

From NER to Business Process Automation: Comparative Evaluation of CNN, LSTM, and Transformer Models for Address Intelligence

Saurabh Kumar Srivastava

drsaurabhksrivastava@gmail.com

SCSET, Bennett University <https://orcid.org/0000-0003-4981-4657>

Vikas Jalodia

SCSET, Bennett University <https://orcid.org/0000-0003-4952-5265>

Divya Srivastava

SCSET, Bennett University <https://orcid.org/0000-0003-3114-7737>

Research Article

Keywords: Business Process Automation, NER, BERT, DistilBERT and ELECTRA

Posted Date: September 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11113348/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

From NER to Business Process Automation: Comparative Evaluation of CNN, LSTM, and Transformer Models for Address Intelligence

Vikas Jalodia, Divya Srivastava, Saurabh Kumar Srivastava

School of Computer Science Engineering & Technology
Bennett University, Greater Noida, UP, India.

Emails: vikasjalodia@gmail.com, divya.srivastava@bennett.edu.in, drsaurabhksrivastava@gmail.com

Abstract

Named Entity Recognition (NER) can support Business Process Automation (BPA) by converting textual addresses into structured components required for logistics, customer-data management, record validation, and related workflows. Previous studies have commonly evaluated individual neural architectures or single-domain datasets, leaving limited evidence on how heterogeneous model families compare under consistent address-extraction conditions. This study comparatively evaluates five neural architectures—WordCNN, WordLSTM, BERT, DistilBERT, and ELECTRA—for token-level extraction of city, state, country, and postal-code components. Experiments were conducted on two heterogeneous datasets, OASIS aerospace supplier records with 12,059 instances and USA Public Schools records with 102,370 instances, across 50%, 80%, and 100% data-availability regimes. Model-specific tokenization preserved architecture-appropriate input representations, while group-aware holdout partitioning reduced geographic overlap between training and held-out evaluation records. Performance was evaluated using Precision, Recall, F1-score, Accuracy, ROC-AUC, and inference latency. Statistical uncertainty was also reported using 95% confidence intervals for the main accuracy results. At the full-data setting, ELECTRA achieved 99.4% accuracy on OASIS (95% CI: 99.00–99.64%) and 99.6% on USA Public Schools (95% CI: 99.50–99.68%). ELECTRA also required substantially lower measured inference latency than BERT. WordCNN and WordLSTM offered computationally lighter substitutes when faster processing was more important than achieving the highest predictive performance. The main contribution of this study is a controlled comparison of different neural architectures that jointly examines predictive performance, data availability, and computational efficiency for BPA-oriented address extraction.

Keywords: Business Process Automation, NER, BERT, DistilBERT and ELECTRA.

1. INTRODUCTION

Business Process Automation (BPA) increasingly integrates Artificial Intelligence (AI), machine learning (ML), and natural language processing (NLP) to automate the tasks related to structured and unstructured business data [1], [2], [3]. Address processing is an important BPA task because

addresses contain useful components of information, such as city, state, country, and postal code. These components support many business applications including logistics, customer management, record validation, and other data-processing workflows. Named Entity Recognition (NER) provides an effective approach to identify these components from address text. It also helps greatly in converting unstructured address text into structured information for downstream automation.

Address information may appear in different formats and may contain variations in abbreviations, ordering, spelling, and numerical representations. These variations make reliable extraction of address components more difficult especially when the same model is evaluated across datasets from different domains. Therefore, evaluating NER architectures under heterogeneous address conditions is important for determining their suitability for practical BPA applications.

Previous studies have primarily examined NER or AI-based automation using individual models, specific datasets, or domain-specific applications, limiting direct comparison across heterogeneous architectures and business environments [1], [3], [4]. In particular, comparisons among CNN-, LSTM-, and transformer-based architectures under consistent experimental conditions remain limited for structured address extraction. Computational efficiency is also less frequently evaluated together with predictive performance, leaving limited evidence for selecting models according to both accuracy and deployment requirements. Therefore, comparative evaluation across heterogeneous address datasets and different training regimes is needed to determine the relative predictive and computational performance of these architectures for BPA-oriented address extraction.

Based on the identified gap, this study addresses the following research question: How do WordCNN, WordLSTM, BERT, DistilBERT, and ELECTRA compare in token-level address-component extraction performance and inference efficiency across heterogeneous datasets and different data-availability regimes? The corresponding hypothesis is that transformer-based architectures will provide stronger predictive performance, whereas lightweight word-level architectures will retain an advantage in computational efficiency.

The organization of this paper is as follows: Section 2 gives the description of the related work in context with this study. Section 3 details the proposed methodology, including the architecture of five models WordCNN, WordLSTM, BERT, DistilBERT & ELECTRA, and two datasets i.e., OASIS & USA Public Schools, along with experimental setup. Section 4 presents results compilation, with performance metrics, i.e., Precision, Recall, F1-Score, AUC, and Inference Time across all models and all data regimes. This section also shares results discussion for analyzing the computational efficiency, performance trade-offs, and providing rational guidelines for model selection in real-world BPA environments. Section 5 discusses the limitations of the study, Section

6 presents the conclusion, Section 7 summarizes future research directions, and finally Section 8 provides Ethics and Data Availability statements.

2. RELATED WORK

2.1. Levels of Process Automation

The development of BPA can be viewed in a hierarchical format of three levels of automation: rule-based systems/scripting engines, robotic process automation (RPA), and intelligent process automation (IPA). Early RPA tools, which are often defined by [5], [6], [7], and were developed to automate repetitive and monotone activities similar to those of order tracking and invoice processing using the same type of deterministic logic utilized by rule-based systems [8]. Although these tools were able to provide a high level of efficiency in completing repetitive and structured tasks, they were limited to executing single process flows. With the rise of unstructured data and semi-structured communication between stakeholders, organizations began to realize the limitations of traditional RPA. Therefore, the next generation of RPA tools known as cognitive RPA (C-RPA) evolved through the combination of RPA with AI/ML capabilities to create adaptive and contextually aware systems that can execute more complex workflows [9]. Especially in healthcare and finance industries, Workflow management systems (WfMS), enterprise resource planning (ERP) integration mechanisms [10], and data-centric middleware were upgraded on the existing BPA settings [11], [12]. This shows the transition of the existing traditional BPA to AI-enabled systems that can handle more complex data and workflows.

2.2. CNN, LSTM and Transformers

Several AI models, including CNNs, LSTMs, and transformers, offer different strengths for business process automation. The integration of AI with BPA has opened new opportunities for feature learning, decision making, and predictive control. Deep neural architectures have also been applied to predictive process monitoring and next-event prediction in business-process data [13]. CNNs and recurrent architectures provide complementary mechanisms for capturing local and sequential patterns, respectively. CNNs are effective at identifying local feature combinations, whereas LSTM-based models can preserve dependencies across ordered observations. Transformer-based pretrained models extend contextual representation through self-attention and have subsequently become important for information-extraction tasks involving unstructured and semi-structured text. However, the most recent and significant advancement in BPA systems has been through the implementation of transformers. The self-attention mechanism enables transformer architectures to construct context-dependent representations without recurrent processing [14].

Additionally, variants such as BERT, FinBERT [15] and biomedical ELECTRA based models [16] have enhanced the ability to perform deep semantic analysis of business documents, compliance records and clinical data streams, thus further establishing the role of transformer models as the backbone of many modern BPA solutions.

2.3. Areas of Implementation

BPA is applied across manufacturing, healthcare, finance, education, and public-sector environments to automate repetitive activities and support data-driven decision making [11], [12], [17], [18], [3]. AI-based methods are increasingly used in these applications to process unstructured information, improve decision support, and provide greater adaptability than conventional rule-based automation [1], [19], [20], [4]. These developments are relevant to address extraction because address information is commonly exchanged as unstructured or inconsistently formatted text within business workflows.

2.4. Need for Automation

The expansion of Business Process Automation (BPA) is at the intersection of business and operational needs. Economic pressures, lack of skilled personnel and need to lower cost have been identified as the primary reasons to automate processes [21], [22]. The increasing need for operational efficiency and service continuity has further encouraged organizations to adopt AI-based automation [18]. Today, companies are prioritizing solutions that make decisions faster, and increase accuracy, while reducing regulatory compliance time. These all represent end-to-end AI automated processes. Applications of AI-enabled automation across document processing, financial information extraction, and enterprise workflows illustrate the breadth of contemporary BPA adoption [1], [18], [20].

2.5. Challenges

While there are many benefits associated with the deployment of BPA, there are also numerous challenges associated with its implementation. Some of the biggest challenges include integrating legacy systems; this requires a middleware solution to connect older systems with newer AI systems [23]. In addition to legacy system integration issues, there are data privacy concerns with respect to regulations like GDPR and HIPAA when implementing BPA, particularly when multiple companies are involved in BPA implementations [12]. Also, companies may face resource limitations when implementing BPA because large models, such as BERT and GPT-4 require large amounts of computer power to run in real time. Techniques such as pruning and distillation have been used to alleviate some of the resource issues associated with implementing BPA. Lightweight transformer architectures provide potential options for resource constrained BPA deployment, although their practical benefit depends on the required accuracy–latency adjustment. Overcoming

these challenges will be essential to realizing the full potential of BPA in many different contexts.

2.6. Research Studies in BPA

Recent BPA and information-extraction research increasingly employs transformer and hybrid neural architectures for processing unstructured and semi-structured information. BERT- and ELECTRA-based approaches have demonstrated strong contextual representation across domain-specific extraction tasks [16], [24], [25], while CNN-, recurrent-, and hybrid architectures remain relevant for capturing local and sequential patterns [26], [27]. These studies support the need to compare heterogeneous neural architectures not only in terms of predictive performance but also according to their computational requirements and suitability for automated information-processing workflows.

Recent studies demonstrate that reliable data processing, intelligent decision support, and process compliance remain the central concerns in practical BPA environments. Benevento et al. [28] showed that incomplete and noisy process data can compromise process discovery and demonstrated how Interactive Process Discovery combines event data with domain knowledge to improve process-model reliability. Deng et al. [29] developed a recommendation framework that mines patterns from existing processes and uses them to support the construction of new business-process models. Adams et al. [30] examined modern business-process compliance challenges and proved the value of process-mining tools for identifying and fixing compliance-related problems. Although these studies address different BPA phases, they jointly highlight the importance of reliable data representation and intelligent processing. The present study extends that work by analysing how various neural architectures transform textual address information into structured components for downstream automated workflows.

2.7. Measuring Parameters

Evaluation of BPA systems involves both operational and model-level measures. Standard metrics for operational performance include process efficiency, operational output, inference latency, decision accuracy, compliance rate, and cost-effectiveness [31], [22]. Model-level performance can be evaluated using Precision, Recall, F1-score, and ROC-AUC [19], [32]. Compliance metrics are highly relevant in regulated environments [18], [30]. Together these measures offer a comprehensive framework for evaluating the effectiveness of BPA systems.

Benchmarking Table 1: Selected Prior BPA Studies Informing the Experimental Design and Research Gap.

SN	Year of Publishing, Source	Paper Title	Technique/Methodology	Dataset	Research Gaps	Results/ Measures used
1.	2021 [7]	Development of Intelligent Robotic Process Automation: A Utility Case Study in Brazil	Machine Learning Models, Data Management Techniques, Statistical Modeling, Proactive Communication Framework, Visualization Tools (Power BI), Cloud Technologies (Azure Data Lake, Azure Data Factory).	http://doi.org/10.5281/zenodo.3995046	Enhance adaptability of robotic process automation.	Prediction accuracy improved from 59% to 83%. Proactive notifications reduced repeat complaints. Real-time processing enhanced communication. Historical data analysis led to targeted communication strategies.
2.	2021 [1]	Efficient Automated Processing of the Unstructured Documents Using Artificial Intelligence: A Systematic Literature Review and Future Directions	Named Entity Recognition (NER), Advanced Optical Character Recognition (OCR), sentiment analysis, and deep learning are employed for processing unstructured data.	-	Identify and address integration challenges.	Quality assessment criteria and statistical tests are utilized to evaluate the datasets, assuring their appropriateness for training AI models and improving extraction accuracy.
3.	2023 [18]	Understanding How Intelligent Process Automation Impacts Business Continuity	Design Science Research (DSR) methodology, focusing on integrating ISO 22301 and IEEE 2755.2 standards.	-	Develop a mapping between IPA and business continuity standards.	Compatibility: 95% rated the compatibility of ISO 22301's "Evaluation" with IEEE 2755.2's "Performance Evaluation" as high. Integration Feedback: Highlighted advantages like improved understanding and relevance of integrating ISO 22301 and IEEE 2755.2.
4.	2024 [3]	Business Process Automation in SMEs: A Systematic Literature Review	Planning (defining objectives, research questions, and review protocols), conducting (filtering articles, quality assessment, data extraction), and reporting (summarizing key findings).	-	Identify and address barriers to adoption.	Business Process Automation (BPA) leverages tools like RPA, C-RPA, Wfms, ERP, and Blockchain. It comprehensively addresses environmental variables for automation in business processes.
5.	2022 [28]	How Can Interactive Process Discovery Address Data Quality Issues in Real Business Settings? Evidence from a Case Study in Healthcare	Interactive Process Discovery (IPD), process mining, integration of event-log data with domain knowledge, controlled experiment, and real-world case study.	-	Noisy and incomplete event logs reduce process-discovery reliability.	IPD improved process-model reliability by combining data with domain knowledge.
6.	2017 [29]	A Recommendation System to Facilitate Business Process Modeling	Two-stage recommendation framework using offline business-process pattern mining and online activity recommendation; three recommendation strategies were evaluated.	Real-world and synthetic business-process datasets.	Business-process modeling lacks effective intelligent recommendation support.	The recommendation system improved modeling accuracy and computational efficiency.
7.	2026 [30]	Addressing the Contemporary Challenges of Business Process Compliance	Analysis of BPC literature and industry evidence, regulatory-report analysis, consultancy insights, assessment of commercial process mining tools.	-	Current research does not fully address practical process-compliance challenges.	Identified 41 consolidated compliance challenges and mapped process-mining capabilities.
8.	2026 [33]	Robotic Process Automation in Business Process Management: A Systematic Literature Review and an Integrated Framework	Systematic Literature Review following PRISMA 2020, systematic content analysis, thematic synthesis, and development of an integrated RPA-BPM framework.	-	RPA-BPM research remains fragmented across implementation, governance, and integration.	Proposed a three-pillar RPA-BPM framework based on six thematic dimensions.
	2026	Present Study	Comparative evaluation of WordCNN, WordLSTM, BERT, DistilBERT, and ELECTRA models for address parsing (City, State, Country, and Postal Code). Model-specific tokenization and group-aware holdout partitioning were used to support consistent model comparison and reduce geographic overlap between training and held-out evaluation records.	OASIS (Aerospace) and USA Public Schools datasets	Limited benchmarking of diverse deep learning architectures for structured address extraction within BPA.	Metrics used for evaluation using Precision, Recall, F1-score, Accuracy, ROC-AUC, and Inference Time. ELECTRA achieved optimal efficiency; transformer models attained high predictive accuracy; CNN/LSTM models balanced performance with lower latency.

The studies summarized in Benchmarking Table 1 indicate three recurring themes relevant to the present work. First, BPA increasingly depends on AI-enabled processing of semi-structured and unstructured information [1], [2], [32]. Second, data quality, contextual representation, and reliable information transformation influence the effectiveness of downstream automated processes [28], [18], [3]. Third, existing BPA and information-extraction studies commonly evaluate specific model families, datasets, or application settings, leaving limited evidence from controlled comparisons of heterogeneous neural architectures for the same structured information-extraction task [16], [26], [27], [34], [25], [15]. These observations motivate the present evaluation of CNN-, recurrent-, and transformer-based models under a common address-extraction protocol.

