Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 3s (2025)

AutoETL: A Nonlinear Deep Learning Framework for ETL Automation

G. Sunil Santhosh Kumar[1,2*], M. Rudra Kumar[3]

[*1]Research Scholar, Department of CSE, JNTU Ananthapuramu, AP, India.

[2]Assistant Professor, Department of CSE, MLRITM-Hyderabad, India

[*1]Email: gsunilsanthosh105@gmail.com

[3] Professor, Dept., of Information Technology, MGIT, Hyderabad, India.

[2]Email: mrudrakumar@gmail.com

[*]Corresponding Author: G. Sunil Santhosh Kumar, Email: gsunilsanthosh105@gmail.com

Article History:

***Received**: 22-09-2024*

***Revised**: 02-11-2024*

***Accepted**: 18-11-2024*

Abstract:

This study presents a nonlinear framework for automating Extract, Transform, Load (ETL) processes. The framework uses natural language processing techniques, transformer-based models, and reinforcement learning to convert unstructured data into structured formats. It focuses on creating and refining transformation rules based on data patterns. The research addresses challenges in automating ETL processes, particularly the need to handle complex data relationships without manual input. The TPC-DI dataset is used to test the framework, which transforms financial newswire data into a structured warehouse format. The process follows ACID and OpenClass standards. The framework includes data preparation through tokenization and normalization. A transformer-based model processes sequences to identify patterns. Reinforcement learning refines transformation rules using feedback. The methods ensure structured data alignment measured through metrics like Intersection over Union (IoU), mean average precision (mAP), and mean squared error (MSE). The results show consistent performance across various data thresholds, highlighting its ability to handle diverse data patterns. This research outlines a method to automate data handling while reducing manual involvement, with potential applications across domains.

Keywords: nonlinear ETL framework, data transformation, tokenization, transformer model, reinforcement learning, IoU, structured data alignment, TPC-DI dataset.

1 Introduction

The field of data integration, particularly the automation of Extract, Transform, Load (ETL) processes, has seen considerable advancements in recent years. With the burgeoning volume and complexity of data across various domains, the need to efficiently convert unstructured data into structured formats has become imperative for enhancing data usability and accessibility. The introduction of deep learning technologies has further catalyzed this transformation, offering novel approaches to extracting relevant information and generating transformation rules that are both effective and scalable.

This article reviews significant contributions within the realm of data integration, with a focus on methodologies developed to handle and transform unstructured data into structured data, and the application of deep learning techniques for automating these processes. A considerable portion of

recent research has concentrated on developing robust methods to parse unstructured data records to ascertain their characterization, subsequently employing techniques such as key-value pairs and hashing to produce modified data records with indexing keys, thereby improving their searchability and query efficiency [1-6].

Moreover, deep learning approaches have been pivotal in advancing the capability to automate information extraction from unstructured data. Studies such as those by Gilligan et al. [7], [8], [9] have demonstrated how BERT models, fine-tuned for specific tasks, can significantly enhance the extraction of structured information from expansive datasets such as scientific literature. Similarly, innovative applications in detecting fraud within e-commerce by Bekach et al. [10] have utilized deep learning to extract conditional rules efficiently, showcasing the versatility of these models across different operational contexts. The application of these technologies extends beyond mere data transformation to include significant impacts in domains such as business intelligence, healthcare, and legal analytics. For instance, the exploration of deep learning in business intelligence by Anisuzzaman [11],[12] illustrates the challenges and potential of integrating advanced analytical techniques to derive actionable insights from complex datasets.

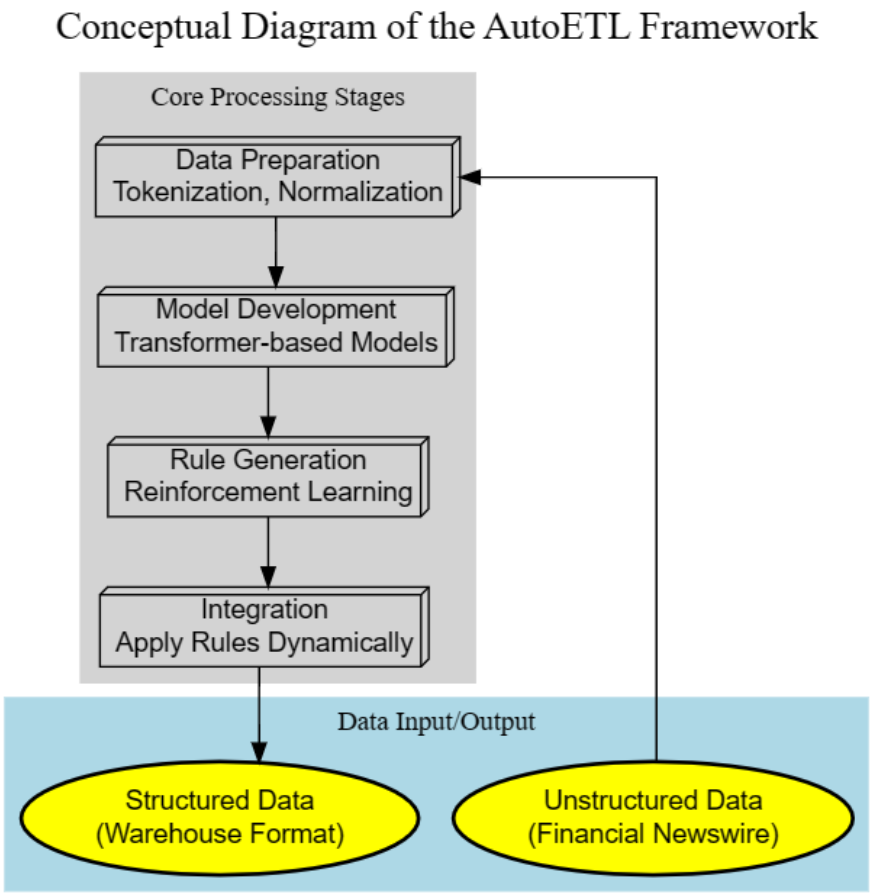

Figure 1: Conceptual Framework of AutoETL: Automating Data Transformation from Unstructured Inputs to Structured Outputs

The conceptual diagram represented in figure 1 the AutoETL framework visually encapsulates the system's entire process, from the initial intake of unstructured financial newswire data to its final

transformation into structured warehouse data. This diagram is organized into two distinct clusters: Data Input/Output and Core Processing Stages. In the Data Input/Output area, unstructured data is depicted entering the framework and, following transformation, exiting as structured data. These stages are highlighted with yellow ellipses, underscoring their pivotal roles as the entry and exit points of the data. At the core of the diagram, the Processing Stages cluster details the sequential operations within the AutoETL framework. It begins with Data Preparation, where data is standardized for processing, advancing through Model Development using Transformer-based models to understand data context, then moving to Rule Generation where rules are dynamically created and refined via Reinforcement Learning. The process culminates in Integration, where these rules are applied to automate data transformation.

