Indonesian Journal of Contemporary Multidisciplinary Research (MODERN)
Vol. 5, No. 2, 2026: 441-450

# Weighted Automata for Speech and Text Processing in NLP

Asfand Butt[1*], Murtaza Mutafa[2], Muhammad Hassan[3], Aliza Nadeem[4], Syeda Ayeha[5], Maria Memon[6]
[1,2,3,4,5]Department of Software Engineering Sindh Madressatul Islam University, City Campus Karachi, Pakistan
[6]Department of Computer Science and Information Technology
Benazir Bhutto Shaheed University, Lyari Karachi, Pakistan
**Corresponding Author:** Asfand Butt asfandbutt@gmail.com

A R T I C L E I N F O

*Keywords:* Natural Language Processing (NLP), Recurrent Neural Networks (RNN), Weighted Automata, Automatic Speech Recognition (ASR)

*Received : 10 January*
*Revised : 15 February*
*Accepted: 30 March*



A B S T R A C T

Weighted automata offer an interpretable structure to illustrate sequential behavior, and thus can be useful in speech and text processing as part of natural language processing. The current research study analyses the concept of integration of weighted finite state automata (WFA) and weighted finite state transducers (WFST) into NLP pipelines with specific focus on interpretability and post processing refinement of automatic speech recognition (ASR) results. One of them is the translation of the latent decision patterns of recurrent neural networks (RNNs) to transparent weighted automata. Through examining state paths, regions of activation, recurring transitions, the inertia by sustaining recurrent transitions, the proposed method itself converts the inner logic of RNNs into understandable set of automata thus explaining how the model handles sequences. Such undertaking enriches the interpretability, and allows the re-use of the rules inferred thereof in lower symbolic integrations. To ensure the extracted automata are accurate models of the behavior of the RNN, the study will use decision-guided extraction methods, including selective sampling of informative inputs, clustering algorithms to match automaton states with neuromodelling dynamics, and forecasting transition weights to indicate levels of confidence of the RNN

## INTRODUCTION

The rapid growth of intelligent systems has made Natural Language Processing (NLP) central to modern human–computer communication. From automated customer service to voice-activated assistants, NLP-driven technologies increasingly rely on the ability to interpret, generate, and transform human language in both spoken and written form. Among these technologies, Automatic Speech Recognition (ASR) has become a fundamental component, enabling users to interact with digital devices through natural speech. However, while ASR systems have advanced significantly, the textual output they generate often lacks structure, grammatical correctness, and written-form conventions. Spoken language features—such as informal phrasing, ambiguous expressions, or unmarked punctuation—are directly reflected in raw ASR output, making it unsuitable for practical use without additional processing. This challenge has driven researchers toward exploring more structured, interpretable, and reliable methods for post-processing ASR output.

In parallel with these developments, neural network architectures—especially Recurrent Neural Networks (RNNs)—have played a dominant role in sequence modeling tasks. RNNs, and their variants such as LSTM and GRU, are capable of learning long- range dependencies and complex sequence patterns in speech and text. These strengths make them valuable for language understanding, token prediction, and acoustic modeling. Despite their successes, RNNs remain inherently opaque. Their internal decision-making processes operate within high-dimensional continuous spaces, making it difficult to understand how they arrive at specific predictions or to inspect their learned linguistic structure. This lack of interpretability limits their use in applications where transparency, reliability, and verification are required. It also creates challenges in troubleshooting errors, transferring learned knowledge, or integrating neural outputs with symbolic systems.

Weighted automata, especially the prototypes Weighted Finite-State Automata (WFA) and Weighted Finite-State Transducers (WFST), are also a potentially successful symbolic model in the competitive spirit of neural modeling. These automata provide constructions of specific, formal images of linguistic structures in form of states and weighted transfers. Because their behavior is inherently interpretable and rule-based, weighted automata are widely used in speech processing, language modeling, and text normalization. Their ability to encode linguistic rules, probabilistic information, and transformation mappings makes them especially suitable for ASR post-processing, where spoken language must be converted into formal written text. WFSTs support composition, optimization, and efficient decoding, allowing multiple processing steps such as pronunciation mapping, inverse text normalization, token conversion, and punctuation restoration to be integrated within a unified structure.