Benchmarking Table 2: Recent AI-Based Studies Relevant to NER and Comparative Model Selection.

Sn.	Year of Publishing Source	Paper Title	Technique/Methodology	Dataset	Results/Measures Used
1.	2020 [35]	Partial Annotation Scheme for Active Learning on Named Entity Recognition Tasks	Partial annotation, active learning	CoNLL-2003 NER dataset	Partial annotation reduced labeling requirements and enabled faster active-learning-based NER training than conventional full-sentence annotation
2.	2023 [24]	A New Italian Cultural Heritage Dataset: Detecting Fake Reviews With BERT and ELECTRA Leveraging the Sentiment	BERT, ELECTRA, SHAP	Italian Cultural Heritage dataset	95% accuracy, F1-score in fake review detection
3.	2023 [26]	MCNN-LSTM: Combining CNN and LSTM to Classify Multi-Class Text in Imbalanced News Data	MCNN-LSTM, Tomek-Link algorithm	HuffPost news dataset	Accuracy: 99.71%, F1-score: 98%
4.	2024 [36]	Integrating social media and deep learning for real-time urban waterlogging monitoring	BERT-BiLSTM-CRF, topic modeling, transfer learning	Facebook posts from Swat, Pakistan	Deep-learning models achieved over 79% accuracy in waterlogging detection and demonstrated the value of social-media information for real-time flood monitoring.
5.	2024 [37]	Development of a Knowledge Base for Construction Risk Assessments Using BERT and Graph Models	BERT-based entity recognition, graph models	Construction safety reports	Constructed a risk-assessment knowledge base containing 59 nodes and 81 edges; the knowledge-base search function achieved an average effectiveness score of 4.44/5.
6.	2024 [25]	Research on the Financial Event Extraction Method Based on Fin-BERT	Fin-BERT with RATT, structured self-attention	Duee-fin and ChFinAnn datasets	F1-score improved by 2.8% (Duee-fin) and 2.5% (ChFinAnn)
7.	2024 [38]	BiodiViz: Leveraging NER and RE for Automated Knowledge Graph Generation in Biodiversity Research	BiodiViz, BERT, ELECTRA, BiodivBERT	BiodivNERE, BiodivNER, BiodivRE datasets	F1-scores: 77.16% (NER), 81.28% (RE)
8.	2024 [16]	Ensemble pretrained language models to extract biomedical knowledge from literature	Ensemble of BioBERT, PubMedBERT, BioM-ELECTRA	PubMed abstracts	Jaccard Similarity: 0.914 (NER), 0.442 (ChatGPT 4)
9.	2025 [39]	Named Entity Recognition Method Based on Multi-Feature Fusion	BERT, BiLSTM, DGCNN, CRF, multi-head attention	Custom online reviews dataset	F1-score of 86.83%, robustness in noisy data
10.	2025 [27]	Biomedical Named Entity Recognition Using Improved Green Anaconda-Assisted Bi-GRU-Based Hierarchical ResNet Model	IGa-BiHR BNERM, ROBERT-WWM	MACCROBAT Dataset	99.11% accuracy, 6% F1-score improvement

Benchmarking Table 2 summarizes ten representative studies that directly inform the neural architectures and information-extraction setting evaluated in this work. Prior research demonstrates the effectiveness of transformer models such as BERT and ELECTRA for contextual representation, while CNN-, LSTM-, and hybrid architectures remain useful for capturing local and sequential patterns [26], [24]. Domain-specific transformer studies further demonstrate the effectiveness of contextual representations for financial and biomedical information extraction [16], [39], [25]. Recent NER approaches also combine recurrent, convolutional, and transformer components to improve entity recognition under heterogeneous data conditions [27]. Collectively, these studies motivate the comparative evaluation of convolutional, recurrent, and transformer architectures under a common address-component extraction protocol. Recent NER studies further illustrate complementary design considerations: partial-annotation strategies can reduce annotation requirements [35], transformer–recurrent architectures can extract location-related information from noisy social-media text [36], BERT-based entity extraction can be integrated with graph representations [37], and multi-feature fusion can combine heterogeneous contextual and sequential features for NER [39].

The reviewed studies also informed specific choices in the present experimental design. Prior NER research showing the value of local, sequential, and contextual representations motivated the inclusion of WordCNN, WordLSTM, and transformer architectures within the same comparison [16], [26], [27], [34], [25], [15]. Studies on compact pretrained encoders and computationally

efficient pretraining further motivated joint reporting of predictive performance and inference latency [40], [41]. In addition, evidence that data quality can materially influence data-driven business processes motivated conservative preprocessing and leakage-aware partitioning of the address records [28]. The present study therefore does not treat the benchmarking tables as an independent literature catalogue; their identified methodological gaps directly motivate the architecture selection, evaluation measures, and experimental controls used in the comparative benchmark.

Recent studies further broaden the context of NLP-enabled automation. Garg [42] demonstrated the use of NLP, including NER, for automating business-requirement elicitation from unstructured information, while Alotaibi et al. [43] reviewed AI- and NLP-based techniques for intelligent business-document processing. Studies on sequential and graph-based representation learning, such as DSIM [44] and GIENN [45], demonstrate how temporal, contextual, and relational information can be modelled in complex information systems. Although these studies address different application tasks, they provide useful perspectives for extending BPA-oriented information extraction toward richer document, sequence, and relation-aware representations.

2.8 Identified Gaps

One of the major gaps in existing studies is the limited benchmarking of heterogeneous deep learning architectures under consistent conditions for the same structured entity-extraction task. Most of the studies focus either on architectural frameworks, interoperability, process mining, or qualitative BPA evaluations. Quantitative AI-based studies usually evaluate a single model or a single paradigm (e.g., ML models, transformers, or rule-based approaches) [1], [3]. Despite reported metrics, direct comparisons across CNNs, RNNs, and transformer-based models under identical experimental conditions remain absent. This gap is critical for structured information extraction tasks such as address parsing, which serves as a foundational pillar for Business Process Automation (BPA) [4].

A second major gap is that computational efficiency and deployability are underexplored together with accuracy. Many studies have evaluated models using accuracy, precision, or F1-score, but inference time and scalability trade-offs are rarely evaluated jointly with ROC-AUC, especially across datasets of different sizes and domains. As a result, existing literature provides limited direction on which models are suitable for real-time or large-scale BPA systems, where latency, robustness, and generalization are as critical as predictive performance. A wide variety of business processes further shows that automated frameworks must be able to handle different process structures [46].

2.9 Our Contribution

- This study conducts a comparative evaluation of five deep learning architectures: WordCNN, WordLSTM, BERT, DistilBERT, and ELECTRA for token-level extraction of city, state, country, and postal-code components across two heterogeneous address datasets.
- Evaluated the five architectures under three training-data regimes of 50%, 80%, and 100% to examine the effect of training-data availability on model performance.
- Compared predictive performance using Accuracy, Precision, Recall, F1-score, and ROC-AUC together with inference time to assess the accuracy–efficiency trade-off relevant to BPA deployment.
- Derived practical model-selection guidance by comparing transformer and lightweight neural architectures across the OASIS and USA Public Schools datasets.

The findings have practical relevance for address-processing workflows in logistics, customer management, and data-cleaning applications. Unlike models that require comparatively greater computational resources, the lightweight WordCNN model also performed very well to process the addresses on the evaluated GPU configuration. This observed throughput indicates that WordCNN can provide a computationally efficient option for address-processing workflows where inference latency is an important consideration. The comparison across training regimes also provides evidence on how additional training data affects model performance and can inform data-labelling decisions. For more complex address-processing tasks, ELECTRA provides a favorable balance between predictive performance and computational efficiency among the evaluated transformer models. These findings provide practical guidance for selecting models according to accuracy, latency, and available computational resources.

3. METHODOLOGY AND EXPERIMENTAL SETUP

This section describes the methodology and experimental framework used to compare WordCNN, WordLSTM, BERT, DistilBERT, and ELECTRA models for the token-level address component extraction.

3.1. Objective

This study aims to strengthen the connection between Business Process Automation (BPA) and Artificial Intelligence (AI) by evaluating natural language processing (NLP) models for structured address extraction. Address-component extraction is important in applications such as logistics, customer data management, and enterprise resource planning systems [1], [10], [4]. Existing studies have examined Named Entity Recognition (NER) and BPA, but mostly independently [2], [31]. Therefore, limited work has systematically compared different deep learning architectures for

address extraction under consistent experimental conditions. Hence, this study focuses on the following objectives:

- 3.1.1 **Comparative Analysis:** Evaluate five neural architectures i.e. WordCNN, WordLSTM, BERT, DistilBERT, ELECTRA on OASIS and USA Public Schools datasets to compare how suitable they are for automated address-component extraction in BPA workflows.
- 3.1.2 **Benchmark Performance:** Compare the models in terms of predictive performance, inference efficiency, and data requirements using a common experimental environment based on a Google Colab T4 GPU.
- 3.1.3 **Practical Deployment Guidelines:** Identify suitable models for different BPA requirements by examining the trade-off between predictive performance, inference latency, scalability, and computational resources.
- 3.1.4 **Academic-Industrial Synergy:** Illustrate how academic AI research can directly solve BPA problems and lower the costs of manual data processing while illustrating a paradigm for incorporating leading edge NLP into commercial systems.

Each objective is linked directly to the reported experimental evidence. Objective 3.1.1 is addressed through the comparative evaluation of WordCNN, WordLSTM, BERT, DistilBERT, and ELECTRA on both datasets under the 50%, 80%, and 100% data regimes. Objective 3.1.2 is addressed through the Accuracy, Precision, Recall, F1-score, ROC-AUC, and inference-time results reported in Sections 4.1 and 4.2. Objective 3.1.3 is addressed through the comparative accuracy–latency interpretation and model-selection guidance presented in Sections 4.3 and 4.4. Objective 3.1.4 is addressed by interpreting the observed model trade-offs in relation to BPA-oriented address-processing requirements and practical resource constraints.

Through establishing these benchmarks, this work will contribute to the advancement of the theoretical knowledge base of model efficiency in NER tasks and deliver a practical guidebook for companies so they may automate the processing of addresses.

3.2. Model Architectures for Token-Level Address Component Extraction

Five neural architectures representing convolutional, recurrent, and transformer-based approaches to address component classification were used in this study. Each of the five models represents a different paradigm in natural language processing and therefore different challenges in the parsing of addresses.

The WordCNN architecture (Figure 1) illustrates a multi-scale convolutional neural network (CNN) designed for efficient local pattern extraction [47]. It consists of a 256-dimensional embedding layer followed by three parallel one dimensional (1D) convolutional branches with the

kernel sizes of 3, 5, and 7 tokens, respectively. The design of the parallel branches allows for the simultaneous extraction of short (e.g., state abbreviations), medium (city names), and long (multi-word components) n-gram features. Each convolutional branch contains 64 filters and is followed by batch normalization and ReLU activation. The three branch outputs are subsequently concatenated into a 192-dimensional token-wise representation and passed through fully connected layers of 128 and 64 units before the final four-logit projection. A final linear projection layer generates the four-class output for the four different entities i.e. city, state, country, and postal code.

All five architectures were formulated as token-level sequence-labeling models, following the general neural sequence-labeling paradigm used in named entity recognition [48]. The notation used in Equations (1)–(20) is defined as follows. For each token position i , the final classifier produces a four-dimensional logit vector $Z_i \in \mathbb{R}^4$. The notation “ $\rightarrow 4$ ” in Figures 1–5 therefore denotes a shared four-logit classifier applied independently to every token representation, rather than a single four-class prediction for the complete address. Collectively, the token-level outputs form $Z = [Z_1, Z_2, \dots, Z_L]^T \in \mathbb{R}^{L \times 4}$. No global sequence pooling, attention-based sequence aggregation, or [CLS]-only representation is used for final address-component classification.

Let $X=(x_1, x_2, \dots, x_L)$ denote a tokenized address sequence of length L , where $i \in \{1, 2, \dots, L\}$ denotes a token position and $c \in \{1, 2, 3, 4\}$ denotes one of the four target classes: City, State, Country, and Postal Code. W_e denotes the trainable embedding matrix, while $E \in \mathbb{R}^{L \times d}$ denotes the resulting token-embedding sequence and d is the embedding dimension. The symbol h denotes the hidden size of each LSTM direction. $\text{Conv}_k(\cdot)$ denotes a one-dimensional convolution with kernel width k , and C_k denotes its token-wise feature representation. H_i , U_i , and V_i denote intermediate representations associated with token position i , while model superscripts identify the corresponding architecture. W_j and b_j denote trainable projection matrices and bias vectors. BN, ReLU, GELU, and LayerNorm denote batch normalization, rectified linear unit, Gaussian error linear unit, and layer normalization, respectively, while $\parallel$ denotes feature concatenation. For token i , $Z_i \in \mathbb{R}^4$ denotes the four-class logit vector and $P_{i,c}$ denotes the predicted probability of class c . The reference class label is denoted by y_i , and N_a denotes the number of active token positions for which $y_i \neq -100$. Positions assigned the ignore index -100 do not contribute to the supervised loss. Dropout is applied at the locations shown in Figures 1–5 and is omitted from Equations (1)–(18) for compactness. The mathematical formulation of the WordCNN architecture is represented using the notation and operations in equations (1)–(5).

Embedding Layer:

$$E = W_e [X] \in \mathbb{R}^{L \times d} \quad (1)$$

Parallel Convolutional Layers:

$$C_k = \text{ReLU}(\text{BN}(\text{Conv}_k(E))), k \in \{3, 5, 7\} \quad (2)$$

Token-wise Feature Concatenation:

$$H^{CNN} = [C_3 \| C_5 \| C_7] \in \mathbb{R}^{L \times 192} \quad (3)$$

Token-wise Projection:

$$U_i^{CNN} = \text{ReLU}(W_1 H_i^{CNN} + b_1) \quad (4)$$

Classifier:

$$Z_i^{CNN} = W_3 \text{ReLU}(W_2 U_i^{CNN} + b_2) + b_3 \in \mathbb{R}^4 \quad (5)$$

where $W_1 \in \mathbb{R}^{128 \times 192}$, $W_2 \in \mathbb{R}^{64 \times 128}$, $W_3 \in \mathbb{R}^{4 \times 64}$

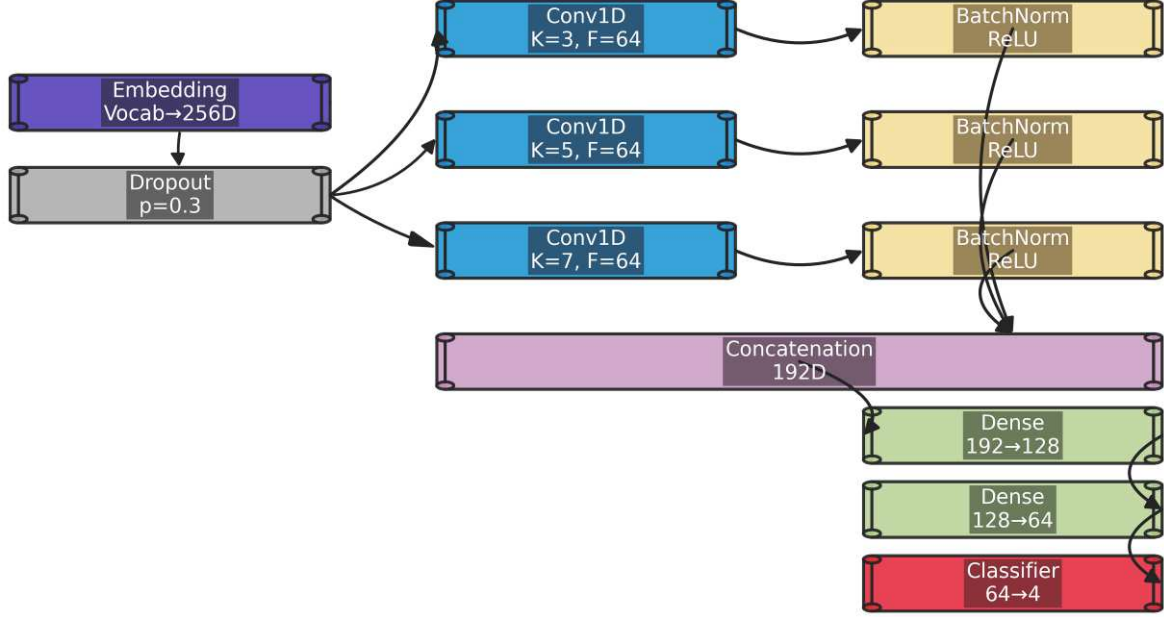


Figure 1. Architecture of WordCNN for Token-Level Address Component Extraction

To model sequential dependencies between address tokens, WordLSTM (see Figure 2) uses stacked bidirectional LSTM layers, building on the LSTM architecture introduced by Hochreiter and Schmidhuber [49] and its established use in neural sequence labeling and NER [48]. The stacked bidirectional layers generate a contextual representation for each token position. Unlike a sequence-level classifier, the hidden-state sequence is preserved throughout the model and each contextual token representation is projected independently to the four address-component classes. No global attention pooling is applied. Bidirectional processing allows the representation of a token to incorporate both preceding and following context, which is useful when the interpretation of a location term depends on neighbouring components. At the full-data regime, WordLSTM achieved 92.1% accuracy on OASIS compared with 91.2% for WordCNN, while requiring greater inference time. The result indicates a modest but consistent predictive benefit from bidirectional sequential context rather than from an attention mechanism.