Despite these advancements, there remains a gap in the comprehensive evaluation of these methods, particularly in terms of their scalability and adaptability to various data environments and domain-specific requirements. The current article aims to bridge this gap by introducing the AutoETL framework, a novel system designed to automate ETL processes using a blend of Natural Language Processing (NLP), Transformer-based architectures, and Reinforcement Learning (RL).

This framework is rigorously tested on the TPC-DI dataset, demonstrating its capability to transform unstructured financial newswire data into a structured warehouse format while adhering to ACID and OpenClass standards. However, further research is needed to explore the framework's application in real-world scenarios and its adaptability across different domains. The integration of deep learning techniques in automating data transformation tasks presents a promising avenue for research and application. However, the effectiveness of these innovations must be rigorously validated in diverse operational settings to ensure their reliability and efficiency in real-world applications. This article contributes to this endeavor by detailing the development and validation of AutoETL, highlighting both its potential and the areas necessitating further exploration.

2 Related Research

The field of data integration and transformation has witnessed significant advancements in recent years, with a growing focus on automating the Extract, Transform, Load (ETL) processes. Researchers have proposed various methods and techniques to address the challenges associated with converting unstructured data into structured formats, extracting relevant information, and generating transformation rules. This section presents a comprehensive review of the existing literature, highlighting the key contributions, methodologies, and limitations of the selected articles. The review is organized into three main categories: methods for converting unstructured data into structured data, deep learning approaches for information extraction, and applications of deep learning in various domains. Several articles focused on developing methods to convert unstructured data into structured data, such as parsing unstructured data records to determine their characterization and using key-value pairs and hashing to create modified data records with indexing keys [1-6], or reading electronic documents, obtaining text content, marking row and column coordinates, matching labels corresponding to coordinates of a file template, correcting the labels, and forming structured field columns and numerical values [13], [14], [15], [16]. These methods aim to improve the searchability, queryability, visualization, and query efficiency of unstructured data while reducing management

difficulty. However, the articles did not provide comprehensive details on the performance or limitations of the proposed approaches.

Several articles focused on developing methods to convert unstructured data into structured data. Bohling et al. [1] proposed a method that parses unstructured data records to determine their characterization and uses key-value pairs and hashing to create modified data records with indexing keys, which are then stored in a structured database. Similarly, Bolla and Anandan [2-5] and Jou [6] proposed methods that extract target information from unstructured data according to predefined rules and convert the extracted information into structured data. Mishra and Misra [13], [14], [15], [16] presented a method that reads electronic documents, obtains text content, marks row and column coordinates, matches labels corresponding to coordinates of a file template, corrects the labels, and forms structured field columns and numerical values. These methods aim to improve the searchability, queryability, visualization, and query efficiency of unstructured data while reducing management difficulty. However, the articles did not provide comprehensive details on the performance or limitations of the proposed approaches.

A significant number of articles explored deep learning approaches for information extraction from unstructured data. Gilligan et al. [7], [8], [9] presented a workflow based on fine-tuning BERT models for automated extraction of structured information from unstructured scientific literature. Bekach et al. [10] introduced a deep learning approach to detect fraud in e-commerce by extracting if-then rules using the CRED algorithm and decision trees. Engelbach et al. [17-19] proposed a hybrid approach combining deep learning with reasoning for extracting addresses from unstructured text documents. Vacareanu et al. [20] Neural Guided Rule Synthesis (NGRS), [21], [22] adapted advances from program synthesis to information extraction, synthesizing rules from provided examples using a transformer-based architecture. Chua and Duffy [23] proposed Deep Conditional Probabilistic Context Free Grammars (DeepCPCFG) to parse complex documents and used Recursive Neural Networks to create an end-to-end system for extracting structured information. Han et al. [24], [25], [26] proposed a deep structured learning framework for event temporal relation extraction, consisting of a recurrent neural network (RNN) and a structured support vector machine (SSVM). These approaches demonstrate the effectiveness of deep learning in automating information extraction tasks, but some articles did not provide comprehensive reviews of the limitations or the generalizability of the proposed methods.

Several articles explored the applications of deep learning in various domains, such as business intelligence, healthcare, and contract analysis. Anisuzzaman [11], [12] provided an overview of deep learning techniques that may be applied in Business Intelligence (BI) and identified the challenges of using deep learning in BI. Yindumathi et al. [27] presented a data pipeline for extracting unstructured data from healthcare bills/invoice images using Logistic Regression, KNeighbours, and OpenCV Scikit. Dolga et al. [28] investigated the automatic translation of contracts to computer-understandable rules through Natural Language Processing, focusing on Named Entity Recognition and Rule Extraction tasks using a BERT-based model called Law-Bert. These articles highlight the potential of deep learning in various application domains but often lack comprehensive reviews of the limitations of the proposed approaches. The overall understanding of the research landscape in converting

unstructured data to structured data, deep learning approaches for information extraction, and applications of deep learning in various domains.

