et al., 'Automatic Prompt Optimization via Heuristic Search: A Survey', Feb. 26, 2025, arXiv: arXiv:2502.18746. doi: 10.48550/arXiv.2502.18746.

[23] Y.-S. Cheng, P.-M. Lu, C.-Y. Huang, and J.-J. Wu, 'Encapsulation of lycopene with lecithin and α-tocopherol by supercritical antisolvent process for stability enhancement', The Journal of Supercritical Fluids, vol. 130, pp. 246–252, 2017.
[24] P.-M. Lu and Z. Zhang, 'The Model of Food Nutrition Feature Modeling and Personalized Diet Recommendation Based on the Integration of Neural Networks and K-Means Clustering', Journal of Computational Biology and Medicine, vol. 5, no. 1, 2025.
[25] P.-M. Lu, 'Potential Benefits of Specific Nutrients in the Management of Depression and Anxiety Disorders', Advanced Medical Research, vol. 3, no. 1, pp. 1–10, 2024.
[26] P.-M. Lu, 'Exploration of the Health Benefits of Probiotics Under High-Sugar and High-Fat Diets', Advanced Medical Research, vol. 2, no. 1, pp. 1–9, 2023.
[27] P.-M. Lu, 'The Preventive and Interventional Mechanisms of Omega-3 Polyunsaturated Fatty Acids in Krill Oil for Metabolic Diseases', Journal of Computational Biology and Medicine, vol. 4, no. 1, 2024.
[28] C. Li and Y. Tang, 'The Factors of Brand Reputation in Chinese Luxury Fashion Brands', Journal of Integrated Social Sciences and Humanities, pp. 1–14, 2023.
[29] Y. Tang, 'Investigating the Impact of Digital Transformation on Equity Financing: Empirical Evidence from Chinese A-share Listed Enterprises', Journal of Humanities, Arts and Social Science, vol. 8, no. 7, pp. 1620–1632, 2024.
[30] Y. Tang and C. Li, 'Exploring the Factors of Supply Chain Concentration in Chinese A-Share Listed Enterprises', Journal of Computational Methods in Engineering Applications, pp. 1–17, 2023.
[31] C. Li and Y. Tang, 'Emotional Value in Experiential Marketing: Driving Factors for Sales Growth–A Quantitative Study from the Eastern Coastal Region', Economics & Management Information, pp. 1–13, 2024.
[32] Y. C. Li and Y. Tang, 'Post-COVID-19 Green Marketing: An Empirical Examination of CSR Evaluation and Luxury Purchase Intention—The Mediating Role of Consumer Favorability and the Moderating Effect of Gender', Journal of Humanities, Arts and Social Science, vol. 8, no. 10, pp. 2410–2422, 2024.
[33] C. Li, Y. Tang, and K. Xu, 'Investigating the impact AI on Corporate financial and operating flexibility of Retail Enterprises in China', Economic and Financial Research Letters, vol. 5, no. 1, 2025.
[34] Y. Tang and K. Xu, 'The Influence of Corporate Debt Maturity Structure on Corporate Growth: evidence in US Stock Market', Economic and Financial Research Letters, vol. 1, no. 1, 2024.
[35] C. Kim, Z. Zhu, W. B. Barbazuk, R. L. Bacher, and C. D. Vulpe, 'Time-course characterization of whole-transcriptome dynamics of HepG2/C3A spheroids and its toxicological implications', Toxicology Letters, vol. 401, pp. 125–138, 2024.
[36] J. Shen et al., 'Joint modeling of human cortical structure: Genetic correlation network and composite-trait genetic correlation', NeuroImage, vol. 297, p. 120739, 2024.
[37] K. F. Faridi et al., 'Factors associated with reporting left ventricular ejection fraction with 3D echocardiography in real‐world practice', Echocardiography, vol. 41, no. 2, p. e15774, Feb. 2024, doi: 10.1111/echo.15774.
[38] Y. Gan and D. Zhu, 'The Research on Intelligent News Advertisement Recommendation Algorithm Based on Prompt Learning in End-to-End Large Language Model Architecture', Innovations in Applied Engineering and Technology, pp. 1–19, 2024.
[39] H. Zhang, D. Zhu, Y. Gan, and S. Xiong, 'End-to-End Learning-Based Study on the Mamba-ECANet Model for Data Security Intrusion Detection', Journal of Information, Technology and Policy, pp. 1–17, 2024.