Bidirectional LSTM Layers:

$$H^{(1)} = \text{BiLSTM}_1(E) \in \mathbb{R}^{L \times 2h} \quad (6)$$

$$H^{(2)} = \text{BiLSTM}_2(H^{(1)}) \in \mathbb{R}^{L \times 2h} \quad (7)$$

(where $h=128$ denotes the hidden size of each LSTM direction)

Contextual Representation Normalization:

$$\tilde{H}^{(2)} = \text{LayerNorm}(H^{(2)}) \quad (8)$$

Classifier:

$$Z_i^{\text{LSTM}} = W_3 \text{ReLU}(W_2 \text{ReLU}(W_1 \tilde{H}_i^{(2)} + b_1) + b_2) + b_3 \in \mathbb{R}^4 \quad (9)$$

Where:

$$W_1 \in \mathbb{R}^{128 \times 256}$$

$$W_2 \in \mathbb{R}^{64 \times 128}$$

$$W_3 \in \mathbb{R}^{4 \times 64}$$

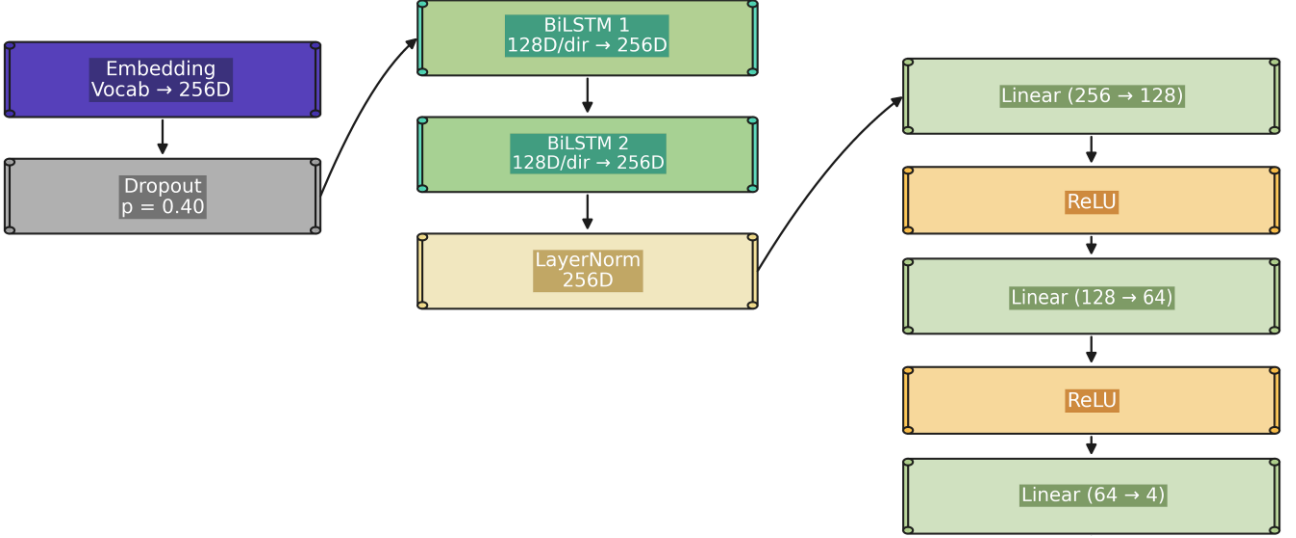


Figure 2. Architecture of the BiLSTM Network for Token-Level Address Component Extraction

We also tested transformer-based models to assess the influence of pre-trained contextual representations. The BERT [14] architecture (see Figure 3) utilizes a 12-layer transformer encoder (with 768-D hidden states) trained using a masked language modeling task. For address extraction, we replace the standard classification head with a custom two-layer MLP ($768 \rightarrow 384 \rightarrow 4$) with GELU activation and layer normalization. BERT achieved high predictive performance on both datasets, but its larger parameter count resulted in substantially higher inference latency than the lightweight architectures. BERT-based architecture is formally described in Equations (10)–(12):

BERT Encoder:

$$H^{BERT} = \text{BERT}(X) \in \mathbb{R}^{L \times 768} \quad (10)$$

$$U_i^{BERT} = \text{LayerNorm}(\text{GELU}(W_1 H_i^{BERT} + b_1)) \quad (11)$$

Classifier Head:

$$Z_i^{BERT} = W_2 U_i^{BERT} + b_2 \in \mathbb{R}^4 \quad (12)$$

Where:

$$W_1 \in \mathbb{R}^{384 \times 768}$$

$$W_2 \in \mathbb{R}^{4 \times 384}$$

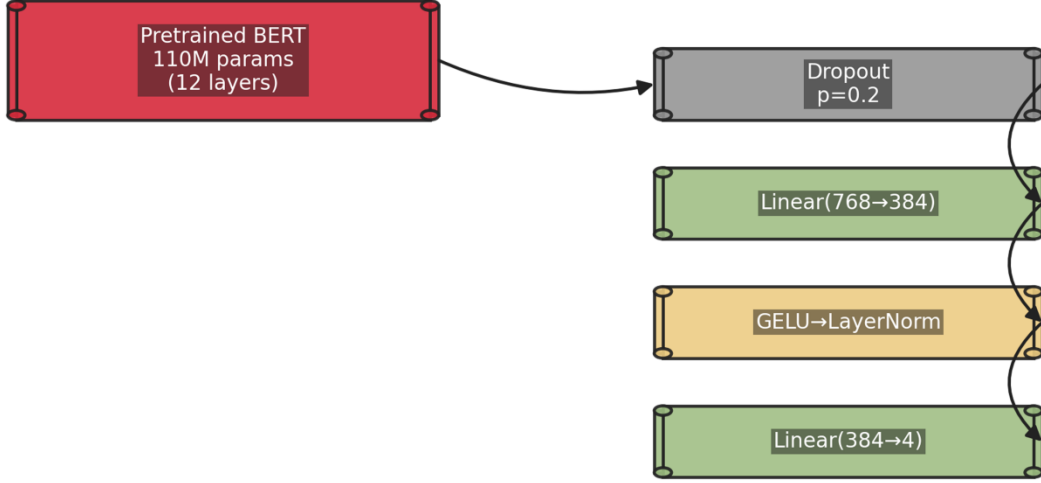


Figure 3. Architecture of BERT for Token-Level Address Component Extraction

To address BERT's computational constraints, we examined DistilBERT [40] (see Figure 4), a six-layer knowledge-distilled variant of BERT designed to reduce model size and inference cost while retaining most of BERT's language-understanding capability. In the present experiments, DistilBERT required substantially lower inference time than BERT while maintaining high predictive accuracy. To account for the reduced pretrained capacity, we extended the classification head to include three linear layers ($768 \rightarrow 384 \rightarrow 192 \rightarrow 4$) and added additional layer normalization to improve training stability. The DistilBERT variant is formalized in Equations (13–15) as a combination of a distilled 6-layer transformer encoder:

DistilBERT Encoder:

$$H^{DB} = \text{DistilBERT}(X) \in \mathbb{R}^{L \times 768} \quad (13)$$

$$\begin{aligned} V_i^{DB} &= \text{GELU}(\text{LayerNorm}(W_1 H_i^{DB} + b_1)) \\ U_i^{DB} &= \text{GELU}(\text{LayerNorm}(W_2 V_i^{DB} + b_2)) \end{aligned} \quad (14)$$

Classifier Head:

$$Z_i^{DB} = W_3 U_i^{DB} + b_3 \in \mathbb{R}^4 \quad (15)$$

Where:

$$W_1 \in \mathbb{R}^{384 \times 768}$$

$$W_2 \in \mathbb{R}^{192 \times 384}$$

$$W_3 \in \mathbb{R}^{4 \times 192}$$

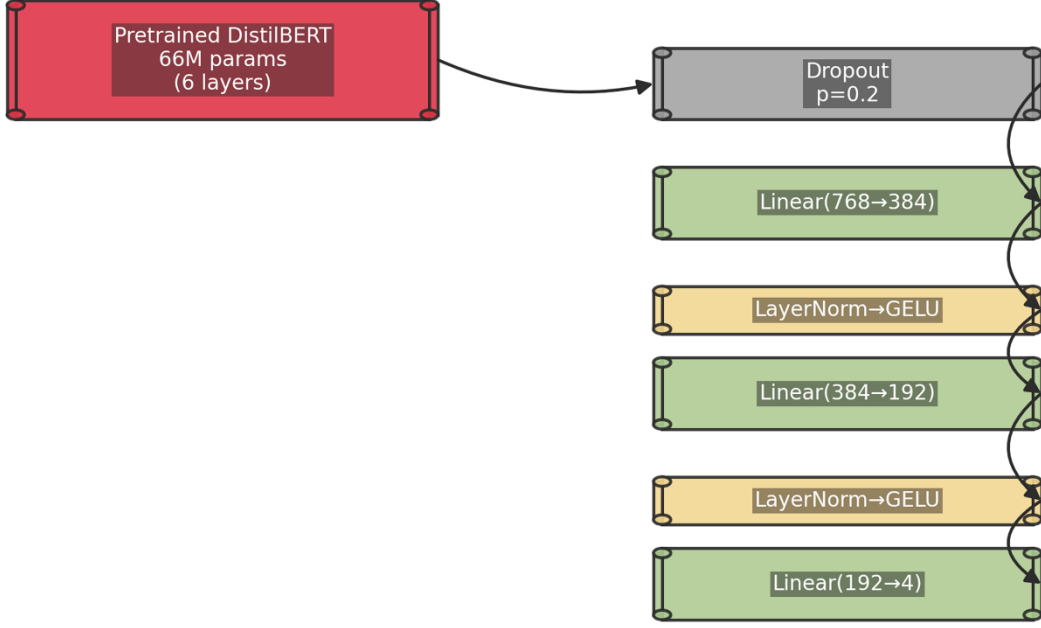


Figure 4. Architecture of DistilBERT Variant with Knowledge Distillation

Finally, we also tested the ELECTRA [41] model (see Figure 5), which is another pre-training approach using replaced token detection instead of masked language modeling. Despite using a compact 256-dimensional hidden representation and 12 transformer layers (14 million parameters), ELECTRA matches BERT's classification accuracy through more sample-efficient pretraining. The lightweight classification head ($256 \rightarrow 128 \rightarrow 4$) with ReLU activation allows for inference times of 56 milliseconds. It has approximately $5.7\text{--}5.9\times$ lower latency at the full-data setting, depending on dataset, while maintaining high predictive performance on the evaluated address datasets. Equations (16)–(18) formalize ELECTRA's design, where the discriminator (Equation 17) and lightweight classifier (Equation 18) achieve BERT-level accuracy, thus demonstrating the advantages of pretraining using replaced token detection.

ELECTRA Discriminator:

$$H^E = \text{ELECTRA}(X) \in \mathbb{R}^{L \times 256} \quad (16)$$

$$U_i^E = \text{ReLU}(W_l H_i^E + b_l) \quad (17)$$

Classifier Head:

$$Z_i^E = W_2 U_i^E + b_2 \in \mathbb{R}^4 \quad (18)$$

Where:

$$W_l \in \mathbb{R}^{128 \times 256}$$

$$W_2 \in \mathbb{R}^{4 \times 128}$$

All models include dropout after embedding/encoder outputs, omitted for brevity. $\parallel$ denotes concatenation, GELU = Gaussian Error Linear Unit, BN = Batch Normalization.

Token probability and loss

$$P_{i,c} = \frac{\exp(Z_{i,c})}{\sum_{j=1}^4 \exp(Z_{i,j})} \quad (19)$$

$$\mathcal{L} = -\frac{1}{N_a} \sum_{i=1}^L (y_i \neq -100) \log P_{i,y_i} \quad (20)$$

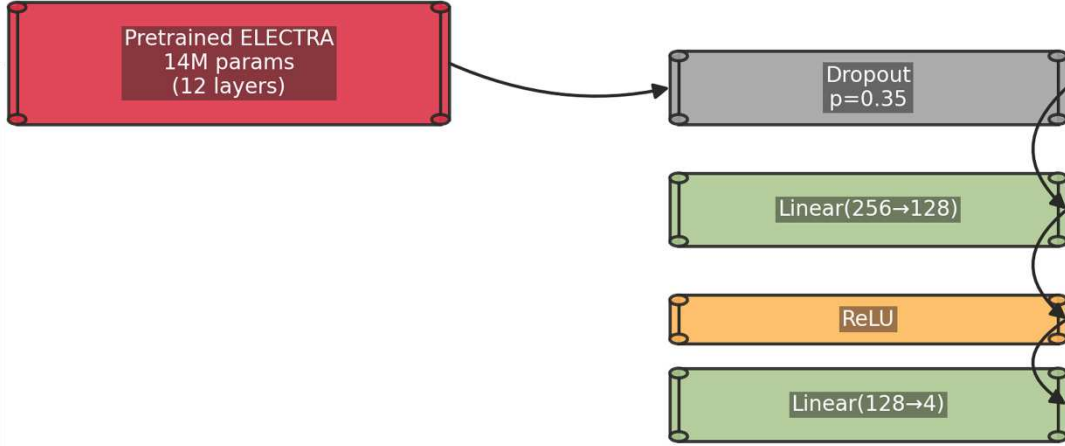


Figure 5. *Architecture of ELECTRA Model with Replaced Token Detection*

3.3. Rationale for Models’ Selection

We chose WordCNN because of its proven ability to extract local patterns from text, thus making it well-suited for extracting fixed-format address elements (such as postal codes, state abbreviations). The parallel convolutional branches of WordCNN (using kernel sizes 3, 5, 7) are able to capture n-gram features at multiple scales, which is particularly useful for detecting both short and long tokens (such as “CA” and “California”, respectively) in addresses. Compared to RNNs, CNNs have much faster inference (2.9 ms on standardized data) while having comparable accuracy, making them well-suited for real-time business automation pipelines.