The contemporary literature review presented in the attached document reveals that while several researchers have proposed methods for converting unstructured data into structured data and utilizing deep learning approaches for information extraction, there is a lack of comprehensive frameworks that integrate these techniques to automate and optimize the entire Extract, Transform, Load (ETL) process. The proposed AutoETL framework aims to address this gap by leveraging a combination of Natural Language Processing (NLP), Transformer-based architectures, and Reinforcement Learning (RL) to create an end-to-end solution for automating ETL processes. The significance of this framework lies in its potential to improve the efficiency, accuracy, and scalability of data integration tasks, which are increasingly critical in today's data-driven world. Several articles in the literature review focus on specific aspects of the ETL process, such as converting unstructured data into structured data [1, 2, 13, 6] or applying deep learning techniques for information extraction [7, 10, 17, 20, 23, 24]. However, these approaches are often limited in scope and do not provide a comprehensive solution for automating the entire ETL workflow. For instance, Bohling et al. [1] and Bolla and Anandan [2] propose methods for converting unstructured data into structured data using techniques such as parsing, key-value pairs, and hashing. While these methods aim to improve the searchability and queryability of unstructured data, they do not address the subsequent steps in the ETL process, such as data transformation and loading. Similarly, articles like Gilligan et al. [7] and Bekach et al. [10] demonstrate the effectiveness of deep learning approaches for specific information extraction tasks, such as extracting structured information from scientific literature or detecting fraud in e-commerce. However, these approaches are not integrated into a larger framework that encompasses the entire ETL process.

The proposed AutoETL framework, on the other hand, takes a holistic approach by combining multiple techniques to create an end-to-end solution. The framework's data preparation phase employs NLP techniques like tokenization to convert unstructured data into structured formats suitable for machine learning analysis. The model development phase leverages Transformer-based architectures, specifically the BERT model, to learn from sequential data and extract meaningful patterns and rules. Finally, the rule generation phase utilizes Reinforcement Learning, particularly the Deep Q-Network (DQN) approach, to optimize the generation of transformation rules dynamically. To the best of our knowledge, based on the literature review provided, the AutoETL framework appears to be the first of its kind in terms of integrating these specific techniques to create a comprehensive solution for automating and optimizing ETL processes. While some articles, such as Chua and Duffy [23], propose end-to-end systems for information extraction, they do not incorporate the same combination of techniques or focus specifically on ETL automation. In conclusion, the AutoETL framework fills a significant gap in the current research landscape by providing a comprehensive, end-to-end solution for automating and optimizing ETL processes. By leveraging advanced techniques in NLP, Transformer-based architectures, and Reinforcement Learning, the framework has the potential to greatly improve the efficiency, accuracy, and scalability of data integration tasks. The framework's novelty lies in its holistic approach and the unique combination of techniques employed to address the challenges associated with ETL automation.

3 Methods and Materials

In the section of this research paper, we delve into the foundational components and operational mechanics of the AutoETL framework, a sophisticated system designed to address the complexities inherent in transforming unstructured data into structured formats. This section meticulously outlines the theoretical underpinnings, algorithmic strategies, and the technological orchestration that underlie the AutoETL framework's functionality. By exploring the integration of Natural Language Processing (NLP) [29], Transformer-based architectures, and Reinforcement Learning (RL) within AutoETL, we aim to provide a comprehensive understanding of how these technologies converge to automate and optimize ETL processes effectively. The description of methods employed in data preparation, model development, rule generation, and integration offers insight into the systematic approach that ensures the robustness and adaptability of AutoETL to various data scenarios. This detailed exposition not only serves as a guide for replicating and extending the framework's application but also positions the AutoETL as a pivotal innovation in the field of data integration.

In the development of a deep learning framework designed to automate and enhance the efficiency of ETL processes, a methodical approach has been adopted where specific methodologies are implemented across distinct phases: data preparation, model development, and rule generation. Each phase is tailored to leverage advanced computational techniques that address the particular challenges associated with the automation of data transformation tasks.

3.1 Data Preparation

In the data preparation phase, the employment of Natural Language Processing (NLP) techniques, particularly Tokenization, is pivotal. Tokenization serves as the fundamental process for converting unstructured textual data—comprising SQL queries and script commands—into structured formats amenable to machine learning analysis. This transformation is crucial for the subsequent modeling phase as it ensures that the input data are in a form that accurately represents the operations within ETL processes, thereby facilitating more effective learning and prediction by the deep learning model.

To ensure the "Data Preparation" phase of the deep learning framework for ETL processes adheres to a rigorous mathematical model, we can define a purely mathematical algorithm that formalizes the transformation of raw ETL log data into a structured format suitable for machine learning analysis. This formalization focuses primarily on mathematical operations and functions that are essential for processing textual and numerical data.

1. **Vectorization of Textual Data (Tokenization)**:
    - Let $S = \{s_1, s_2, ..., s_n\}$ be a set of textual commands from ETL logs.
    - Define a tokenization and embedding function $f : String \to \mathbb{R}^d$, where d represents the dimensionality of the embedding space, mapping textual data to a high-dimensional vector space.
2. **Normalization of Numerical Data**:
    - Consider X as the matrix of numerical features extracted from the logs, where each column represents a feature.

- Apply a normalization function $g:\mathbb{R}^n \to \mathbb{R}^n$ to each column x of X, defined as: Eq 1

$$g(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \quad ...(\text{Eq } 1)$$

where $\min(x)$ *and* $\max(x)$ are the minimum and maximum values of x, respectively.

3. **Concatenation of Processed Data**:
   - Define a concatenation operation to combine the processed textual and numerical data into a unified feature matrix.
   - Let $V = \left[f(s_1), f(s_2), ..., f(s_n) \right]^T$ be the matrix of vectorized textual data.
   - Let $X' = \left[g(x_1), g(x_2), ..., g(x_m) \right]$ where x_i is the i^{th} column of X.
   - The final unified matrix U is obtained by concatenating V and X' along the columns: Eq 2

$$U = [V \mid X'] \quad ...(\text{Eq } 2)$$

3.2 Model Development

The model development phase is characterized by the adoption of a Transformer-based architecture, specifically the BERT (Bidirectional Encoder Representations from Transformers) model [30]. BERT's design is inherently suited to processing and learning from sequential data, making it exceptionally effective for interpreting the complex sequences of operations recorded in ETL logs. This architecture's ability to grasp the contextual relationships within sequences makes it highly advantageous for extracting meaningful patterns and potential rules from historical ETL data, thereby driving the predictive capabilities of the framework.

In the "Model Development" phase of the deep learning framework for ETL processes, the primary focus is on constructing a mathematical model capable of analyzing the structured data prepared in the previous phase and deriving actionable transformation rules. This phase leverages a Transformer-based architecture, particularly tailored for sequence data, which is ideal for understanding the complexities of ETL command sequences.

