Reinforcement Learning for Part-of-Speech Tagging

A Constraint-Aware and Oracle-Augmented Approach

Robert Dincă
University of Bucharest

Ştefan Tacu
University of Bucharest

Ruxandra Petrescu
University of Bucharest

Albert Moloceniuc
University of Bucharest

January 2026

Abstract

Part-of-Speech (POS) tagging is a core task in Natural Language Processing, traditionally formulated as a supervised sequence labeling problem. While modern neural architectures achieve high accuracy, they operate under a static prediction paradigm that limits explicit control over sequential decision-making, structured constraints, and interactive feedback.

This work reformulates POS tagging as a sequential decision-making problem using Reinforcement Learning (RL). We model the task as a finite Markov Decision Process in which an agent assigns POS tags incrementally, one token at a time, while interacting with a linguistically constrained environment. The formulation enables explicit modeling of action validity, reward shaping, and uncertainty-aware decision strategies.

Two distinct Reinforcement Learning environments are investigated. The first environment addresses standard POS tagging under lexical and grammatical constraints, focusing on stabilizing learning in large discrete action spaces through Offline Reinforcement Learning techniques. The second environment extends this formulation by introducing a Human-in-the-Loop Oracle mechanism, allowing the agent to defer decisions under high uncertainty at a controlled cost.

The proposed framework combines structured state representations, lexical action masking, conservative value-based learning algorithms, and uncertainty-driven Oracle invocation. This design allows the agent to balance autonomy and reliability while maintaining linguistic plausibility.

Beyond POS tagging, the framework illustrates how constraint-aware Offline Reinforcement Learning and selective human intervention can be integrated into structured sequential prediction tasks.

Contents

1 Introduction

Part-of-Speech (POS) tagging is a fundamental task in Natural Language Processing (NLP), providing essential syntactic information for downstream applications such as syntactic parsing, information extraction, semantic analysis, and machine translation. The task consists of assigning a grammatical category to each token in a sentence, typically based on both lexical properties and contextual cues. Due to its foundational role, POS tagging has been extensively studied and serves as a standard benchmark for evaluating sequence modeling approaches in NLP.

Historically, POS tagging has been formulated as a supervised sequence labeling problem. Early approaches relied on probabilistic graphical models, most notably Hidden Markov Models (HMMs), which model the joint probability of word and tag sequences under strong independence assumptions. Later developments introduced Conditional Random Fields (CRFs), enabling more expressive feature representations and direct modeling of conditional distributions over tag sequences. More recently, deep learning architectures, including Bidirectional Long Short-Term Memory networks (Bi-LSTMs) and Transformer-based models, have become the dominant paradigm, achieving state-of-the-art performance by leveraging contextual word representations and large-scale pretraining.

Despite their empirical success, these approaches share a common structural assumption: POS tagging is treated as a static prediction problem in which the model observes an entire sentence and predicts all tags simultaneously. While effective, this formulation abstracts away the inherently sequential nature of the task, in which each tagging decision depends on previously assigned tags and influences subsequent ones. Moreover, static supervised models offer limited mechanisms for explicitly incorporating structured constraints, dynamic decision costs, or interactive feedback during inference.

An alternative perspective is to view POS tagging as a sequential decision-making process. From this viewpoint, tagging a sentence corresponds to a sequence of actions taken by an agent, where each action assigns a POS tag to the current word based on the current context and prior decisions. This interpretation naturally aligns with the Reinforcement Learning (RL) framework, in which an agent interacts with an environment, observes states, selects actions, and receives feedback in the form of rewards.

However, applying Reinforcement Learning to natural language tasks presents significant challenges. The state and action spaces are large and highly structured, reward signals are sparse, and unconstrained exploration can easily lead to degenerate policies. These difficulties are particularly pronounced in Online Reinforcement Learning settings, where agents must learn directly through interaction with the environment. As a result, naive RL formulations often exhibit unstable learning dynamics and poor convergence behavior when applied to linguistic prediction tasks.