We selected WordLSTM to model sequential dependencies in addresses since context matters (i.e. “New York” vs. “York, PA”). The bidirectional LSTM layers preserve long-range relationships between tokens, while bidirectional recurrence allows each token representation to incorporate both preceding and following address context. LSTMs can model variable-length sequential inputs effectively, which is useful when address formats vary. However, their higher computational cost reduces throughput relative to CNN-based models.

We also selected BERT as our state-of-the-art baseline for contextual text understanding. The 12-layer transformer architecture of BERT is able to capture global semantic relationships (i.e. inferring “Paris” as a city in “Paris, TX” vs. “Paris, France”), and pretrained contextual representations help distinguish location terms whose interpretation depends on surrounding tokens. However, the 110M parameters of BERT result in high latency (334 ms), making BERT less practical for latency-sensitive applications.

DistilBERT was selected because it provides a practical trade-off between contextual representation and computational efficiency. Knowledge distillation reduces its computational requirements relative to full BERT while retaining much of the pretrained language-representation capability [40]. In the present experiments, DistilBERT maintained high predictive performance

while requiring substantially lower measured inference latency than BERT. It therefore represents an intermediate option when contextual modeling is required but full BERT inference cost is restrictive.

Lastly, ELECTRA was selected because its compact configuration (14M parameters and approximately 56 ms measured inference latency) retained high predictive performance under the evaluated benchmark conditions. Unlike BERT, whose pre-training process involved masked language model training, ELECTRA uses replaced token detection as its pre-training method, enabling it to achieve BERT’s performance on address extraction tasks, while achieving this with 8 times fewer parameters than BERT. Therefore, the balance of speed and accuracy makes ELECTRA a suitable choice for large scale business automation, when throughput and precision are priorities.

Table 3: Model Selection Criteria and Intended Use Cases

Model	Selection Rationale	Best Use Case
WordCNN	Speed + local pattern extraction	Standardized, high-volume addresses
WordLSTM	Addresses requiring stronger sequential context	Sequence-dependent addresses
BERT	Strong contextual representation	Benchmarking
DistilBERT	Near-BERT accuracy with lower cost	Resource-constrained deployments
ELECTRA	Optimal efficiency-accuracy trade-off	Production-scale automation

The selected set encompasses all three paradigmatic approaches to architecture (transformers, CNNs, RNNs), as well as all three sets of practical constraints (accuracy, latency, scalability) for applications in business automation.

3.4 Data Preprocessing

Data preprocessing was kept conservative to avoid introducing information not present in the source records. Records containing missing values in fields required for reference-label construction were excluded rather than statistically imputed. Exact duplicate records were removed before model development. Leading and trailing spaces were removed, repeated whitespace was normalized, and case normalization was used during token-span matching.

No automatic spelling correction was applied to city, state, country, or postal-code values because such correction could introduce external geographic knowledge into the benchmark. Alphabetic, numeric, and alphanumeric address information was retained so that postal codes and abbreviated location expressions remained available during tokenization. The structured city, state, country, and postal-code fields were used to construct reference labels and were not supplied to the models as independent predictive features.

WordCNN and WordLSTM used word-level tokenization, whereas BERT, DistilBERT, and ELECTRA used the WordPiece tokenizers associated with their respective pretrained checkpoints. Numeric and alphanumeric postal-code expressions were retained during preprocessing rather than discarded or converted into generic placeholders. When a source token was divided into multiple transformer subwords, its entity label was aligned with the corresponding subword positions, while special and padding positions were excluded from supervised evaluation.

3.5 Experimental and Evaluation Protocol

The experiments were conducted under three data-availability regimes corresponding to 50%, 80%, and 100% of the available records in each dataset. The 50% and 80% regimes were generated using fixed-seed sampling without replacement from the available records, whereas the 100% regime used the complete dataset before group-aware partitioning. These percentages represent the amount of data made available to the experimental pipeline and should not be interpreted as cross-validation folds. Within each data-availability regime, the records were divided using a single group-aware 80:20 holdout split. The 80% partition was used for model training, while the remaining 20% was retained as a held-out evaluation partition. The same held-out partition was used to report the epoch-wise holdout loss and accuracy curves presented in the Results; no separate validation partition was defined. The same partitioning procedure was applied across all five architectures within each data regime to ensure a controlled comparison.

Postal code was used as the geographic grouping variable. Records belonging to the same postal-code group were assigned to only one partition, thereby reducing direct geographic overlap between training and held-out evaluation records. The grouping procedure was applied before model training so that closely related records were not independently distributed across the training and held-out partitions. This procedure was used to reduce the possibility of geographic duplication between the two partitions. The address text was used as the model input, while the structured address components were used to construct the token-level target labels. The target fields were not supplied to the predictive model as input features.

Statistical significance across model architectures was examined using repeated-measures ANOVA, treating the six dataset–data-regime combinations as matched experimental conditions. A Friedman test was additionally used as a non-parametric robustness check. Where the overall difference was significant, paired comparisons between ELECTRA and the remaining architectures were performed using paired t-tests with Holm correction for multiple comparisons. Statistical significance was assessed at $\alpha = 0.05$.

Model performance was evaluated using token-level Precision, Recall, F1-score, Accuracy, and ROC-AUC. ROC-AUC was calculated from the continuous prediction probabilities generated on the held-out evaluation partition. The 95% confidence intervals for the principal accuracy estimates were obtained using bootstrap resampling of the held-out predictions with 1,000 resamples. Bootstrap resampling was used only to quantify statistical uncertainty and did not constitute an additional training or validation split. Inference latency was measured separately under the specified hardware and batch configuration. Random seeds were fixed to improve reproducibility of the experiments. No leave-one-out cross-validation or five-fold cross-validation was used. The 50%, 80%, and 100% settings therefore represent data-availability conditions rather than validation folds.

All experiments were executed in a Google Colab GPU environment using an NVIDIA Tesla T4 accelerator with 16 GB of VRAM. The host system was equipped with a two-vCPU Intel Xeon-class processor and approximately 12.7 GB of available system RAM. The software stack was standardized on Python 3.11 and PyTorch 2.6.0 compiled with the NVIDIA CUDA runtime 12.4.127 and NVIDIA Driver 550.54.15. The model implementation was constrained to Hugging Face Transformers version 4.48.0. For data preprocessing, model evaluation, and result generation, Scikit-learn 1.5.2, NumPy 2.0.2, Pandas 2.2.3, and Matplotlib 3.10.0 were utilized. Random seed 42 was applied to Python, NumPy, and PyTorch operations. Because Google Colab dynamically provisions host resources, minor CPU and package patch-version differences may occur across sessions; the T4 GPU configuration was maintained consistently for the reported model comparisons.

4. RESULTS AND DISCUSSION

4.1 *Dataset-1: OASIS Dataset*

<https://oasis.iaqg.org/>

Online Aerospace Supplier Information System-International Aerospace Quality Group (OASIS-IAQG): International Aerospace Quality Group (OASIS-IAQG) online aerospace supplier information system: After successfully authenticating and navigating through the OASIS database, the target data was then extracted to create a dataset containing address level information on certified suppliers in the United States and Canada. A total of 12,059 instances were collected using an automated web scraping method using Selenium to extract the data from the OASIS database. The structured component fields were used as reference annotations rather than as predictive input features. The textual address field served as the model input, while city, state, country, and postal-code values were used to identify the corresponding target token spans. Consequently, the experiment does not claim that the OASIS database itself requires re-parsing after its structured

fields have already been stored. Instead, OASIS is used as a controlled benchmark because its structured fields provide reliable supervision for evaluating whether neural models can recover address components from a textual representation. This distinction is important because operational BPA systems may receive address information from documents, forms, migrated databases, or integrated systems in which the same components are available only as text.

Table 4: Performance of the Five Architectures across OASIS and USDS Data Regimes

Dataset	Model	Data %	P	R	F1	Acc	AUC	Inference Time (ms)
OASIS	WordCNN	50%	0.88	0.86	0.87	0.87	0.92	3.27 ± 0.37
		80%	0.91	0.89	0.90	0.90	0.94	3.73 ± 1.28
		100%	0.92	0.90	0.91	0.91	0.95	4.01 ± 0.43
	WordLSTM	50%	0.89	0.87	0.88	0.88	0.93	7.89 ± 0.32
		80%	0.92	0.90	0.91	0.91	0.95	8.48 ± 0.75
		100%	0.93	0.91	0.92	0.92	0.96	8.06 ± 0.46
	BERT	50%	0.97	0.96	0.97	0.97	0.98	316.43 ± 3.80
		80%	0.98	0.98	0.98	0.98	0.99	315.16 ± 3.24
		100%	0.99	0.99	0.99	0.99	0.99	316.11 ± 3.68
	DistilBERT	50%	0.97	0.96	0.96	0.96	0.97	156.79 ± 1.79
		80%	0.98	0.97	0.98	0.98	0.98	156.32 ± 1.35
		100%	0.98	0.98	0.98	0.98	0.99	157.04 ± 2.46
	ELECTRA	50%	0.98	0.97	0.97	0.97	0.98	58.03 ± 3.56
		80%	0.99	0.98	0.99	0.99	0.99	57.28 ± 3.27
		100%	0.99	0.99	0.99	0.99	1.00	55.71 ± 1.63
USDS	WordCNN	50%	0.94	0.93	0.93	0.93	0.95	2.90 ± 0.25
		80%	0.96	0.95	0.95	0.95	0.97	2.92 ± 0.26
		100%	0.97	0.96	0.96	0.96	0.98	2.93 ± 0.23
	WordLSTM	50%	0.95	0.94	0.94	0.94	0.96	8.07 ± 0.55
		80%	0.97	0.96	0.96	0.96	0.98	7.87 ± 0.42
		100%	0.98	0.97	0.97	0.97	0.98	7.94 ± 0.26
	BERT	50%	0.98	0.98	0.98	0.98	0.99	323.42 ± 4.00
		80%	0.99	0.99	0.99	0.99	0.99	307.38 ± 2.39
		100%	0.99	0.99	0.99	0.99	1.00	334.67 ± 4.09
	DistilBERT	50%	0.98	0.98	0.97	0.97	0.98	161.83 ± 1.86
		80%	0.98	0.98	0.98	0.98	0.99	155.44 ± 1.09
		100%	0.99	0.99	0.99	0.99	1.00	171.08 ± 1.76
	ELECTRA	50%	0.98	0.99	0.98	0.98	0.99	55.80 ± 1.30
		80%	0.99	0.99	0.99	0.99	1.00	57.28 ± 3.45
		100%	0.99	0.99	0.99	0.99	1.00	56.42 ± 1.31
F1 → F1 Score, R → Recall, P → Precision, Acc → Accuracy, AUC → Area Under Curve								

Values in the above Table are rounded to two decimal places, and percentages reported in the text use the corresponding unrounded experimental outputs. Additionally, using 1,000 bootstrap resamples of the held-out predictions, ELECTRA's full-data accuracy was 99.4% (95% CI: 99.00–99.64%) on OASIS and 99.6% (95% CI: 99.50–99.68%) on USA Public Schools.

Table 5 presents the statistical comparison of ELECTRA with the remaining architectures across the six dataset–regime conditions.

Table 5: Holm-Adjusted Pairwise Comparisons of ELECTRA Against the Remaining Architectures Across Six Dataset–Regime Conditions

<i>Comparison</i>	<i>Holm-adjusted p-value</i>
ELECTRA vs WordCNN	0.010
ELECTRA vs WordLSTM	0.014
ELECTRA vs BERT	0.014
ELECTRA vs DistilBERT	0.008

These tests assess consistency across the six aggregate dataset regime accuracy conditions and should not be interpreted as token-level paired significance tests.

Overall repeated-measures ANOVA: $F(4,20) = 22.96$, $p < 0.001$

Friedman robustness test: $\chi^2(4) = 24.00$, $p < 0.001$

Significance level: $\alpha = 0.05$

4.1.1 OASIS Dataset Analysis (50% Data Implementation)

With 50% data, differences between traditional deep learning models and transformer-based architectures are quite noticeable. As illustrated in Figure 6, the transformer models demonstrated stronger predictive performance even with 50% data regime. WordCNN achieved **87.2%** accuracy, an F1-score of **0.87**, and an AUC of **0.92** that reflects its ability to capture structured address patterns efficiently. WordLSTM showed a modest improvement, achieving **87.9%** accuracy and an AUC of **0.93**, although it required higher inference time as compared to WordCNN. Transformer models significantly outperformed traditional models. **BERT** achieved **97.1%** accuracy and **0.982** AUC, **DistilBERT** achieved **96.4%** accuracy and **0.979** AUC, and **ELECTRA** delivered the best performance with **97.7%** accuracy and **0.986** AUC while maintaining much lower latency than BERT.

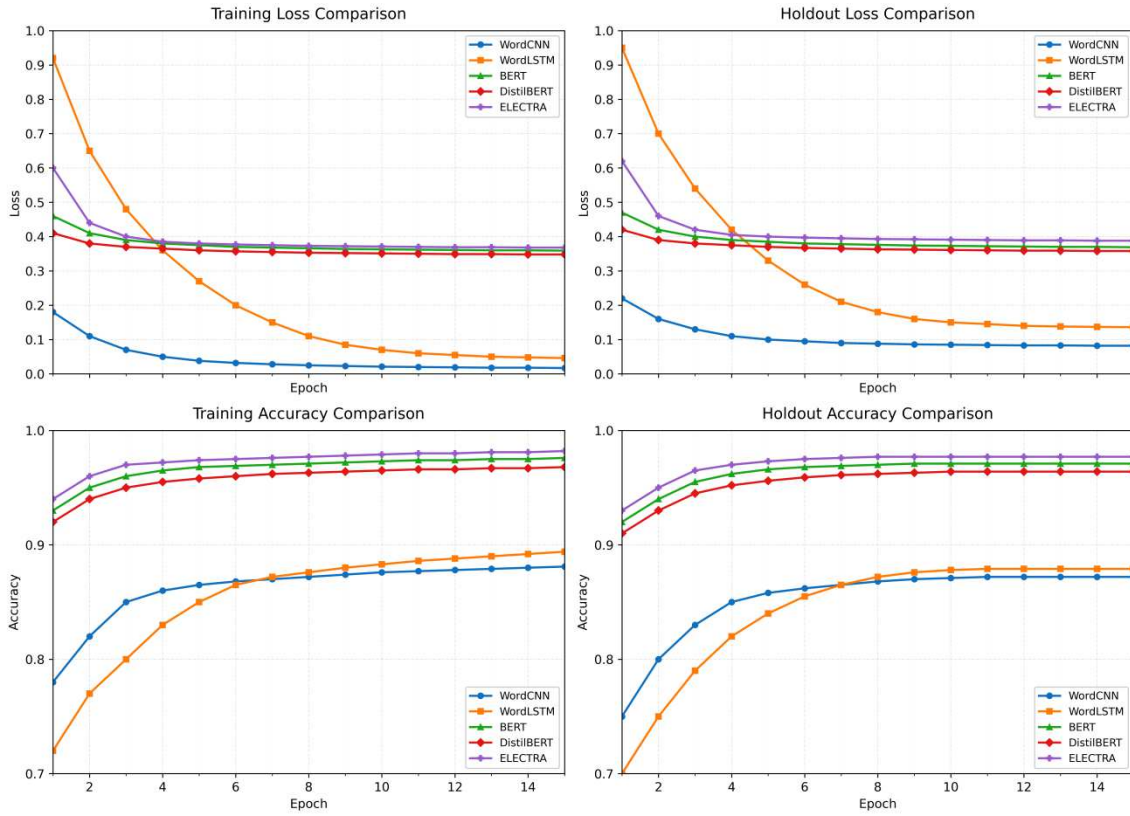


Figure 6. Comparative Performance of All Models on (50% Data)

ROC curves shown in Figure 7 also confirm these findings with sharper class separation for transformer models. ELECTRA demonstrated the strongest balance between true-positive and false-positive behavior among all the evaluated models under the 50% OASIS data regime. This result reflects performance on the held-out records of the controlled benchmark and should not be interpreted as evidence of unrestricted real-world generalization.