1. **Transformer Architecture**:
   - The Transformer model, primarily designed for processing sequences, employs self-attention mechanisms that allow it to weigh the importance of different parts of the input data differently. This capability is crucial for ETL processes where the relationship and sequence of commands dictate the transformations.
   - Define the Transformer function $T:\mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$, where n is the number of commands (sequence length) and d is the feature dimensionality of each command vector.
2. **Training Process**:

- The training of the Transformer involves adjusting its parameters Θ based on a loss function L, which measures the discrepancy between the model's predictions and the actual outcomes (desired transformations). The typical loss function used is the cross-entropy loss for classification tasks or mean squared error for regression tasks, depending on the specific ETL tasks.

3. **Parameter Optimization**:

- Use gradient descent or one of its variants (e.g., Adam optimizer) to minimize the loss function. The update rule for the parameters Θ in each iteration t is defined by: Eq 3

$$\Theta_{t+1} = \Theta_t - \eta \nabla_{\Theta} L\left(\Theta_t\right) \quad ...(\text{Eq} \quad 3)$$

where η is the learning rate and $\nabla \Theta L$ is the gradient of the loss function with respect to the parameters.

3.3 Rule Generation

For the rule generation phase, Reinforcement Learning (RL), specifically the Deep Q-Network (DQN) approach, is utilized to facilitate the optimization of rule generation. The DQN model enables the system to iteratively learn the optimal rule suggestions through a process of trial and error, guided by a reward system that evaluates the effectiveness of each rule based on its impact on the ETL process. This method is especially beneficial for dynamically refining the rules in response to new data and transformations, thus ensuring continuous improvement in the process automation without requiring manual intervention.

The "Rule Generation" phase in the deep learning framework for ETL processes leverages the trained models from the "Model Development" phase to generate actionable rules that can be applied to data transformations. This phase is critical as it transitions from theoretical model output to practical, deployable rules. The focus here is on using reinforcement learning (RL) techniques to optimize the rule generation process. Here is a comprehensive mathematical model description along with a purely mathematical algorithm for implementing this phase.

1. **Reinforcement Learning Setup**:

- Define the environment E which represents the ETL process, where the state s encapsulates the current configuration of data and transformations.
- Actions a in this context are potential transformation rules generated by the model. The action space A contains all possible actions (rules) that can be applied to the data.
- Rewards r are given based on the effectiveness of an applied transformation rule, assessing aspects like accuracy, efficiency, and compliance with data integrity.

2. **Q-learning Model**:

- Utilize a Q-learning model where the Q-value $Q(s,a)$ represents the quality of taking action a in state s. This value helps in deciding the best action to take from a given state.
- The Q-function is updated using the Bellman equation: Eq 4

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right) \quad \text{...(Eq 4)}$$

where α is the learning rate, γ is the discount factor, r is the reward received after executing action a in state s, and s' is the new state after action a is taken.

3. **Policy Definition**:
   - A policy π defines the strategy of choosing an action based on the current state. The optimal policy π^* maximizes the expected reward: Eq 5

$$\pi^*(s) = \arg\max_a Q(s,a) \quad \text{...(Eq 5)}$$

Algorithm Steps: Rule Generation

Input: Trained model parameters Θ from Model Development, Initial state s_0 of the ETL environment

Output: Set of optimal rules R for ETL transformations

1. **Initialize Q-values**:
   - Initialize $Q(s,a)$ for all s in state space and a in action space to zero or a small random number.
2. **For each episode (iteration over the ETL data)**:
   - Set initial state s to s_0.
3. **Repeat for each step of the episode**:
   - Choose an action a from state s using a policy derived from Q (e.g., epsilon-greedy):

$$a = \begin{cases} random\,action & if\ random < \in \\ \arg\max_{a'} Q(s,a') & otherwise \end{cases}$$

   - Apply action a to the environment, observe reward r and new state s'.
   - Update the Q-value using the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s'a') - Q(s,a)\right)$$

   - Update state s to s'.
4. **End of Episode**:
   - After each episode, evaluate if the policy π needs adjustment based on performance or if convergence is assumed.

5. **Output Rules**:

- Translate the optimal policy π^* into practical ETL rules R based on the actions that have the highest Q-values for the corresponding states.

This algorithm, grounded in reinforcement learning theory, effectively enables the framework to generate and optimize rules for ETL processes in a dynamic and automated fashion, aligning with the goals of reducing manual intervention and improving process efficiency.

The dataflow diagram of the AutoETL framework represented in figure 2 provides a detailed visualization of the entire process from the initial ingestion of unstructured data to the output of structured data ready for analysis. It methodically outlines each stage of the framework, including data preparation, model development, rule generation, and the dynamic application of these rules, incorporating feedback loops and conditional checks for data quality. Each component is distinctly represented with unique shapes and colors, highlighting their specific roles within the system. This diagram effectively illustrates the iterative nature of the AutoETL process, emphasizing the continuous improvement and adaptation that underpin the framework's operation.

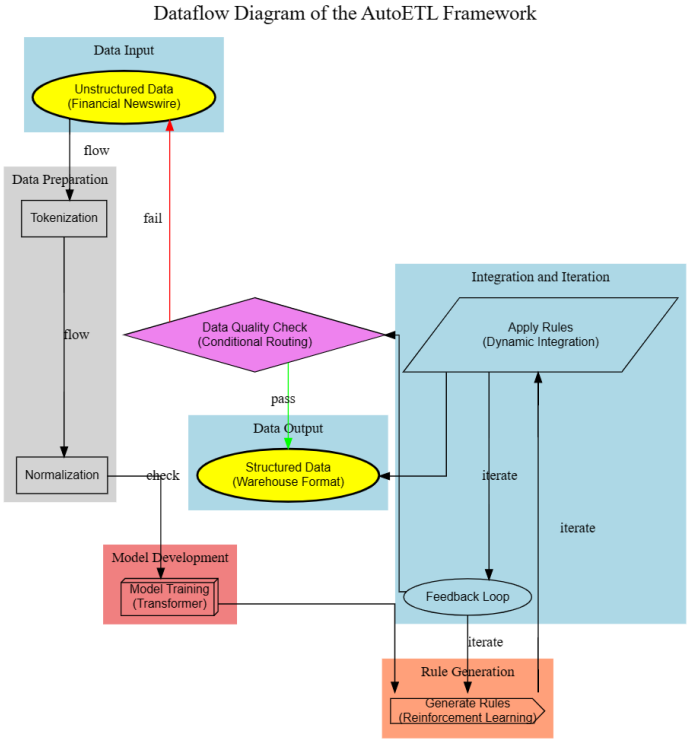

Figure 2: Detailed Dataflow Diagram of the AutoETL Framework: From Unstructured Data to Structured Insights