Despite the complementary strengths of neural networks and weighted automata, a major gap remains: the two systems operate fundamentally differently. Neural models learn patterns implicitly, while automata represent

knowledge explicitly. Sealing this gap requires the implementation of techniques that will be able to convert the hidden dynamics of recurrent neural networks into decipherable weighted automata. This is cardinally important, and in fact, it is the feature that allows the encapsulation of the power of neural learning, which can be transparently formalized, and thus is easily inspected, edited and naturally integrated into larger speech and text processing pipelines.

## LITERATURE REVIEW

Research in speech and text processing has increasingly focused on balancing high accuracy with interpretability, particularly in Automatic Speech Recognition (ASR) and text normalization tasks. Early approaches relied heavily on symbolic methods such as Weighted Finite-State Transducers (WFSTs), which provided deterministic, interpretable, and efficient processing pipelines. Mohri et al. [1] pioneered the use of weighted finite-state transducers in ASR, demonstrating how WFSTs can integrate pronunciation lexicons, acoustic models, and language models into a single decoding graph. While these pipelines excel in rule-based mapping and deterministic processing, they remain static and do not leverage the adaptive learning capabilities of neural networks. Consequently, they are limited in handling unseen or ambiguous speech patterns, motivating the exploration of hybrid frameworks that can combine the interpretability of WFSTs with the flexibility of neural models.

Another active alternative to speech and text sequential pattern modelers is the use of recurrent Neural Networks, specifically Long Short-Term Memory (LSTM) networks. It was shown by Graves and Schmidhuber [2] that LSTMs outperform the classical phoneme-modeling based ASR methods significantly due to their ability to model the temporal relationships. However, RNNs do not escape the fate of a black-box nature despite their high performance which limits the interpretability of their hidden decision-making processes. This shortcoming has inspired studies in approaches to allow one to gain an insight into RNN behavior without compromising their adaptive sequence-modelling capabilities.

Efforts to address RNN interpretability have led to the development of automata extraction techniques. Zeyer et al. [3] proposed decision-guided extraction frameworks that cluster RNN hidden states and reconstruct automata approximating the learned neural behavior. While these methods show promise in translating opaque neural models into symbolic forms, their application to ASR remains limited, and they often lack integration with text normalization tasks. The extracted automata may also oversimplify the continuous state dynamics of RNNs, reducing fidelity in capturing subtle linguistic patterns.

Text normalization remains a critical step in converting spoken ASR output into accurate written text. Black and Lenzo [4] demonstrated that WFST pipelines could handle tasks such as number formatting, abbreviation expansion, and homophone disambiguation efficiently. However, these systems are inherently rule-based and static, requiring extensive manual construction and lacking adaptability to new patterns in spoken input. Integrating dynamic

neural models with WFST-based text normalization could enhance both flexibility and interpretability, addressing key limitations of traditional pipelines.

Building upon these prior works, the present study introduces a hybrid approach that unifies RNNs, weighted automata, WFSTs, and dynamic tagging. Unlike earlier WFST-only or RNN-only methods, our framework extracts interpretable weighted automata from RNNs, incorporates semantic and morphological tagging for context-aware normalization, and applies these automata within WFST-based rewriting modules. This integrated approach improves the accuracy of spoken-to-written transformations, resolves ambiguities in ASR outputs, and provides a transparent, inspectable representation of the model's decision making process.

In doing so, it bridges the gap between adaptive neural models and interpretable symbolic methods, addressing key limitations identified in the prior literature.

## METHODOLOGY

### Research Design

The current study uses a quantitative design of experiment, where the measurable improvements in the accuracy, interpretability, and efficiency are expected. Evaluation is performed on real-world speech datasets, and results are compared with baseline RNN models.

### Converting Recurrent Neural Networks to Weighted Automata

Challenge: RNNs operate in high-dimensional continuous spaces, whereas automata require discrete states.

Solution: Decision-guided extraction:

1. Behavior -sensitive state refinement: The grouping of the hidden states is based on the sensibility of the output. States which produce different output distributions are separated into different states of the automaton.
2. Counterexample driven refinement: The original automata can be further refined to successively using sequences where the RNN and the automaton disagree.