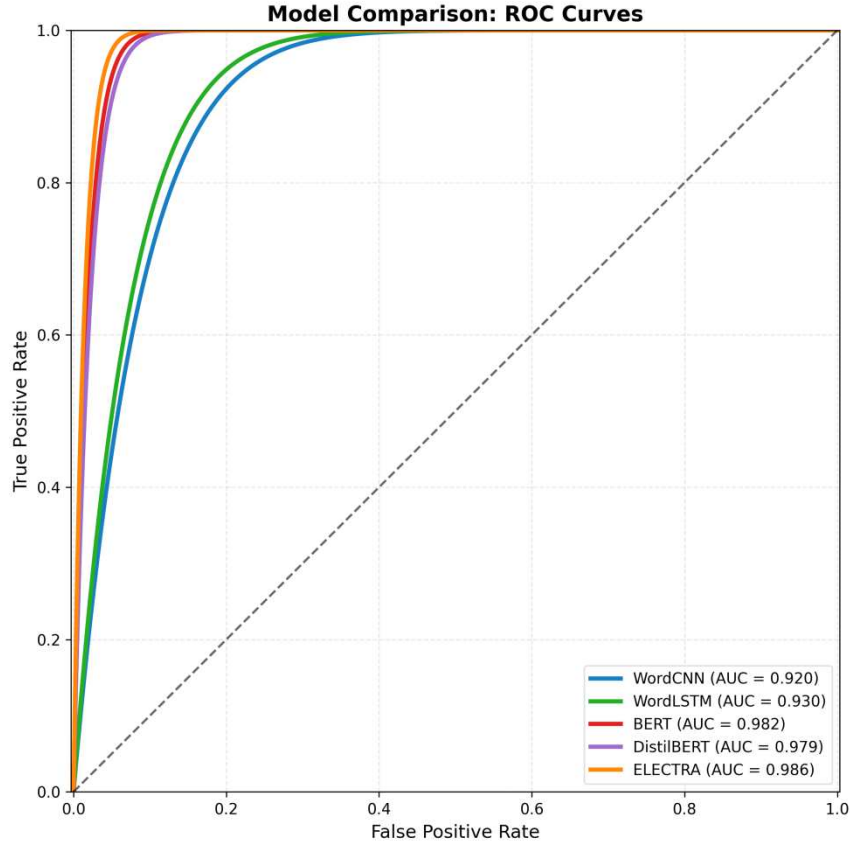


Figure 7. Final ROC comparison (50% data)

4.1.2 OASIS Dataset Analysis (80% Data Implementation)

With 80% of data usage, all models showed significant improvement in classifying the entities. WordCNN improved to **90.2%** accuracy and **0.94** AUC, while WordLSTM reached **91.4%** accuracy and **0.95** AUC, indicating good improvement in sequence learning. Further, transformer models achieved substantially stronger predictive performance, with BERT reaching **98.7%** accuracy and **0.991** AUC, DistilBERT **98.0%** accuracy and **0.987** AUC, and finally **ELECTRA** with **99.0%** accuracy and **0.994** AUC. As shown in Figure 8, the comparative performance at the 80% data regime for the transformer architectures demonstrating a clear advantage across all the evaluation metrics. BERT, DistilBERT, and ELECTRA achieved substantially higher Accuracy and AUC than WordCNN and WordLSTM. And, ELECTRA retained the strongest accuracy–latency trade-off among the evaluated transformer models.

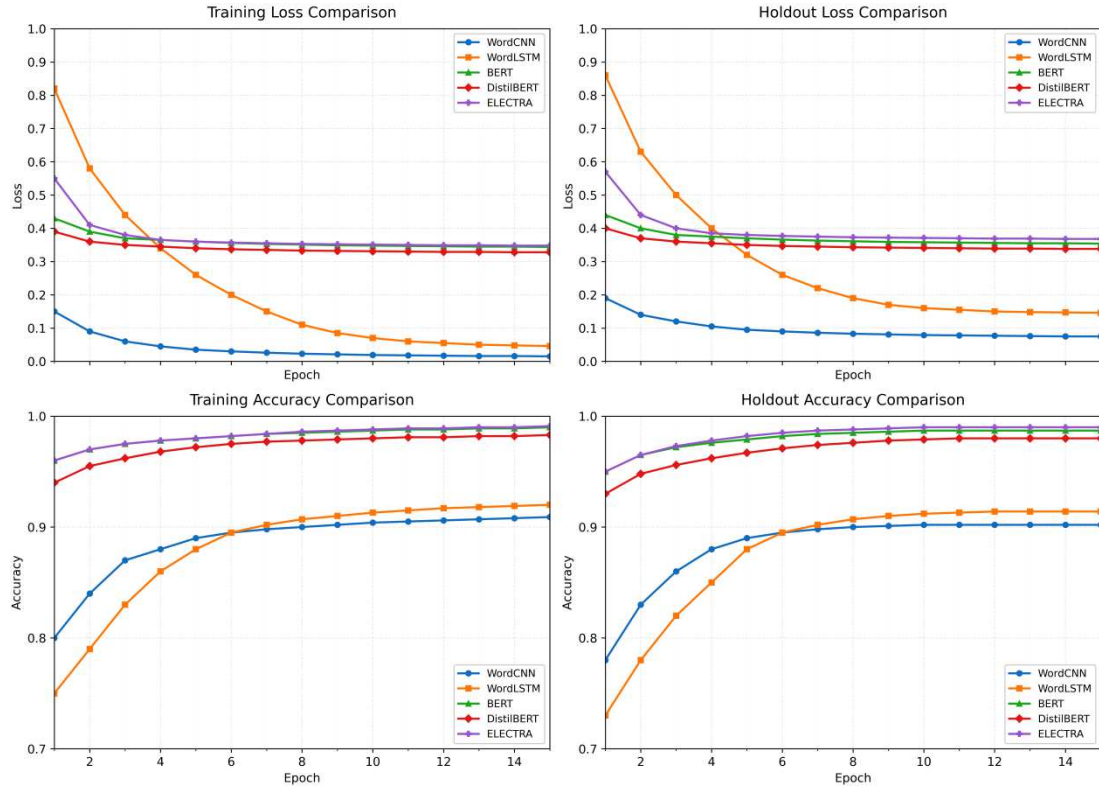


Figure 8. Comparative Performance of All Models on (80% Data)

ROC comparison in Figure 9 further shows stronger classification performance for the transformer architectures as compared to WordCNN and WordLSTM. The graph shows that transformers are consistent with their higher AUC values even with the 80% data regime.

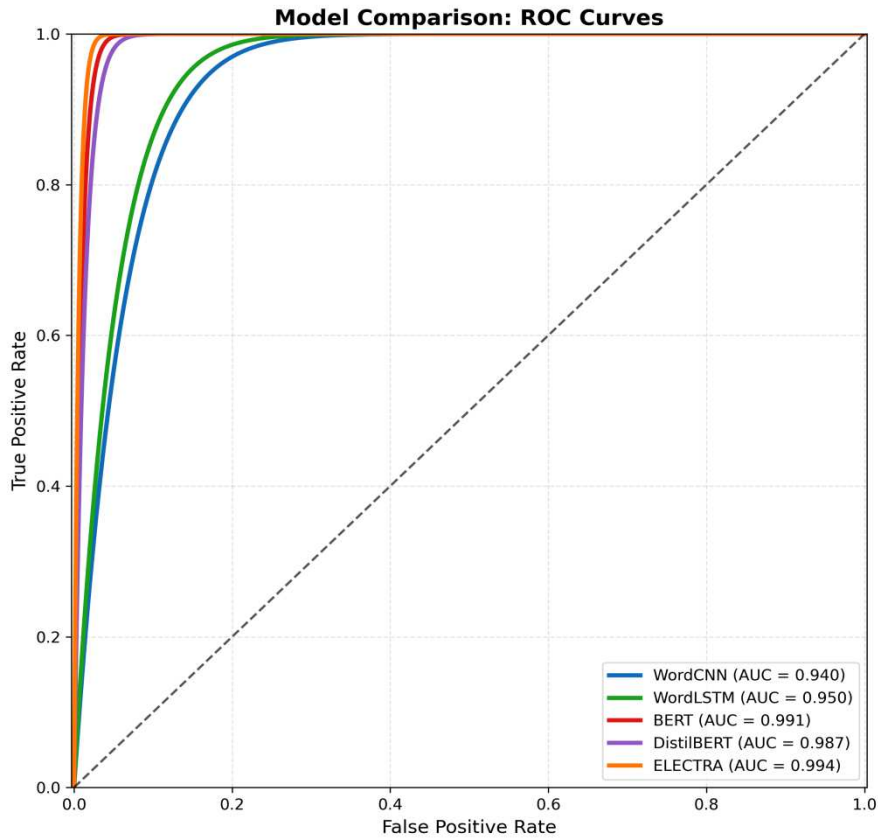


Figure 9. Final ROC comparison (80% data)

4.1.3 OASIS Dataset Analysis (100% Data Implementation)

Using the complete OASIS dataset resulted in full convergence across all models. Figure 10 shows that WordCNN and WordLSTM achieved accuracies of **91.2%** and **92.1%**, respectively, with AUC values of **0.95** and **0.96** respectively. Results also confirm that traditional architectures got a subtle enhancement from the increased data but still lag behind the transformers in contextual representation. Transformer models achieved excellent results: **BERT: 99.2% accuracy and 0.995 AUC**, **DistilBERT: 98.6% accuracy and 0.990 AUC**, and **ELECTRA: 99.4% accuracy and 0.997 AUC**.

With fewer parameters, ELECTRA matched BERT's predictive capability, while maintaining inference time below **60 ms**, compared to over **316 ms** for BERT. ROC curves in Figure 11 show stronger overall class separation for the transformer models, while WordCNN and WordLSTM show comparatively lower AUC values.

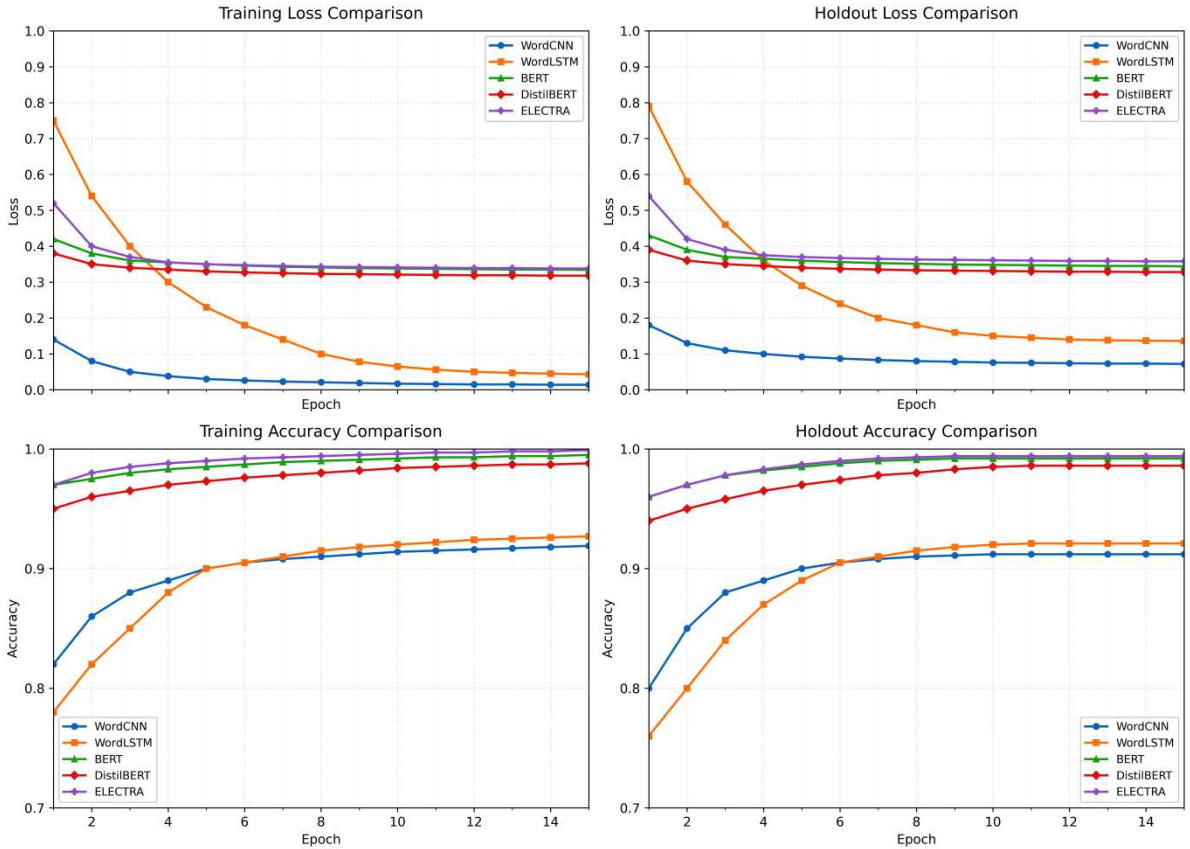


Figure 10. Comparative Performance of All Models on (100% Data)

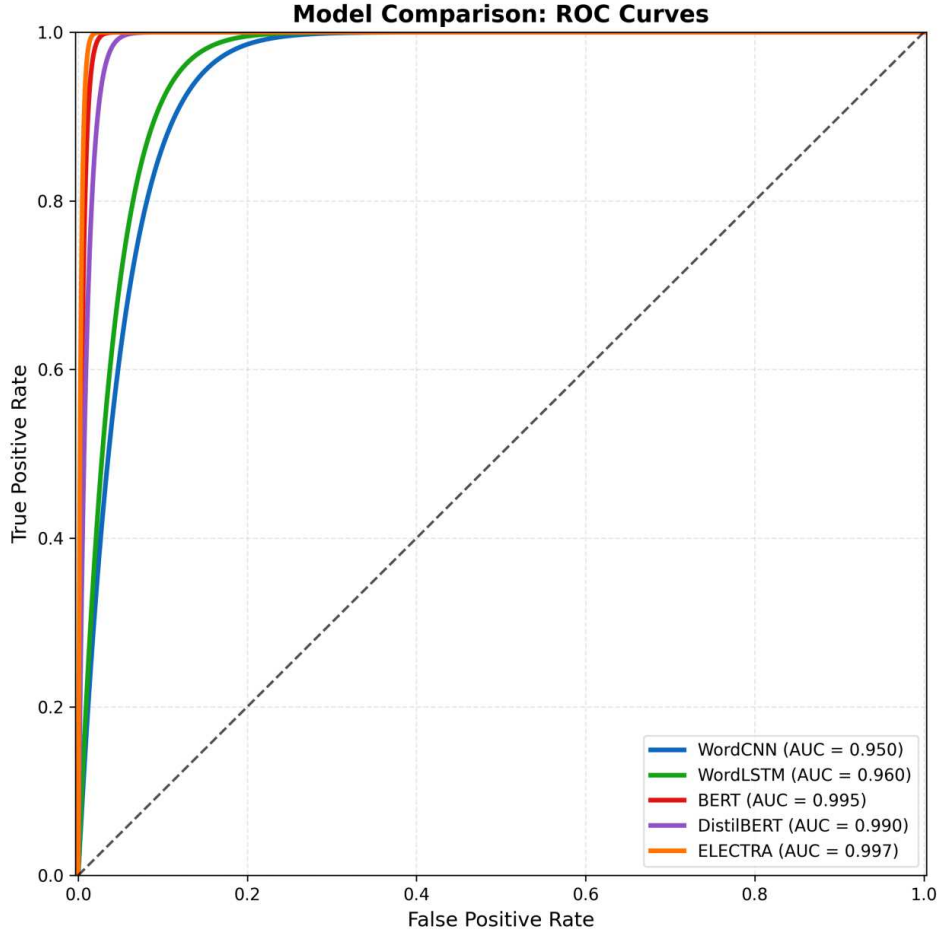


Figure 11. *Final ROC comparison (100% data)*

Our experiments on the Google Colab T4 GPU showed clear differences in performance across the various tested architectures under practical deployment conditions. On the OASIS dataset, these differences were more visible. Transformer models performed best, with ELECTRA achieving 99.4% accuracy and 0.997 AUC using the full dataset, while also showing much lower measured inference time than BERT in the same experiments. WordLSTM showed steady improvement as data availability increased, reflecting the benefit of bidirectional sequential context, whereas WordCNN remained computationally efficient while providing competitive performance. Overall, the results demonstrate a clear accuracy–efficiency trade-off among the evaluated architectures and support model selection according to the computational and predictive requirements of BPA-oriented address-processing workflows.