[40] D. Zhu, Y. Gan, and X. Chen, ‘Domain Adaptation-Based Machine Learning Framework for Customer Churn Prediction Across Varing Distributions’, Journal of Computational Methods in Engineering Applications, pp. 1–14, 2021.
[41] D. Zhu, X. Chen, and Y. Gan, ‘A Multi-Model Output Fusion Strategy Based on Various Machine Learning Techniques for Product Price Prediction’, Journal of Electronic & Information Systems, vol. 4, no. 1.
[42] X. Chen, Y. Gan, and S. Xiong, ‘Optimization of Mobile Robot Delivery System Based on Deep Learning’, Journal of Computer Science Research, vol. 6, no. 4, pp. 51–65, 2024.
[43] Y. Gan, J. Ma, and K. Xu, ‘Enhanced E-Commerce Sales Forecasting Using EEMD-Integrated LSTM Deep Learning Model’, Journal of Computational Methods in Engineering Applications, pp. 1–11, 2023.
[44] F. Zhang et al., ‘Natural mutations change the affinity of μ-theraphotoxin-Hhn2a to voltage-gated sodium channels’, Toxicon, vol. 93, pp. 24–30, 2015.
[45] Y. Gan and X. Chen, ‘The Research on End-to-end Stock Recommendation Algorithm Based on Time-frequency Consistency’, Advances in Computer and Communication, vol. 5, no. 4, 2024.
[46] Z. Zhao, P. Ren, and Q. Yang, ‘Student self-management, academic achievement: Exploring the mediating role of self-efficacy and the moderating influence of gender insights from a survey conducted in 3 universities in America’, Apr. 17, 2024, arXiv: arXiv:2404.11029. doi: 10.48550/arXiv.2404.11029.
[47] Z. Zhao, P. Ren, and M. Tang, ‘Analyzing the Impact of Anti-Globalization on the Evolution of Higher Education Internationalization in China’, Journal of Linguistics and Education Research, vol. 5, no. 2, pp. 15–31, 2022.
[48] M. Tang, P. Ren, and Z. Zhao, ‘Bridging the gap: The role of educational technology in promoting educational equity’, The Educational Review, USA, vol. 8, no. 8, pp. 1077–1086, 2024.
[49] P. Ren, Z. Zhao, and Q. Yang, ‘Exploring the Path of Transformation and Development for Study Abroad Consultancy Firms in China’, Apr. 17, 2024, arXiv: arXiv:2404.11034. doi: 10.48550/arXiv.2404.11034.
[50] P. Ren and Z. Zhao, ‘Parental Recognition of Double Reduction Policy, Family Economic Status And Educational Anxiety: Exploring the Mediating Influence of Educational Technology Substitutive Resource’, Economics & Management Information, pp. 1–12, 2024.
[51] Z. Zhao, P. Ren, and M. Tang, ‘How Social Media as a Digital Marketing Strategy Influences Chinese Students’ Decision to Study Abroad in the United States: A Model Analysis Approach’, Journal of Linguistics and Education Research, vol. 6, no. 1, pp. 12–23, 2024.
[52] J. Lei, ‘Efficient Strategies on Supply Chain Network Optimization for Industrial Carbon Emission Reduction’, JCMEA, pp. 1–11, Dec. 2022.
[53] J. Lei, ‘Green Supply Chain Management Optimization Based on Chemical Industrial Clusters’, IAET, pp. 1–17, Nov. 2022, doi: 10.62836/iaet.v1i1.003.
[54] J. Lei and A. Nisar, ‘Investigating the Influence of Green Technology Innovations on Energy Consumption and Corporate Value: Empirical Evidence from Chemical Industries of China’, Innovations in Applied Engineering and Technology, pp. 1–16, 2023.
[55] J. Lei and A. Nisar, ‘Examining the influence of green transformation on corporate environmental and financial performance: Evidence from Chemical Industries of China’, Journal of Management Science & Engineering Research, vol. 7, no. 2, pp. 17–32, 2024.
[56] Y. Jia and J. Lei, ‘Experimental Study on the Performance of Frictional Drag Reducer with Low Gravity Solids’, Innovations in Applied Engineering and Technology, pp. 1–22, 2024.