To address these challenges, this work adopts an Offline Reinforcement Learning formulation for POS tagging. Learning is performed on a fixed, annotated corpus, allowing the agent to exploit existing linguistic data while avoiding the instability associated with uncontrolled exploration. Furthermore, strong inductive biases are introduced through structured state representations, lexical action constraints, and conservative learning objectives. These design choices substantially reduce the effective search space and enable stable learning in a high-dimensional discrete domain.

In addition to autonomous decision-making, natural language tasks frequently involve intrinsic ambiguity, where multiple interpretations may be plausible even given local context. In practical systems, such uncertainty is often resolved through external supervision or human intervention. Motivated by this observation, the proposed framework also investigates an extended setting in which the agent is equipped with a Human-in-the-Loop Oracle mechanism. This mechanism allows the agent to defer decisions in cases of high uncertainty at a controlled cost, enabling a principled trade-off between autonomy and reliability.

The contributions of this work can be summarized as follows:

- Reformulating Part-of-Speech tagging as a finite Markov Decision Process suitable for Reinforcement Learning.
- Proposing an Offline Reinforcement Learning framework that incorporates lexical and grammatical constraints to stabilize learning.
- Defining and evaluating two distinct environments: a standard POS tagging environment and an Oracle-augmented environment with human-in-the-loop decision support.
- Providing an empirical analysis of agent behavior, uncertainty management, and performance under different learning configurations.

2 Background and Related Work

Part-of-Speech tagging has a long history in Natural Language Processing and has served as a testbed for a wide range of sequence modeling techniques. Early approaches to POS tagging were based on probabilistic generative models, most notably Hidden Markov Models (HMMs). In this framework, the tagging problem is formulated as the inference of a hidden tag sequence given an observed word sequence, under assumptions of conditional independence. Despite their simplicity, HMM-based taggers demonstrated that sequential dependencies between tags play a crucial role in syntactic disambiguation.

Subsequent work introduced discriminative models, particularly Conditional Random Fields (CRFs), which directly model the conditional distribution of tag sequences given observed words. By relaxing the independence assumptions of HMMs and allowing the incorporation of rich, overlapping feature sets, CRFs achieved substantial improvements in accuracy and robustness. As a result, CRFs became a dominant approach for POS tagging and other sequence labeling tasks for many years.

The emergence of deep learning led to a new generation of POS tagging models based on neural sequence encoders. Bidirectional Long Short-Term Memory networks (Bi-LSTMs) enabled models to capture both past and future context, significantly improving performance over purely left-to-right architectures. These models typically operate by encoding the entire sentence into contextualized word representations, followed by a classification or CRF decoding layer. More recently, Transformer-based architectures have further advanced the state of the art by leveraging self-attention mechanisms and large-scale pretraining on massive corpora.

Although these neural models achieve high accuracy, they share a common conceptual structure: POS tagging is treated as a static inference problem. The model processes the entire input sentence and predicts all tags simultaneously, without explicitly modeling the sequential decision process involved in assigning tags one by one. From this perspective, the task is solved through global pattern recognition rather than through explicit action selection under uncertainty.

Reinforcement Learning offers an alternative modeling paradigm in which sequential prediction tasks are formulated as decision-making problems. In RL, an agent interacts with an environment over time, observes states, selects actions, and receives feedback in the form of rewards. This framework has been successfully applied to a variety of problems involving sequential control, including robotics, game playing, and resource allocation. In the context of Natural Language Processing, RL has been explored for tasks such as dialogue management, text generation, and structured prediction.

Several prior works have investigated the use of Reinforcement Learning for structured NLP tasks, including POS tagging, syntactic parsing, and named entity recognition. These approaches typically model tagging as a sequence of actions, where each action assigns a label to a token. While conceptually appealing, early RL-based formulations often struggled with instability, slow convergence, and poor sample efficiency. These difficulties are largely attributable to the large discrete action spaces, sparse reward signals, and the challenges of exploration in high-dimensional linguistic domains.