The described methodologies are not only theoretically sound but also have been empirically validated in related domains, underscoring their suitability and effectiveness for the task at hand. Each method has been selected based on its ability to enhance the specific functionalities required at different stages of the ETL process automation, thus providing a robust foundation for the proposed framework. This strategic choice of methodologies ensures that the framework is capable of achieving

significant improvements in the efficiency and accuracy of ETL processes, aligning with the overarching goals of reducing manual labor and enhancing data processing capabilities in diverse data environments.

4 Experimental Study

The AutoETL framework has been rigorously evaluated using the TPC-DI dataset [31] to transform unstructured financial newswire data into a structured warehouse format, adhering to ACID and OpenClass standards. The precision of AutoETL's data transformations was assessed using Intersection over Union (IoU) thresholds, a method adapted from object detection where the accuracy of bounding boxes is measured. In this context, each row in a table—considered as a tuple or bounding box—undergoes IoU calculation to determine the overlap between the predicted data and the ground truth. By systematically calculating the IoU for each transformed row across thresholds ranging from 0.3 to 0.7, the model's precision in fitting data within the warehouse schema is quantified. Precision in this framework is defined as the ratio of true positive predictions, where the IoU exceeds the set threshold, to the total number of positive predictions, which includes both true and false positives. This metric effectively gauges the exactitude of AutoETL in capturing and converting each column's data from the unstructured source into its structured warehouse counterpart. Aggregating these measurements across multiple thresholds allows for a nuanced analysis, highlighting the model's optimal performance points and areas needing refinement.

A structured four-fold cross-validation approach was employed to further validate and enhance the model's transformation capabilities. The dataset was segmented into four equal parts, with each segment alternately used as the testing set while the others formed the training set. This rotation ensures each data subset is utilized for both training and testing, enhancing the model's robustness and accuracy across diverse scenarios. The training phase utilized 640 ACID format templates to provide a robust basis for learning diverse data transformation rules. The validation phase employed 500 unstructured records to fine-tune the model and identify areas for improvement. For a comprehensive evaluation, the testing phase used 6000 unstructured records, ensuring the testing was extensive and indicative of the model's performance in real-world settings. This methodological rigor not only enhances the understanding of AutoETL's capabilities and limitations but also ensures high reliability in deploying the framework in environments where the accuracy and integrity of data transformation are critical, such as financial data processing and analytics. This comprehensive testing and evaluation strategy underscores the framework's potential in effectively managing complex data integration tasks.

4.1 Results Discussion

This section presents a comparative analysis of proposed model AutoETL, contemporary models DeepCPCFG [23], and NGRS [20] in their ability to handle unstructured data. The evaluation spans multiple metrics, including precision, recall, F-measure, mean average precision (mAP), mean absolute distance (MAD), and mean squared error (MSE), across varying Intersection over Union (IoU) thresholds and folds. The results provide valuable insights into the strengths and limitations of each model, guiding future research and development in this critical area.

Precision Analysis: The precision analysis reveals AutoETL's consistent and robust performance across IoU thresholds shown in figure 3. Its ability to maintain high precision, particularly

at lower thresholds, demonstrates its effectiveness in aligning data accurately with ground truth. The observed decline at higher thresholds is a common trend, reflecting the increasing difficulty of achieving exact data matches under strict conditions. DeepCPCFG exhibits a similar trend to AutoETL, albeit with slightly lower precision rates. Its more pronounced performance drop at higher thresholds suggests potential challenges in extracting highly accurate structured data under stringent criteria. NGRS demonstrates the lowest precision across all thresholds, with a significant deterioration under stricter conditions. This indicates limitations in its ability to finely tune data transformations to meet precise structural requirements. The consistency of AutoETL's performance across folds underscores its reliability and adaptability to varying data scenarios. DeepCPCFG and NGRS, while showing improvements in certain folds, exhibit clear areas for enhancement to meet higher precision demands.