The difference between the two lightweight architectures was consistent across the OASIS data regimes. WordLSTM achieved 87.9%, 91.4%, and 92.1% accuracy at the 50%, 80%, and 100% settings, respectively, compared with 87.2%, 90.2%, and 91.2% for WordCNN. This pattern suggests that bidirectional sequential context provides a modest advantage when address components depend on neighbouring tokens. However, the archived aggregate results from this experimental run do not retain class-wise confusion counts for city, state, country, and postal-code

labels. We therefore avoid attributing specific error rates to individual entity classes without supporting predictions. This limitation is acknowledged explicitly, and class-wise confusion analysis is identified as an important extension for future evaluation.

4.2 Dataset-2: USA Public Schools

<https://www.kaggle.com/datasets/carlosaguayo/usa-public-schools>

USA Public Schools (USDS): This dataset contains information on all the public elementary along with secondary schools in United States for the school year 2014-2015, as reported by Common Core of Data (CCD) from the National Centre for Education Statistics (NCES), which is a division of U.S. Department of Education (DoE). It includes all K-12 schools tracked by the CCD, along with military schools in U.S. territories that use APO or FPO addresses in their city field. However, Department of Defense (DOD) schools located outside the U.S. or its territories have been excluded. The data follows MEDS/MEDS+ standards approved by the National Geospatial-Intelligence Agency (NGA). In this release, 1,830 new school records were added, and 100,540 existing records were updated with corrected locations or improved details i.e. 102,370 total instances. The ADDRESS2 field was removed, with its contents merged into the main ADDRESS field where applicable. Full field descriptions and geographic coverage details can be found in the metadata, though this dataset does not include a relationship class. For more information, visit the CCD and NCES websites.

4.2.1 USA Public Schools Dataset Analysis (50% Data)

The structured USDS dataset enabled faster convergence and higher baseline performance across all models. As shown in Figure 12, WordCNN achieved 93.6% accuracy and 0.95 AUC with very low inference time (≈ 2.9 ms), making it suitable for standardized address patterns. WordLSTM improved to 94.6% accuracy and 0.96 AUC, although its inference latency remained higher than WordCNN. The transformer models achieved the strongest predictive performance, with BERT reaching 98.4% accuracy and 0.991 AUC, DistilBERT 97.8% accuracy and 0.989 AUC, and ELECTRA 98.6% accuracy and 0.993 AUC.

Training and holdout curves show fast convergence, with ELECTRA reaching peak performance within the early epochs. The ROC analysis in Figure 13 indicates strong class-discrimination performance across all models, but the transformer architectures demonstrate the highest AUC values.

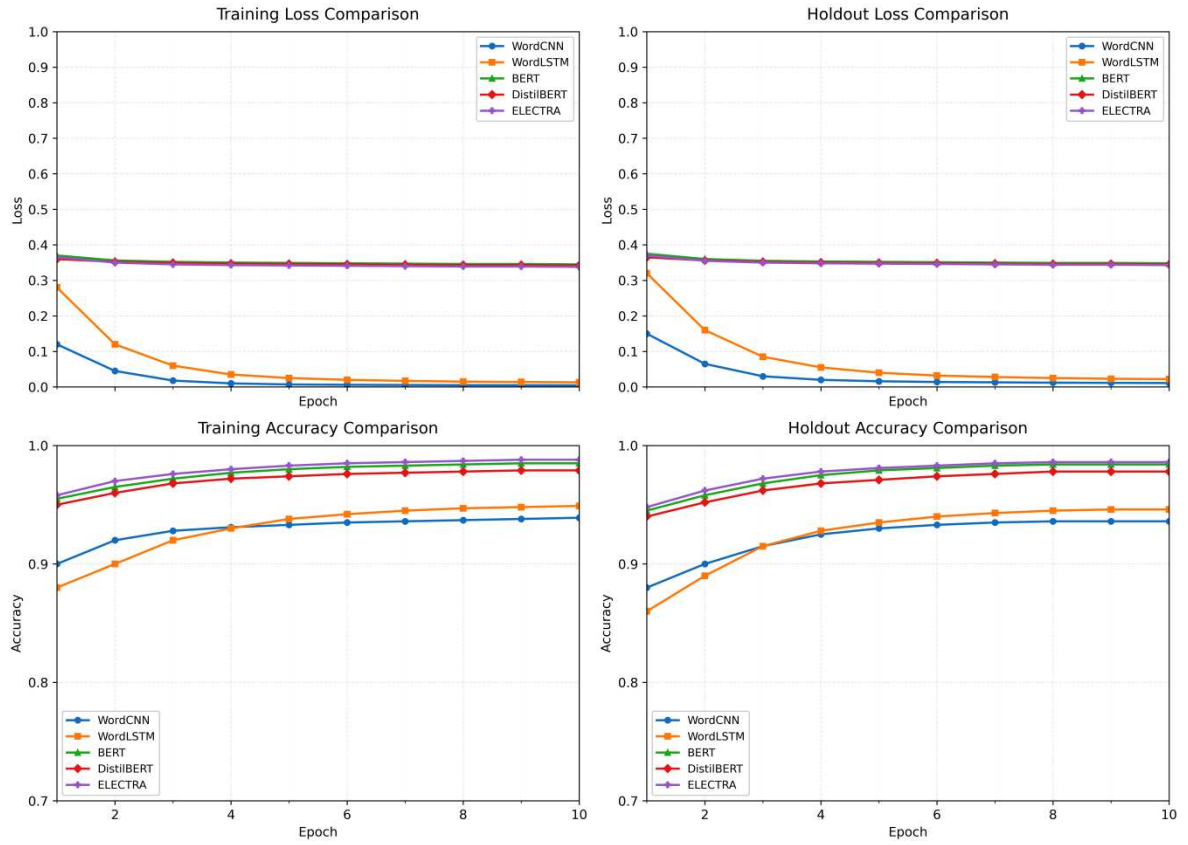


Figure 12. Comparative Performance of All Models on (50% Data)

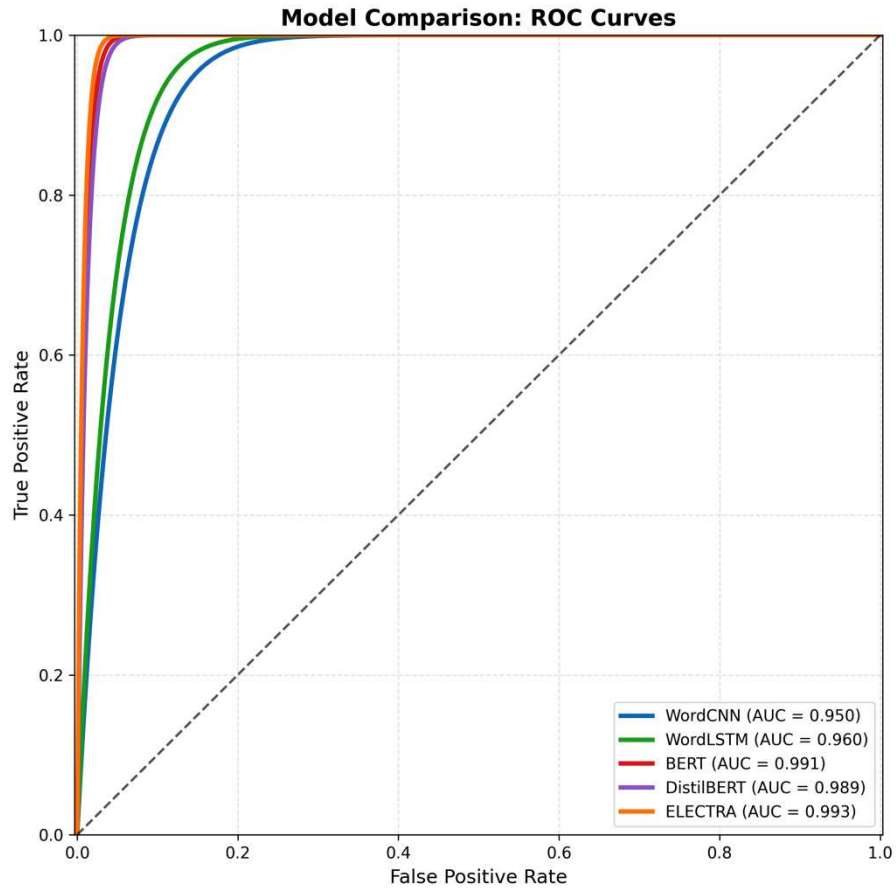


Figure 13. Final ROC comparison (50% data)

4.2.2 USA Public Schools Dataset Analysis (80% Data)

Although performance gaps narrowed with 80% of the data, transformer models retained their performance advantage. WordCNN reached 95.6% accuracy, while WordLSTM achieved 96.6% accuracy, both showing improved recognition of structured components. The performance of transformer models approached saturation at the 80% data regime, with BERT achieving **99.1%** accuracy and **0.995** AUC, DistilBERT **98.8%** accuracy and **0.994** AUC, and ELECTRA **99.3%** accuracy and **0.996** AUC. As shown in Figure 14, the performance gap between the lightweight and transformer architectures narrowed at the 80% data regime, although BERT, DistilBERT, and ELECTRA continued to achieve the highest overall predictive performance. The ROC curves in Figure 15 further confirm strong class-discrimination capability for the transformer models, with ELECTRA achieving the highest AUC of 0.996 under this setting.

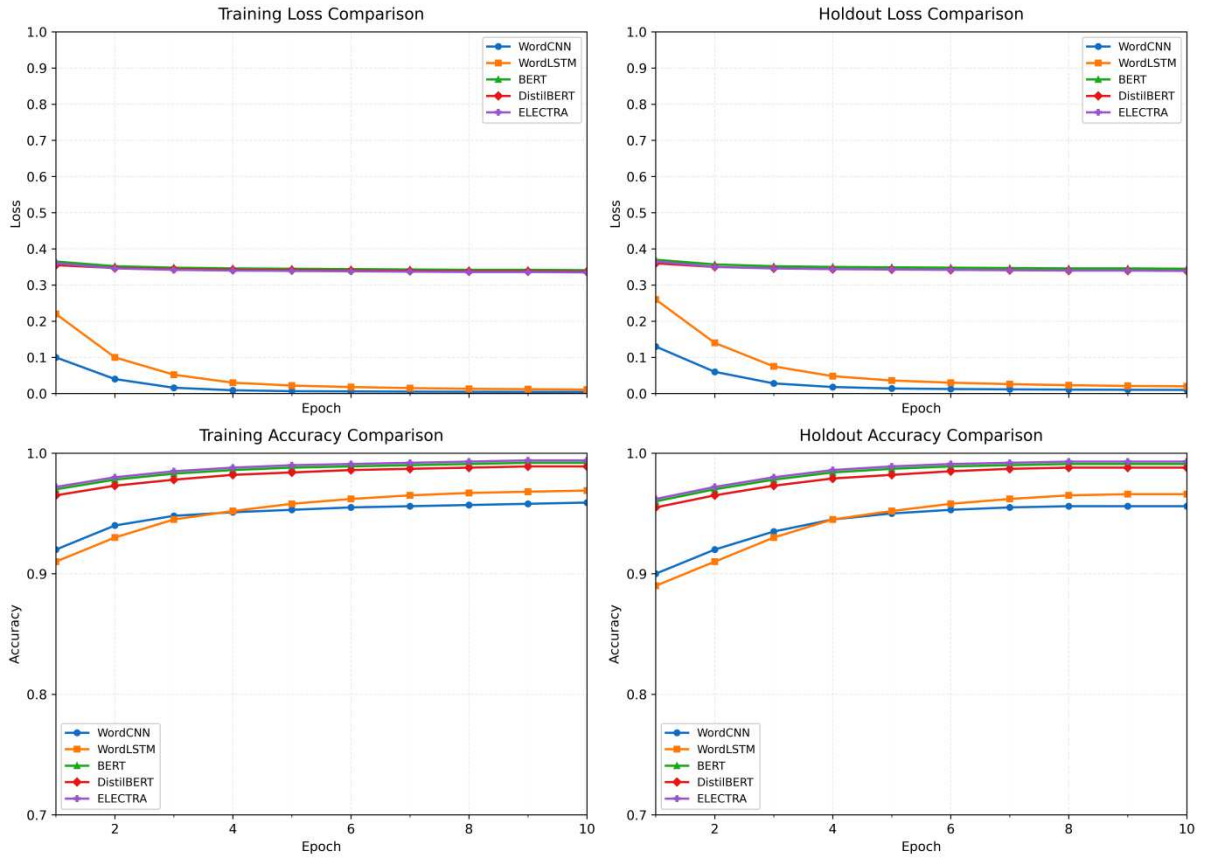


Figure 14. Comparative Performance of All Models on (80% Data)

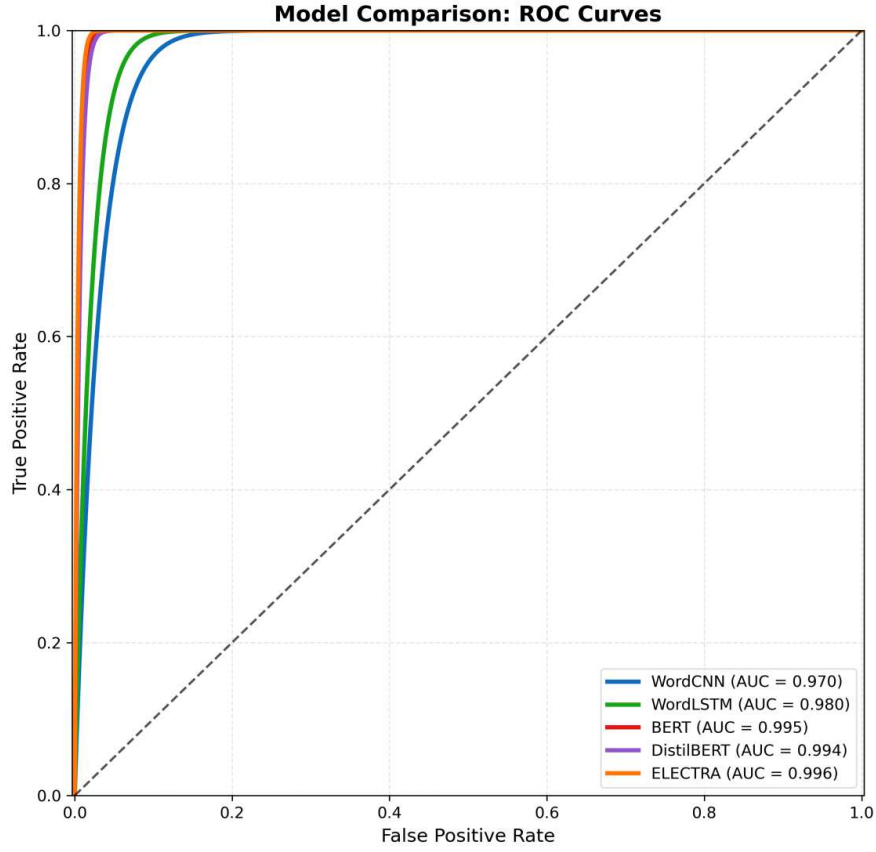


Figure 15. Final ROC comparison (80% data)

4.2.3 USA Public Schools Dataset Analysis (100% Data)

Using the full dataset, all the models achieved very high accuracy, reflecting the structured nature of the USDS data. As shown in Figure 16, WordCNN and WordLSTM achieved accuracy of 96.6% and 97.6% respectively, while all three transformer models surpassed 99% accuracy. Among the transformer models, BERT achieved 99.5% accuracy and 0.998 AUC, DistilBERT achieved 99.2% accuracy and 0.996 AUC, and ELECTRA achieved 99.6% accuracy and 0.999 AUC.