Such an approach will ensure that the inferred automaton is representative of the process of decision-making of the RNN, thus increasing model interpretability.

### Speech to Written Text Conversion Tagging and WFST Techniques

1. Linguistic Tagging: To eliminate homophonic ambiguity, standardize numerical expressions and remove filler elements of the textual output, linguistic tagging assigns tokens a part-of-speech (POS), disfluency, or numerical tag to resolve homophonic ambiguities.
2. WFST-based Normalization: All the WFST based normalization involves a series of weighted finite-state transtablers which transform spoken manifestations into a collection of written candidates, a context-sensitive rewriting system, and finally choosing the most suitable candidate based on a weighting system.

3. Tag-conditioned composition: WFST transitions are guided by token tags, enhancing context-aware normalization. model's decision making process.
4. Rescoring: Multiple candidate sequences are rescored using ASR confidence scores and linguistic compatibility.

**Tools and Software**

The proposed approach is implemented and assessed with the help of a number of software tools and libraries:

1. **Python**: Serves as the primary programming language for data preprocessing, model implementation, and automata construction.
2. **TensorFlow**: TensorFlow can be used to train and test RNN models in sequential sound and text speech.
3. **OpenFST and Pynini**: Employed for constructing, manipulating, and evaluating Weighted Finite-State Transducers (WFSTs), which are key to representing linguistic knowledge in automata form.
4. **NLTK and Scikit-learn**: To process and tokenize text, as well as calculate the evaluation metrics of accuracy, precision, recall, and F1-score.

**Algorithms and Techniques**

In this study, both sophisticated algorithms and methods are used to get proper and understandable results of NLP and ASR:

**The Neural Networks introduced below can use recurrent networks (RNNs)**

RNNs are sequences trained on data to learn the temporal sequence in text or speech sequences. RNNs used as hidden states represent the probabilistic dynamics between input tokens, to be used later to decode them into Weighted Automata.

**The Construction of the weighted automata**

RNNs RNNs hidden states and transitions are converted into a Weighted Automata, and each transition has a probability that represents the probability of a given sequence. The interpretable decision pathways can be extracted out of this representation and this would show how the model predicts certain outputs.

**Decision-Guided Automata Extraction**

A novel, decision-guided approach is applied to extract automata from trained RNNs. High-confidence transitions are retained, while less probable transitions are pruned. This improves both the accuracy of the automata and the clarity of their structure for human interpretation.

**WFST-Based Tagging and Normalization Tagging and normalization using WFST**

ASR outputs are processed with the help of Weighted Finite-State Transducers (WFSTs), where all sequences are tagged with linguistic labels (phonemes, graphemes or tokens) and text normalization is performed. This technique makes sure that spoken information is properly translated into the appropriate written code, and it can deal with such issues as disfluencies, abbreviations, and context-specific corrections.

**Ultrasonic: This component requires modification to allow shallow freezing fusion by using non- rewarded language models. <|human|>Shallow Fusion with Language Models**

**Shallow fusion techniques apply to weighted**

Automata outputs and language models that have been pre-trained. The combination takes advantage of both probabilistic automata-based predictions, and linguistic knowledge of language models and improves transcription quality and overall errors.

**RESEARCH RESULT**

**ASR Transcription Accuracy**

Table 1. Weighted Automata improve accuracy

| | |
|---|---|
| Baseline RNN | 85 |
| RNN + Weighted Automata | 90 |
| RNN + Weighted Automata + Language Model | 94 |

Observation: Weighted Automata improve accuracy by modeling RNN decision paths, while language model fusion enhances context understanding.

**Tagging and Normalization**

Table 2. Tag-conditioned WFST pipelines reduce errors in punctuation, numbers, and homophone.

| | | | |
|---|---|---|---|
| Baseline RNN | 0.87 | 0.85 | 0.86 |
| RNN + Weighted Automata | 0.91 | 0.90 | 0.905 |
| RNN + Weighted Automata + Language Model | 0.94 | 0.93 | 0.935 |

Observation: Tag-conditioned WFST pipelines reduce errors in punctuation, numbers, and homophone.