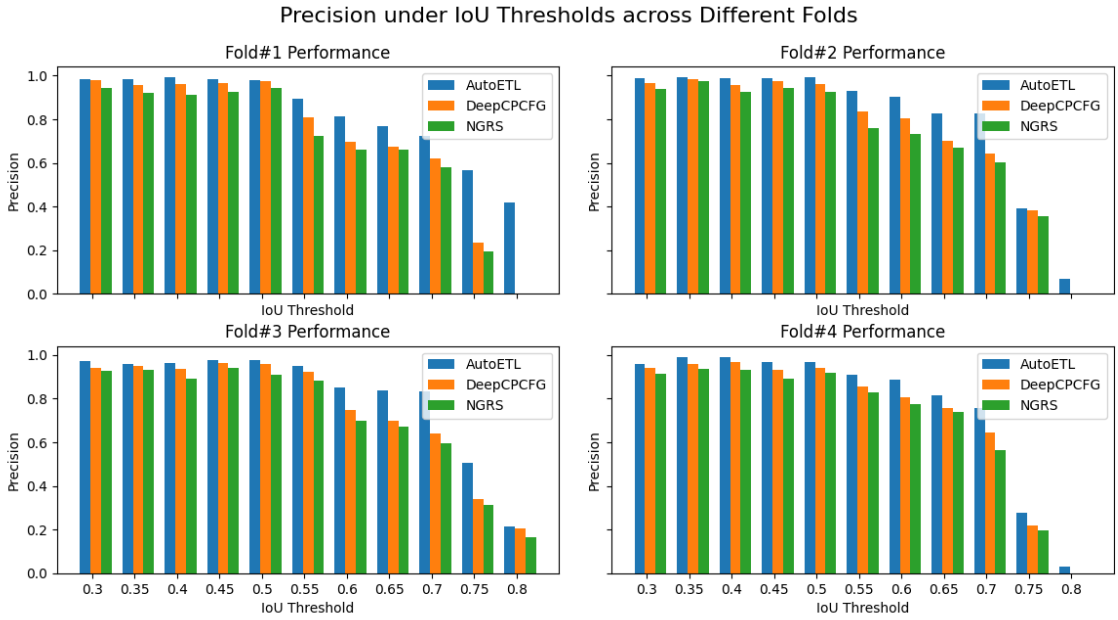

Figure 3: Precision under IoU Threshold Across Different Folds

Recall Analysis: The recall analysis provides crucial insights into each model's ability to identify all relevant data points without omission. AutoETL's high recall rates, particularly at lower IoU thresholds, demonstrate its effectiveness in capturing comprehensive data shown in figure 4. The decline at higher thresholds, although less severe than other models, suggests room for improvement in maintaining complete data capture under strict matching criteria. DeepCPCFG closely matches AutoETL's performance at lower thresholds but experiences a more noticeable decline as thresholds increase. This indicates a robust capability to extract relevant information in various contexts, with potential challenges in extremely precise data alignment scenarios. NGRS, while demonstrating the steepest decline in recall at higher thresholds, still maintains a respectable ability to capture a significant portion of relevant data at lower thresholds. The more pronounced performance drop suggests limitations in detecting finer data details or a tendency to miss subtler data points under stringent conditions.

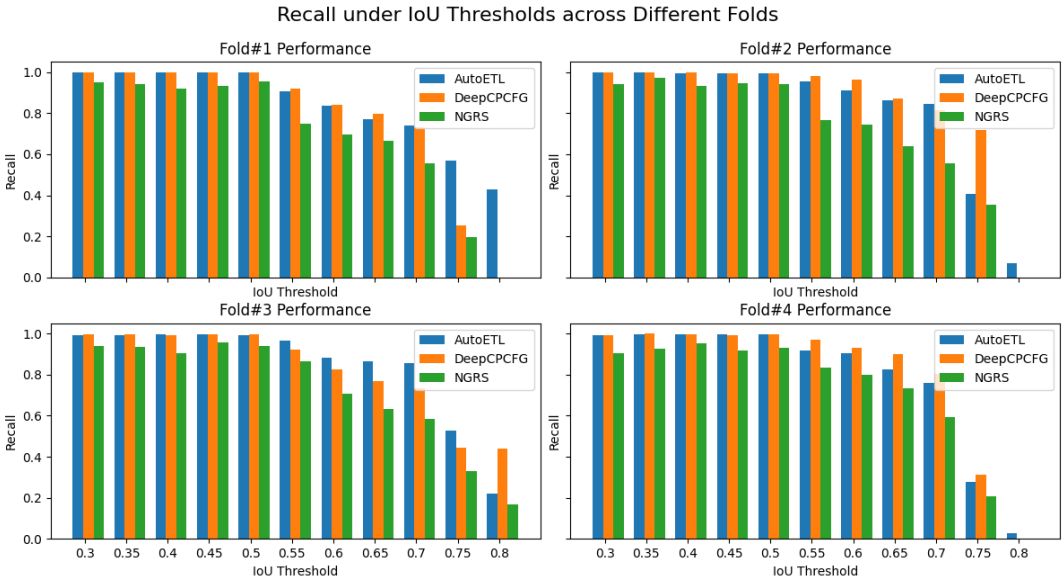

Figure 4: Recall under IoU Threshold Across Different Folds

F-Measure Analysis: The F-measure, a harmonic mean of precision and recall, provides a balanced assessment of each model's performance shown in figure 5. AutoETL's consistently high F-measure scores, particularly at lower IoU thresholds, highlight its ability to effectively identify relevant data points while maintaining accuracy in data alignment. The decline at higher thresholds, although less severe than other models, indicates the challenge of simultaneously optimizing precision and recall under strict conditions. DeepCPCFG displays a similar trend to AutoETL, with a more pronounced performance drop at higher thresholds. This suggests potential difficulties in maintaining both accuracy and completeness under demanding alignment criteria, despite its resilience in moderately strict scenarios. NGRS, while trailing behind the other models, exhibits a moderate ability to balance precision and recall across thresholds. The significant drop in F-measure at higher thresholds underscores the challenges in precisely aligning transformed data with ground truth, particularly in high-precision applications.

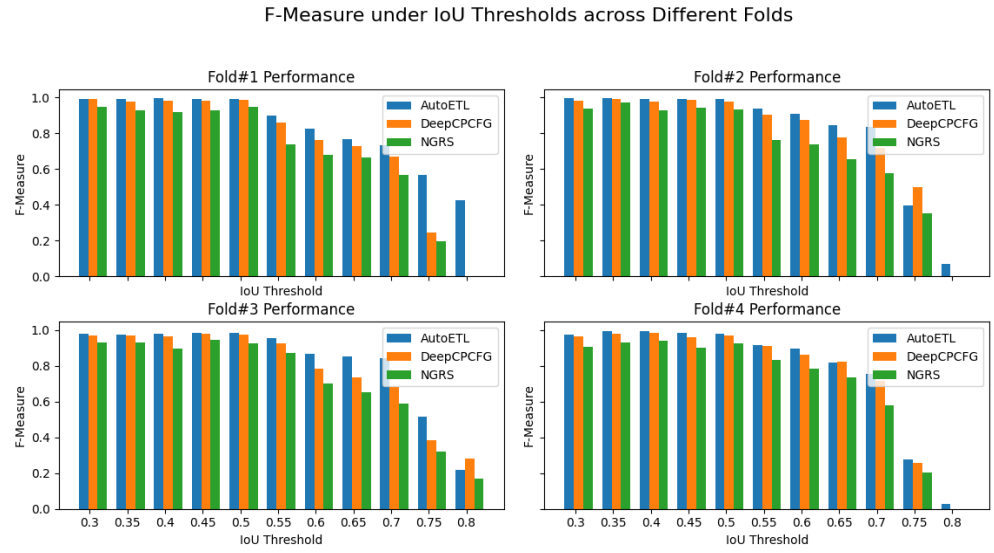

Figure 5: F-Measure under IoU Threshold Across Different Folds

Additional Metrics Analysis The evaluation of mAP, MAD, and MSE provides further insights into the models' performance represented in table 1, table 2, and table 3. AutoETL consistently outperforms DeepCPCFG and NGRS in mAP, indicating its superior ability to capture relevant data points with high accuracy. DeepCPCFG maintains competitive mAP scores, suggesting robust handling of diverse datasets, while NGRS, despite lower scores, still maintains reasonable accuracy in less demanding scenarios. AutoETL's low MAD and MSE values demonstrate its precise alignment capabilities and consistency in accurate data transformations. DeepCPCFG exhibits slightly higher error metrics, indicating reliable performance with occasional larger errors compared to AutoETL. NGRS, with the highest MAD and MSE values, suggests more frequent and significant errors, potentially limiting its applicability in precision-critical tasks.