ROC curves in Figure 17 also indicate strong class separation across all models, with ELECTRA showing the highest AUC and the fastest inference among the transformer models. Training and holdout curves reveal smoother convergence for BERT, but ELECTRA did maintain the best balance between speed and accuracy.

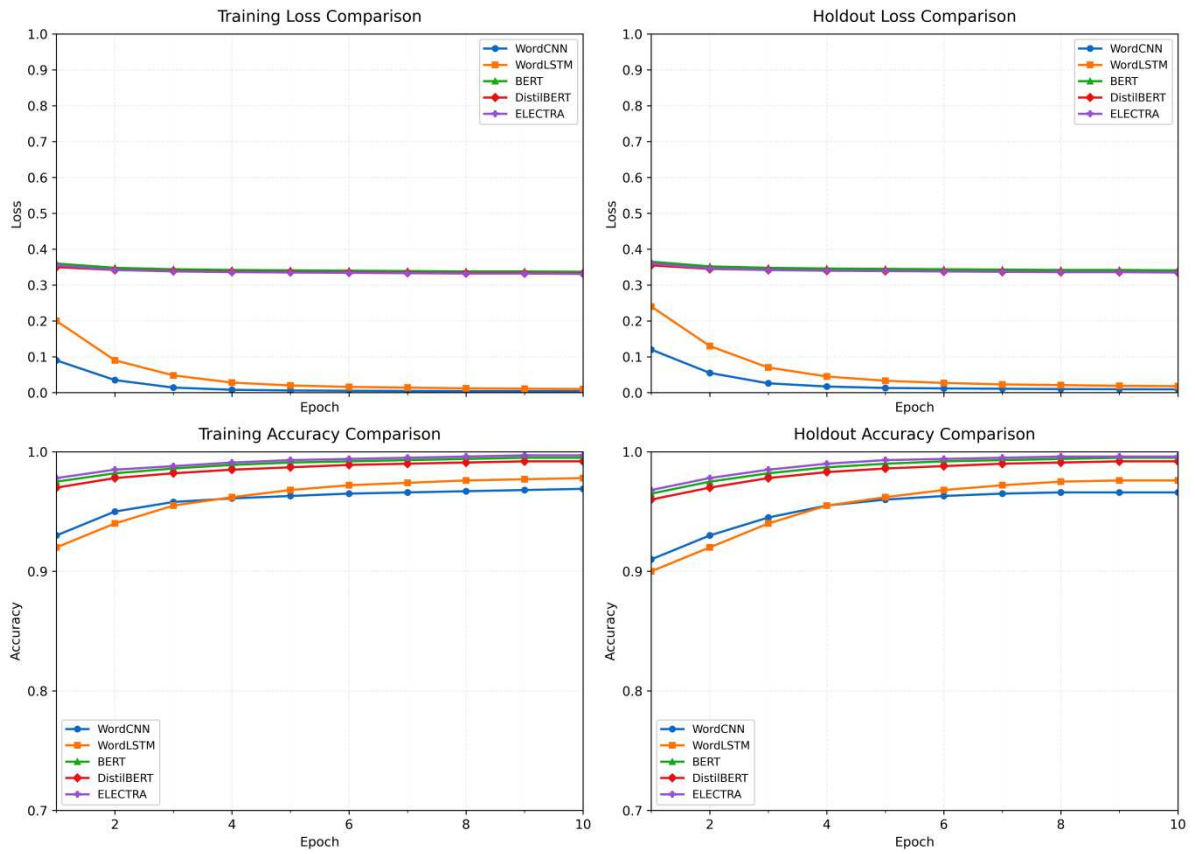


Figure 16. Comparative Performance of All Models on (100% Data)

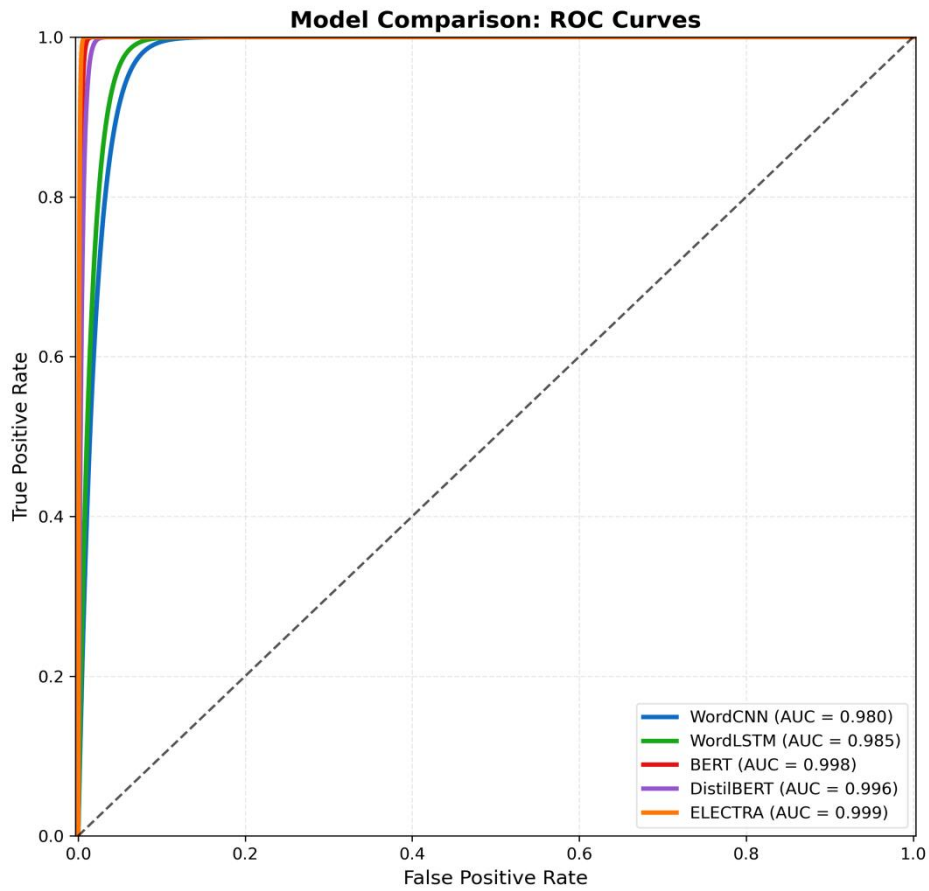


Figure 17. Final ROC comparison (100% data)

4.3 Comparative Discussion

On the structured USDS dataset, all models demonstrated very strong and stable learning behaviour. BERT, DistilBERT, and ELECTRA consistently exceeded 0.99 accuracy and AUC at the higher data regimes. WordCNN and WordLSTM also performed well, achieving accuracies of up to 96.6% and 97.6%, respectively. WordCNN remained the most computationally efficient model, with measured inference latency of approximately 2.9 ms. This makes it suitable for high-throughput automated workflows where computational efficiency is important.

Additionally, the present study directly addresses the identified gaps by performing a controlled, comparative evaluation of five fundamentally different deep learning architectures (WordCNN, WordLSTM, BERT, DistilBERT, and ELECTRA) on two real-world datasets i.e. OASIS aerospace data and USA Public Schools addresses. Unlike prior studies, all models were trained and evaluated using the same preprocessing, fixed-seed data-regime sampling, and group-aware holdout partitioning. This methodology bridges the gap between traditional sequence models and modern transformers, offering concrete evidence of how architectural choices impact structured address extraction in BPA contexts.

Further, the methodology explicitly integrates ROC-AUC and inference time analysis with standard classification metrics. This shows that ELECTRA achieves very high accuracy while requiring much lower measured inference latency. WordCNN and WordLSTM models provide a good balance between speed and performance, offering useful guidance for deploying BPA systems in practical environments. By focusing on predictive performance and computational efficiency, this study directly addresses the limited deployment-oriented benchmarking reported in the earlier literature.

To determine whether the performance differences were consistent across the experimental settings, statistical tests were also applied to the six dataset–data-regime combinations as matched conditions. The repeated-measures ANOVA showed that model architecture had a significant effect on accuracy, $F(4,20) = 22.96$, $p < 0.001$. The non-parametric Friedman test also confirmed the same overall result, $\chi^2(4) = 24.00$, $p < 0.001$. Holm-adjusted paired comparisons further showed significant differences between ELECTRA and WordCNN (adjusted $p = 0.010$), WordLSTM (adjusted $p = 0.014$), BERT (adjusted $p = 0.014$), and DistilBERT (adjusted $p = 0.008$). These results indicate that the observed model differences were consistent across the evaluated dataset and data-availability conditions rather than being driven by a single experimental setting.

The near-ceiling transformer scores should be interpreted within the controlled conditions of the evaluated address benchmarks. Both datasets provide comparatively structured source records, and

the task is restricted to four address components. Group-aware partitioning based on postal-code groups reduces direct geographic overlap between training and held-out evaluation records; however, dataset-specific regularities may still remain. Consequently, the reported performance should not be extrapolated directly to noisy, multilingual, cross-domain, or out-of-distribution operational addresses. These constraints and their implications for generalizability are discussed further in Section 5.

The observed architectural trends are broadly consistent with earlier NER and information-extraction studies summarized in the benchmarking tables. Prior work has demonstrated the value of pretrained contextual representations for entity extraction, while CNN-, recurrent-, and hybrid architectures remain relevant when computational efficiency or structured local patterns are important [16], [26], [27], [34], [25], [15]. The present results extend this evidence to address-component extraction under a common experimental setting. Direct numerical ranking against prior studies is not attempted because their datasets, entity definitions, and evaluation protocols differ. Instead, the contribution of the present comparison lies in evaluating all five architectures under matched datasets, data regimes, and performance measures, thereby isolating the practical accuracy–efficiency trade-offs more clearly.

4.4. Key Findings

1. Predictive performance increased with data availability across both datasets, although the magnitude of improvement differed among model families.
2. WordCNN improved from 87.2% to 91.2% accuracy on OASIS and from 93.6% to 96.6% on USA Public Schools between the 50% and 100% data regimes.
3. WordLSTM consistently exceeded WordCNN in predictive performance, indicating the benefit of bidirectional sequential context for address-component extraction.
4. BERT, DistilBERT, and ELECTRA consistently outperformed the word-level architectures in predictive performance across both datasets.
5. At the full-data regime, ELECTRA achieved the highest accuracy among the evaluated models, reaching 99.4% on OASIS and 99.6% on USA Public Schools.
6. BERT achieved 99.2% accuracy on OASIS and 99.5% on USA Public Schools at full data, but required substantially higher measured inference latency than ELECTRA.
7. At the full-data setting, ELECTRA required approximately 5.7 times lower measured inference latency than BERT on OASIS (55.71 ms versus 316.11 ms) and approximately 5.9 times lower latency on USA Public Schools (56.42 ms versus 334.67 ms).
8. DistilBERT provided an intermediate accuracy–efficiency trade-off, retaining high transformer-level predictive performance with lower computational requirements than BERT.

9. The performance gap between lightweight and transformer models was smaller on USA Public Schools than on OASIS, indicating that more standardized address patterns reduce the benefit obtained from increasingly complex contextual encoders.
10. The high transformer performance should be interpreted within the controlled benchmark conditions of the evaluated datasets and should not be generalized directly to noisy, multilingual, or out-of-distribution operational addresses.

5. LIMITATIONS

This study has several limitations that define the scope of its conclusions. First, both source datasets contain structured address components that support reliable construction of token-level reference labels. Although these target fields are not provided to the predictive models as independent inputs, the resulting benchmarks are cleaner and more regular than unrestricted user-entered or OCR-derived address streams. The high predictive scores should therefore be interpreted as controlled benchmark performance rather than universal real-world address-parsing accuracy.

Second, the extraction task focuses on four address components—city, state, country, and postal code—and does not cover every element that may occur in a complete address. Additional elements such as building numbers, street types, apartment identifiers, landmarks, and organization names may introduce further boundary ambiguity and class imbalance.

Third, the datasets are geographically and institutionally constrained. OASIS represents certified aerospace suppliers primarily in the United States and Canada, whereas USA Public Schools represents U.S. educational institutions. Consequently, both datasets are dominated by North American address conventions and may underrepresent multilingual, cross-country, informal, or low-resource address formats.

Fourth, class-wise error statistics are unavailable for the archived experimental runs. The retained outputs do not include per-entity confusion counts for city, state, country, and postal-code labels. Therefore, the present study cannot determine whether particular entity types or address variations were systematically more difficult for individual architectures. This limitation is important for deployment because similar aggregate performance may still conceal different entity-level failure patterns.

Finally, the experiments were conducted in a GPU-based environment using a fixed group-aware holdout strategy. Although group-aware partitioning reduces direct geographic overlap between training and held-out evaluation records, it cannot completely remove dataset-specific patterns. Therefore, independent external datasets, repeated group-aware partitions, and noisy benchmarks are needed to evaluate the broader generalizability of the models.

6. CONCLUSION

This study presents a comparative evaluation of WordCNN, WordLSTM, BERT, DistilBERT, and ELECTRA for token-level extraction of city, state, country, and postal-code components in BPA-oriented address processing. Transformer-based architectures achieved the strongest predictive performance across both datasets, whereas the word-level models required substantially fewer computational resources. At the full-data regime, BERT and ELECTRA achieved 99.2% and 99.4% accuracy on OASIS, respectively, while their corresponding accuracies on USA Public Schools were 99.5% and 99.6%. DistilBERT also maintained high predictive performance with lower computational requirements than BERT.

The results show that model selection should consider practical requirements rather than accuracy alone. ELECTRA showed the best balance between predictive performance and inference efficiency among the transformer models. That makes it suitable for workflows that require high accuracy with lower latency than BERT. BERT is more appropriate when contextual understanding is the highest priority and higher inference time is acceptable. DistilBERT provides a middle option by offering strong transformer performance with lower computational cost. WordCNN is well suited for standardized and high-volume address processing, while WordLSTM is useful when sequential context is important and a slightly higher inference cost is acceptable.

These conclusions should be understood within the characteristics of the evaluated datasets. Both datasets contain relatively structured address information and mainly follow the North American address conventions. Consequently, the reported results should not be interpreted as universal performance for all types of real-world addresses. Validation on noisy, multilingual, cross-domain, and out-of-distribution address data remains necessary before broader deployment claims can be made.