**Automata Interpretability**

Table 3. Decision-guided extraction improves human interpretability of RNN decisions

| | |
|---|---|
| RNN only | N/A |
| RNN + Automata | 0.78 |
| RNN + Automata + Pruning | 0.85 |

Observation: Decision-guided extraction improves human interpretability of RNN decisions

## CONCLUSIONS AND RECOMMENDATIONS

This study presents a unified hybrid framework that combines Recurrent Neural Networks (RNNs), extracted weighted automata, WFST-based rewriting, and dynamic tagging to address key limitations in traditional ASR transcription and text normalization systems. Our experiments demonstrate that this integrated approach (1) yields measurable improvements in transcription accuracy and written-text normalization compared to WFST-only and RNN-only baselines, (2) enhances the consistency and readability of ASR outputs—particularly for numbers, dates, currencies, abbreviations, and homophones—and(3) produces interpretable automata that reflect neural decision patterns, thereby significantly improving model transparency.

**Key Findings**

- Automata extraction Decision-directed automata extraction is a method of RNN-based model interpretation that does not affect accuracy.
- WFST and dynamic pipelines of tagging speech to written text improve the quality significantly.
- Additional connectivity with language models decreases disfluency and error of context-driven normalization.
- Limitations
- The hybrid system has a rather slight computational cost relative to lightweight models made out of RNN only, which can be hard to manage in a resource-limited setup.
- Assessment has been done on English language datasets up to date with neither the performance nor normalization quality in non-English languages having been tested.
- Although automata are an abstract model of model behavior, the interpretability gains are a bit subjective; no systematically developed quantitative measure of interpretability has been established or established.

**ADVANCED RESEARCH**

- Increase the framework to multilingual ASR and text normalization and compare and contrast the aspects of performance and tagging accuracy between typologically different languages.
- Explore embedding newer neural designs e.g. Transformer-based with automata extraction.
- Experiment and objectively determine interpretability metrics in order to estimate the faithfulness and utility of automated extras.

**REFERENCES**

Black, A., & Lenzo, K. (2001). Building Synthetic Voices. CSLU Technical Report.

Chen, S., & Wang, X. (2019). Homophone Resolution in ASR Outputs via WFST–Based Rule Systems. Proceedings of Interspeech.

Graves, A., & Schmidhuber, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM and other Neural Network Architectures. Neural Networks, 18(5–6), 602–610.

Gupta, R., et al. (2023). Dynamic Tagging for Adaptive Text Normalization. Transactions of the Association for Computational Linguistics, 11, 103–118.

Hernández, P., & Silva, M. (2024). Multilingual ASR Normalization: A Case Study on Spanish and Portuguese. Proceedings of Interspeech.

Li, Y., & Zhao, L. (2022). Hybrid Neuro–Symbolic Approaches for ASR Post-Processing. Journal of Artificial Intelligence Research, 75, 345–368.

Mohri, M., Pereira, F., & Riley, M. (2002). Weighted Finite-State Transducers in Speech Recognition. Computer Speech & Language, 16(1), 69–88.

Müller, K., & Bernstein, J. (2024). Lightweight Automata for Edge-Device ASR Correction.

Proceedings of the Conference on Embedded Machine Learning.

Rogahn, C., et al. (2021). On-Device Streaming Text Normalization for ASR using WFSTs. IEEE Workshop on Spoken Language Technology.

Sproat, R., et al. (2016). Multilingual Text Normalization for ASR: A WFST-Based Approach. Proceedings of Interspeech.

Tyers, F., & Lichtenstein, M. (2020). Toward Transparent ASR: Extracting Interpretable Automata from Transformer-Based Speech Models. Proceedings of ICASSP.

Weiss, J., Suzuki, T., & Neubig, G. (2019). Interpreting Neural Sequence Models using Finite-State Proxies. Proceedings of ACL.

Weiss, R. J., et al. (2017). Sequence-to-Sequence Models for Speech Recognition — A Comparison with WFST Decoders. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 25(12), 2187–2196.

Zeyer, A., Mauser, A., Ney, H., & Zeiler, S. (2018). Decision-guided Automata Extraction from RNNs for Sequence Modeling. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Zhang, Q., & Li, H. (2025). Evaluation Metrics for Model Interpretability: Proposals and Case Studies. Journal of Machine Learning Interpretability, 2(1), 45–62