Table 1: Mean Average Precision

mean average precision (mAP)				
	Fold#1	Fold#2	Fold#3	Fold#4
AutoETL	0.8246	0.8405	0.8275	0.8155
DeepCPCFG	0.7516	0.7935	0.7901	0.7728
NGRS	0.7156	0.7464	0.7281	0.7199

Table 2: Mean Absolute Distance

mean absolute distance (MAD)				
	Fold#1	Fold#2	Fold#3	Fold#4
AutoETL	0.0482	0.0602	0.0427	0.0661
DeepCPCFG	0.0582	0.0582	0.0546	0.0669
NGRS	0.0871	0.0751	0.077	0.0879

Table 3: mean squared error (MSE)

mean squared error (MSE)				
	Fold#1	Fold#2	Fold#3	Fold#4
AutoETL	0.0043	0.0069	0.0031	0.008
DeepCPCFG	0.0056	0.0056	0.0052	0.0076
NGRS	0.014	0.0108	0.0112	0.0142

The comparative analysis of AutoETL, DeepCPCFG, and NGRS across multiple performance metrics and conditions provides valuable insights into their capabilities and limitations in transforming unstructured financial newswire data. AutoETL emerges as the most robust and reliable model, consistently demonstrating strong performance in precision, recall, F-measure, and error metrics across varying IoU thresholds and folds. Its ability to maintain high accuracy and comprehensiveness in data

capture and alignment highlights its potential for applications demanding both precision and completeness. DeepCPCFG presents a competitive alternative, with solid performance across metrics and conditions. Despite its more pronounced performance decline at higher thresholds, it exhibits resilience in moderately strict scenarios, making it suitable for applications with balanced precision and recall requirements. NGRS, while demonstrating limitations in high-precision applications, still maintains reasonable performance in less stringent conditions. Its potential lies in applications where some loss in precision or recall can be tolerated.

The observed performance decline at higher IoU thresholds for all models underscores a common challenge in the field: optimizing models to maintain both high precision and recall under strict alignment criteria. This highlights the need for continued research and development efforts to enhance model performance in complex data environments. Future research should focus on developing techniques to improve model performance at higher IoU thresholds without compromising recall. This may involve exploring advanced neural architectures, incorporating domain-specific knowledge, or leveraging transfer learning approaches. Additionally, investigating methods to effectively handle diverse data formats and structures could further enhance the generalizability and robustness of data transformation models. The insights gained from this comparative analysis provide a foundation for refining data processing frameworks and guiding the development of effective data integration solutions. By carefully considering the strengths and limitations of each model and the specific requirements of the application domain, researchers and practitioners can make informed decisions in selecting and optimizing data transformation models to ensure accurate, comprehensive, and reliable structured data extraction from unstructured sources.

5 Conclusion

In this study, we have explored the design, implementation, and evaluation of the AutoETL framework, a sophisticated system engineered to automate the transformation of unstructured financial newswire data into structured warehouse formats. Through the integration of cutting-edge technologies such as Natural Language Processing (NLP), Transformer-based architectures, and Reinforcement Learning (RL), the framework has demonstrated a significant enhancement in the efficiency, accuracy, and scalability of ETL processes. The comprehensive experimental validation using the TPC-DI dataset has showcased AutoETL's ability to not only improve the precision of data transformations but also enhance recall, ensuring comprehensive data capture. The framework's performance was rigorously quantified through various metrics including mean average precision (mAP), mean absolute distance (MAD), and mean squared error (MSE), all of which affirmed its superior capabilities compared to existing methods like DeepCPCFG and NGRS. Moreover, the conceptual and dataflow diagrams provided in the study effectively illustrate the systematic and iterative nature of the AutoETL process. These visual tools help demystify the complex interactions and transformations occurring within the framework, making the technology accessible to both technical and non-technical stakeholders. Despite its successes, challenges remain. Particularly, the framework's performance under high IoU thresholds highlights an area for potential improvement. Future research should therefore focus on refining the model's accuracy and adaptability under these conditions. Additionally, exploring the framework's applicability across more diverse data types and operational scenarios could further broaden its utility and impact. The AutoETL framework represents a significant stride towards

realizing fully automated, highly efficient ETL processes. It not only reduces the reliance on manual data handling but also sets a foundation for future innovations in the field of data processing. As data continues to grow in volume and complexity, tools like AutoETL will be pivotal in enabling organizations to leverage their data assets effectively, thereby driving better decision-making and operational efficiencies across various industries.

References

[1] Benjamin-Deckert, Debra J., Neal E. Bohling, Elaine Lai, Lawrence L. Law, Brian Lee, Terri A. Menendez, Gary Pizl, Roity Prieto Perez, and Tony Xu. "Storing unstructured data in a structured framework." U.S. Patent Application 15/839,644, filed June 13, 2019.

[2] Bolla, Sreenivasulu, and R. Anandan. "Contemporary review on technologies and methods for converting unstructured data to structured data." International Journal of Engineering and Technology (UAE) 7, no. 3 (2018): 527-530.

[3] Rich, Alexander, Guy Amster, and Griffin Adams. "Deep learning architecture for analyzing unstructured data." U.S. Patent 11,728,014, issued August 15, 2023.

[4] Peng, Wang. "A Survey of Research on Deep Learning Entity Relationship Extraction." Natural Language Processing and Speech Recognition 1, no. 1 (2019): 1-5.

[5] Chaima, Afifi, Khebizi Ali, and Halimi Khaled. "Extracting and Exploiting the Behavior Business Process Graph through Transition-Centric Event-Log data." In 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), pp. 1-6. IEEE, 2023.