7. FUTURE DIRECTIONS

Future research should extend the present benchmark toward address conditions that are less structured than those evaluated in this study. A first priority is to test the architectures on user-entered and OCR-derived addresses containing missing components, typographical errors, inconsistent ordering, uncommon abbreviations, and previously unseen geographic names. Such evaluation would determine whether the model ranking observed on the controlled datasets remains stable under realistic operational noise. Multilingual and cross-country datasets should also be investigated because the current benchmarks predominantly represent North American address conventions [50]. Repeated group-aware partitions and independent external datasets can further strengthen evidence of generalizability. Class-wise error analysis is also an important priority for

future evaluation. Future experiments will retain token-level predictions and report per-entity Precision, Recall, F1-score, and confusion matrices separately for city, state, country, and postal-code labels. Error categories such as abbreviations, ambiguous place names, numeric-token errors, and entity-boundary errors will also be examined. This analysis will provide more actionable evidence for selecting models in operational address-processing workflows.

From the deployment perspective, quantization, pruning, and knowledge distillation should be evaluated using the same predictive-performance and latency framework. This will help in determining whether transformer inference can be made faster without major loss in accuracy. Finally, BPA or RPA integration should be evaluated using practical measures such as throughput, failed-address rate, exception-handling frequency, reduction in human review, and computational cost. These extensions would move the evaluation beyond the controlled model benchmarking toward complete validation in real-world address-processing environments.

8. ETHICS AND DATA AVAILABILITY

Ethics Statement: This study uses organizational and publicly available address-related records for computational analysis. It does not involve any experimental intervention with human participants. The analysis was restricted to address fields required for component extraction only, and no sensitive personal information was used or inferred.

Data Availability Statement: The USA Public Schools dataset is publicly available from the repository mentioned in the manuscript. OASIS records were accessed through the International Aerospace Quality Group OASIS platform and are subject to the provider’s access and redistribution rules. Therefore, the OASIS records are not shared as part of this study.

Reproducibility Statement: The study clearly describes the data preparation, model settings, data splitting, evaluation measures, and statistical analysis to support reproducibility. The code used to run the experiments and generate the results will be shared in a public research repository after the manuscript is accepted, subject to the access and redistribution conditions of the source datasets.

REFERENCES

- [1] D. Baviskar, S. Ahirrao, V. Potdar, and K. Kotecha, “Efficient Automated Processing of the Unstructured Documents Using Artificial Intelligence: A Systematic Literature Review and Future Directions,” *IEEE Access*, vol. 9, pp. 72894–72936, 2021, doi: 10.1109/ACCESS.2021.3072900.
- [2] T. Chakraborti *et al.*, “From Robotic Process Automation to Intelligent Process Automation: – Emerging Trends –,” in *Business Process Management: Blockchain and Robotic Process Automation Forum*, vol. 393, A. Asatiani, J. M. García, N. Helander, A. Jiménez-Ramírez, A. Koschmider, J. Mendling, G. Meroni, and H. A. Reijers, Eds., in *Lecture Notes in Business Information Processing*, vol. 393. , Cham: Springer International Publishing, 2020, pp. 215–228. doi: 10.1007/978-3-030-58779-6_15.
- [3] S. Moreira, H. S. Mamede, and A. Santos, “Business Process Automation in SMEs: A Systematic Literature Review,” *IEEE Access*, vol. 12, pp. 75832–75864, 2024, doi: 10.1109/ACCESS.2024.3406548.
- [4] M. Skalický, Š. Šimsa, M. Uříčář, and M. Šulc, “Business Document Information Extraction: Towards Practical Benchmarks,” in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, vol. 13390, A. Barrón-Cedeño, G. Da San Martino, M. Degli Esposti, F. Sebastiani, C. Macdonald, G. Pasi, A. Hanbury, M. Potthast, G.

- Faggioli, and N. Ferro, Eds., in *Lecture Notes in Computer Science*, vol. 13390, Cham: Springer International Publishing, 2022, pp. 105–117. doi: 10.1007/978-3-031-13643-6_8.
- [5] F. Santos, R. Pereira, and J. B. Vasconcelos, “Toward robotic process automation implementation: an end-to-end perspective,” *Bus. Process Manag. J.*, vol. 26, no. 2, pp. 405–420, Sep. 2019, doi: 10.1108/BPMJ-12-2018-0380.
 - [6] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. doi: 10.1007/978-3-662-56509-4.
 - [7] B. Vajgel *et al.*, “Development of Intelligent Robotic Process Automation: A Utility Case Study in Brazil,” *IEEE Access*, vol. 9, pp. 71222–71235, 2021, doi: 10.1109/ACCESS.2021.3075693.
 - [8] S. Agostinelli, M. Lupia, A. Marrella, and M. Mecella, “Reactive synthesis of software robots in RPA from user interface logs,” *Comput. Ind.*, vol. 142, p. 103721, Nov. 2022, doi: 10.1016/j.compind.2022.103721.
 - [9] L.-V. Herm, C. Janiesch, A. Helm, F. Imgrund, A. Hofmann, and A. Winkelmann, “A framework for implementing robotic process automation projects,” *Inf. Syst. E-Bus. Manag.*, vol. 21, no. 1, pp. 1–35, Mar. 2023, doi: 10.1007/s10257-022-00553-8.
 - [10] Z. N. Jawad and V. Balázs, “Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review,” *Beni-Suef Univ. J. Basic Appl. Sci.*, vol. 13, no. 1, p. 4, Jan. 2024, doi: 10.1186/s43088-023-00460-y.
 - [11] F. A. Lievano-Martínez, J. D. Fernández-Ledesma, D. Burgos, J. W. Branch-Bedoya, and J. A. Jimenez-Builes, “Intelligent Process Automation: An Application in Manufacturing Industry,” *Sustainability*, vol. 14, no. 14, p. 8804, Jul. 2022, doi: 10.3390/su14148804.
 - [12] Kiran Babu Macha, “Integrating AI, ML, and RPA for end-to-end digital transformation in healthcare,” *World J. Adv. Res. Rev.*, vol. 25, no. 1, pp. 2116–2129, Jan. 2025, doi: 10.30574/wjarr.2025.25.1.0264.
 - [13] K. Heinrich, P. Zschech, C. Janiesch, and M. Bonin, “Process data properties matter: Introducing gated convolutional neural networks (GCNN) and key-value-predict attention networks (KVP) for next event prediction with deep learning,” *Decis. Support Syst.*, vol. 143, p. 113494, Apr. 2021, doi: 10.1016/j.dss.2021.113494.
 - [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.
 - [15] A. H. Huang, H. Wang, and Y. Yang, “FINBERT: A Large Language Model for Extracting Information from Financial Text,” *Contemp. Account. Res.*, vol. 40, no. 2, pp. 806–841, May 2023, doi: 10.1111/1911-3846.12832.
 - [16] Z. Li *et al.*, “Ensemble pretrained language models to extract biomedical knowledge from literature,” *J. Am. Med. Inform. Assoc.*, vol. 31, no. 9, pp. 1904–1911, Sep. 2024, doi: 10.1093/jamia/ocae061.
 - [17] M. Belmouadden, C. Dadouchi, and R. Pellerin, “Artificial intelligence applied in adaptive manufacturing process monitoring: a state-of-the-art in the era of automation,” *Procedia Comput. Sci.*, vol. 256, pp. 47–54, 2025, doi: 10.1016/j.procs.2025.02.094.
 - [18] J. Cascais Brás, R. F. Pereira, S. Moro, I. S. Bianchi, and R. Ribeiro, “Understanding How Intelligent Process Automation Impacts Business Continuity: Mapping IEEE/2755:2020 and ISO/22301:2019,” *IEEE Access*, vol. 11, pp. 134239–134258, 2023, doi: 10.1109/ACCESS.2023.3337159.
 - [19] W. L. Seow, I. Chaturvedi, A. Hogarth, R. Mao, and E. Cambria, “A review of named entity recognition: from learning methods to modelling paradigms and tasks,” *Artif. Intell. Rev.*, vol. 58, no. 10, p. 315, Jul. 2025, doi: 10.1007/s10462-025-11321-8.
 - [20] L. Hillebrand *et al.*, “KPI-BERT: A Joint Named Entity Recognition and Relation Extraction Model for Financial Reports,” in *2022 26th International Conference on Pattern Recognition (ICPR)*, Montreal, QC, Canada: IEEE, Aug. 2022, pp. 606–612. doi: 10.1109/ICPR56361.2022.9956191.
 - [21] R. Quille *et al.*, “Performance Analysis Method for Robotic Process Automation,” *Sustainability*, vol. 15, no. 4, p. 3702, Feb. 2023, doi: 10.3390/su15043702.
 - [22] V. K. Sharma and H. Kumar, “Enablers Driving Success of Artificial Intelligence in Business Performance: A TISM-MICMAC Approach,” *IEEE Trans. Eng. Manag.*, vol. 71, pp. 4665–4675, 2024, doi: 10.1109/TEM.2023.3236768.
 - [23] O. Givehchi, K. Landsdorf, P. Simoens, and A. W. Colombo, “Interoperability for Industrial Cyber-Physical Systems: An Approach for Legacy Systems,” *IEEE Trans. Ind. Inform.*, vol. 13, no. 6, pp. 3370–3378, Dec. 2017, doi: 10.1109/TII.2017.2740434.
 - [24] R. Catelli *et al.*, “A new Italian Cultural Heritage data set: detecting fake reviews with BERT and ELECTRA leveraging the sentiment,” *IEEE Access*, vol. 11, pp. 52214–52225, 2023, doi: 10.1109/ACCESS.2023.3277490.
 - [25] J. He and Y. Sun, “Research on the Financial Event Extraction Method Based on Fin-BERT,” *Int. J. Adv. Netw. Monit. Controls*, vol. 9, no. 4, pp. 67–74, Dec. 2024, doi: 10.2478/ijanmc-2024-0038.
 - [26] K. M. Hasib *et al.*, “MCNN-LSTM: Combining CNN and LSTM to Classify Multi-Class Text in Imbalanced News Data,” *IEEE Access*, vol. 11, pp. 93048–93063, 2023, doi: 10.1109/ACCESS.2023.3309697.
 - [27] R. C. Bhushan, R. K. Donthi, Y. Chilukuri, U. Srinivasarao, and P. Swetha, “Biomedical named entity recognition using improved green anaconda-assisted Bi-GRU-based hierarchical ResNet model,” *BMC Bioinformatics*, vol. 26, no. 1, p. 34, Jan. 2025, doi: 10.1186/s12859-024-06008-w.

- [28] E. Benevento, D. Aloini, and W. M. P. Van Der Aalst, "How Can Interactive Process Discovery Address Data Quality Issues in Real Business Settings? Evidence from a Case Study in Healthcare," *J. Biomed. Inform.*, vol. 130, p. 104083, Jun. 2022, doi: 10.1016/j.jbi.2022.104083.
- [29] S. Deng *et al.*, "A Recommendation System to Facilitate Business Process Modeling," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1380–1394, Jun. 2017, doi: 10.1109/TCYB.2016.2545688.
- [30] N. Adams, A. Augusto, M. Davern, and M. L. Rosa, "Addressing the Contemporary Challenges of Business Process Compliance: The Case for Process Mining in the Banking Industry," *Bus. Inf. Syst. Eng.*, vol. 68, no. 2, pp. 377–400, Apr. 2026, doi: 10.1007/s12599-025-00929-3.
- [31] C. Liu, Q. Zeng, L. Cheng, H. Duan, and J. Cheng, "Measuring Similarity for Data-Aware Business Processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 2, pp. 1070–1082, Apr. 2022, doi: 10.1109/TASE.2021.3049772.
- [32] R. Syed *et al.*, "Robotic Process Automation: Contemporary themes and challenges," *Comput. Ind.*, vol. 115, p. 103162, Feb. 2020, doi: 10.1016/j.compind.2019.103162.
- [33] S. Khantong and P. Sriboonlue, "Robotic Process Automation in Business Process Management: A Systematic Literature Review and an Integrated Framework," *Technologies*, vol. 14, no. 4, p. 225, Apr. 2026, doi: 10.3390/technologies14040225.
- [34] S. Lee *et al.*, "Entity-enhanced BERT for medical specialty prediction based on clinical questionnaire data," *PLOS ONE*, vol. 20, no. 1, p. e0317795, Jan. 2025, doi: 10.1371/journal.pone.0317795.
- [35] K. Kobayashi and K. Wakabayashi, "Partial Annotation Scheme for Active Learning on Named Entity Recognition Tasks," *J. Data Intell.*, vol. 1, no. 3, pp. 319–332, Sep. 2020, doi: 10.26421/JDI1.3-2.
- [36] M. W. Boota *et al.*, "Integrating social media and deep learning for real-time urban waterlogging monitoring," *J. Hydrol. Reg. Stud.*, vol. 56, p. 102070, Dec. 2024, doi: 10.1016/j.ejrh.2024.102070.
- [37] W. Lee and S. Lee, "Development of a Knowledge Base for Construction Risk Assessments Using BERT and Graph Models," *Buildings*, vol. 14, no. 11, p. 3359, Oct. 2024, doi: 10.3390/buildings14113359.
- [38] A. S. Tan, P. M. Dimayuga, and R. Gabud, "BiodiViz: Leveraging NER and RE for Automated Knowledge Graph Generation in Biodiversity Research," *Biodivers. Inf. Sci. Stand.*, vol. 8, p. e140428, Oct. 2024, doi: 10.3897/biss.8.140428.
- [39] W. Huang and X. Yu, "Named Entity Recognition Method Based on Multi-Feature Fusion," *Appl. Sci.*, vol. 15, no. 1, p. 388, Jan. 2025, doi: 10.3390/app15010388.
- [40] D.-L. Huang *et al.*, "A Combined Manual Annotation and Deep-Learning Natural Language Processing Study on Accurate Entity Extraction in Hereditary Disease Related Biomedical Literature," *Interdiscip. Sci. Comput. Life Sci.*, vol. 16, no. 2, pp. 333–344, Jun. 2024, doi: 10.1007/s12539-024-00605-2.
- [41] P. Liu, Z. Sun, and B. Zhou, "An ELECTRA-Based Model for Power Safety Named Entity Recognition," *Appl. Sci.*, vol. 14, no. 20, p. 9410, Oct. 2024, doi: 10.3390/app14209410.
- [42] A. Garg, "How Natural Language Processing Framework Automate Business Requirement Elicitation," *Int. J. Comput. Trends Technol.*, vol. 73, no. 5, pp. 47–50, May 2025, doi: 10.14445/22312803/IJCTT-V73I5P107.
- [43] N. N. Alotaibi, M. Saberi, M. Bandara, and T. Pornaveetus, "Intelligent Business Document Processing Using AI- and NLP-Based Techniques: A Systematic Literature Review," *Analytics*, vol. 5, no. 3, p. 31, Aug. 2026, doi: 10.3390/analytics5030031.
- [44] D. Yu, J. Chen, D. Wang, Y. Xu, Z. Xiang, and S. Deng, "DSIM: dynamic and static interest mining for sequential recommendation," *Knowl. Inf. Syst.*, vol. 64, no. 8, pp. 2267–2288, Aug. 2022, doi: 10.1007/s10115-022-01715-3.
- [45] D. Wang *et al.*, "Graph Intention Embedding Neural Network for tag-aware recommendation," *Neural Netw.*, vol. 184, p. 107062, Apr. 2025, doi: 10.1016/j.neunet.2024.107062.
- [46] M. Fischer, A. Hofmann, F. Imgrund, C. Janiesch, and A. Winkelmann, "On the composition of the long tail of business processes: Implications from a process mining study," *Inf. Syst.*, vol. 97, p. 101689, Mar. 2021, doi: 10.1016/j.is.2020.101689.
- [47] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.
- [48] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California: Association for Computational Linguistics, 2016, pp. 260–270. doi: 10.18653/v1/N16-1030.
- [49] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [50] B. S. Lancheros, G. Corpas Pastor, and R. Mitkov, "Data augmentation and transfer learning for cross-lingual Named Entity Recognition in the biomedical domain," *Lang. Resour. Eval.*, vol. 59, no. 2, pp. 665–684, Jun. 2025, doi: 10.1007/s10579-024-09738-8.