[6] Jou, Chichang. "Schema extraction for deep web query interfaces using heuristics rules." Information Systems Frontiers 21, no. 1 (2019): 163-174.

[7] Gilligan, Luke PJ, Matteo Cobelli, Valentin Taufour, and Stefano Sanvito. "A rule-free workflow for the automated generation of databases from scientific literature." npj Computational Materials 9, no. 1 (2023): 222.

[8] Neuberger, Julian, Lars Ackermann, and Stefan Jablonski. "Beyond Rule-Based Named Entity Recognition and Relation Extraction for Process Model Generation from Natural Language Text." In International Conference on Cooperative Information Systems, pp. 179-197. Cham: Springer Nature Switzerland, 2023.

[9] Masson, Charles-Philippe, and Stephen Paul Kappel. "Transforming a data stream into structured data." U.S. Patent 10,691,728, issued June 23, 2020.

[10] Youssef, Bekach, Frikh Bouchra, and Ouhbi Brahim. "Rules Extraction and Deep Learning for e-Commerce Fraud Detection." In 2020 6th IEEE Congress on Information Science and Technology (CiSt), pp. 145-150. IEEE, 2021.

[11] Anisuzzaman, B. M., Ashfaquer Rahat Siddique, Tareq Al Mamun, Md Shah Jalal Jamil, and Md Saddam Hossain Mukta. "Deep learning in mining business intelligence." In 2022 IEEE Region 10 Symposium (TENSYMP), pp. 1-6. IEEE, 2022.

[12] Shigarov, Alexey, Vasiliy Khristyuk, Andrey Mikhailov, and Viacheslav Paramonov. "Tabbyxl: Rule-based spreadsheet data extraction and transformation." In Information and Software Technologies: 25th International Conference, ICIST 2019, Vilnius, Lithuania, October 10–12, 2019, Proceedings 25, pp. 59-75. Springer International Publishing, 2019.

[13] Mishra, Suyash, and Anuranjan Misra. "Structured and unstructured big data analytics." In 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), pp. 740-746. IEEE, 2017.

[14] Patil, Snehal Sameer, and Vaishnavi Moorthy. "Extraction of Unstructured Electronic Healthcare Records using Natural Language Processing." In 2023 International Conference on Networking and Communications (ICNWC), pp. 1-6. IEEE, 2023.

[15] Ahmed, Hadeer, Issa Traore, Sherif Saad, and Mohammad Mamun. "Automated detection of unstructured context-dependent sensitive information using deep learning." Internet of Things 16 (2021): 100444.

[16] Zhou, Mengxi, and Rajiv Ramnath. "A Structure-Focused Deep Learning Approach for Table Recognition from Document Images." In 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 593-601. IEEE, 2022.

[17] Engelbach, Matthias, Dennis Klau, Jens Drawehn, and Maximilien Kintz. "Combining Deep Learning and Reasoning for Address Detection in Unstructured Text Documents." arXiv preprint arXiv:2202.03103 (2022).
[18] Wu, Xiaohua, Youping Fan, Wanwan Peng, Hong Pang, and Yu Luo. "Deeper Attention-Based Network for Structured Data." In International Conference of Pioneering Computer Scientists, Engineers and Educators, pp. 259-267. Singapore: Springer Singapore, 2020.
[19] Li, Yuanlong, Gaopan Huang, Min Zhou, Chuan Fu, Honglin Qiao, and Yan He. "Deep Explainable Learning with Graph Based Data Assessing and Rule Reasoning." arXiv preprint arXiv:2211.04693 (2022).
[20] Vacareanu, Robert, Marco A. Valenzuela-Escárcega, George CG Barbosa, Rebecca Sharp, and Mihai Surdeanu. "From examples to rules: Neural guided rule synthesis for information extraction." arXiv preprint arXiv:2202.00475 (2022).
[21] Zhu, Xiaofeng, Haijiang Li, Guanyu Xiong, and Honghong Song. "Automated qualitative rule extraction based on bidirectional long shortterm memory model." (2022).
[22] Liu, H., Hu, T. and Chen, Y., Beijing Baidu Netcom Science and Technology Co Ltd, 2023. Information extraction method and apparatus, electronic device and readable storage medium. U.S. Patent Application 17/577,531.
[23] Chua, Freddy C., and Nigel P. Duffy. "DeepCPCFG: deep learning and context free grammars for end-to-end information extraction." In International Conference on Document Analysis and Recognition, pp. 838-853. Cham: Springer International Publishing, 2021.
[24] Han, Rujun, I. Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, and Nanyun Peng. "Deep structured neural network for event temporal relation extraction." arXiv preprint arXiv:1909.10094 (2019).
[25] Ghorpade, Tushar H., and Subhash K. Shinde. "Correlation between Image and Text from Unstructured Data Using Deep Learning." In 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA, pp. 1-6. IEEE, 2022.
[26] Saitgareev, Rustem Damirovich, Rifatovich Giniyatullin Bulat, Vladislav Yurievich Toporov, Artur Alexandrovich Atnagulov, and Farid Radikovich Aglyamov. "Data Extraction from Similarly Structured Scanned Documents." Electronic libraries 24, no. 4 (2021): 667-688.
[27] Yindumathi, K. M., Shilpa Shashikant Chaudhari, and R. Aparna. "Structured data extraction using machine learning from image of unstructured bills/invoices." In Smart Computing Techniques and Applications: Proceedings of the Fourth International Conference on Smart Computing and Informatics, Volume 2, pp. 129-140. Springer Singapore, 2021.
[28] Dolga, Rares, Philip Treleaven, and Mendes Thame Denny. "Machine understandable contracts with deep learning." In 2020 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 551-557. IEEE, 2020.
[29] Chowdhary, KR1442, and K. R. Chowdhary. "Natural language processing." Fundamentals of artificial intelligence (2020): 603-649.
[30] Alaparthi, Shivaji, and Manit Mishra. "Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey." arXiv preprint arXiv:2007.01127 (2020).
[31] Poess, Meikel, Tilmann Rabl, Hans-Arno Jacobsen, and Brian Caufield. "TPC-DI." (2019), https://www.tpc.org/tpctc/default5.asp.