Recent advances in Offline Reinforcement Learning have addressed many of these limitations. Offline RL methods learn policies from fixed datasets without requiring online interaction with the environment, making them well-suited for domains where exploration is expensive or unsafe. Techniques such as conservative value estimation, batch-constrained policy learning, and action masking have been proposed to mitigate extrapolation error and stabilize learning when operating on static datasets.

In parallel, research on Human-in-the-Loop machine learning has emphasized the importance of integrating human expertise into automated systems. Rather than replacing human decision-makers entirely, such systems are designed to escalate uncertain cases to human experts while handling routine decisions autonomously. In NLP, this paradigm is particularly relevant due to the inherent ambiguity of language and the potential downstream impact of incorrect predictions.

The present work builds on these strands of research by combining a Reinforcement Learning formulation of POS tagging with constraint-aware Offline RL techniques and a Human-in-the-Loop Oracle mechanism. By positioning POS tagging as a sequential decision-making problem under structured constraints, the proposed framework aims to bridge the gap between high-performing supervised models and interactive, uncertainty-aware decision systems.

3 Problem Formulation

In this work, Part-of-Speech tagging is formulated as a sequential decision-making problem within the Reinforcement Learning (RL) framework. Rather than predicting an entire tag sequence in a single inference step, the tagging process is modeled as a sequence of decisions made incrementally, one token at a time. This perspective allows the task to be expressed formally as a Markov Decision Process (MDP), providing a principled foundation for learning under uncertainty and structured constraints.

An MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the transition dynamics, $\mathcal{R}$ the reward function, and $\gamma \in (0, 1]$ the discount factor. In the context of POS tagging, each episode corresponds to the tagging of a single sentence, and the episode terminates after the final token has been processed.

At each time step t, the agent observes a state $s_t \in \mathcal{S}$ that encodes information about the current position in the sentence, the current token, and relevant contextual features derived from previous decisions. Based on this state, the agent selects an action $a_t \in \mathcal{A}$ corresponding to the assignment of a POS tag to the current word. The environment then transitions deterministically to the next state s_{t+1}, advancing the token index and updating the contextual representation.

The Markov property is satisfied by construction: the state representation at time t is designed to contain all information necessary to make an optimal tagging decision for the current token, independent of earlier history beyond what is explicitly encoded in the state. This formulation enables the use of standard value-based and policy-based Reinforcement Learning algorithms without requiring access to future tokens beyond the modeled context.

A key characteristic of this formulation is the strictly left-to-right processing order. The agent assigns tags sequentially, and once a decision is made, it cannot be revised. While this constraint distinguishes the approach from bidirectional sequence models, it reflects a realistic online decision-making setting and enables explicit modeling of decision dependencies between adjacent tags.

The objective of the agent is to learn a policy $\pi(a \mid s)$ that maximizes the expected cumulative discounted reward over an episode. In the POS tagging setting, this objective corresponds to producing syntactically and lexically consistent tag sequences that align with ground-truth annotations while respecting structural constraints imposed by the environment.

Importantly, this section defines only the abstract decision-theoretic formulation of the task. The specific design choices regarding state representation, action availability, reward shaping, and environment extensions are introduced in subsequent sections. This separation allows multiple experimental environments to be analyzed within a unified formal framework.

4 Dataset and Preprocessing

The proposed Reinforcement Learning framework is trained and evaluated using a static, fully annotated linguistic corpus. Because the task is formulated within an Offline Reinforcement Learning setting, careful dataset selection and preprocessing are essential to ensure stable learning dynamics and meaningful evaluation. Unlike online environments, no new data is generated through interaction; all transitions are derived from the fixed corpus.

4.1 Dataset Selection

All experiments are conducted on the Brown Corpus, a widely used benchmark dataset in computational linguistics. The corpus consists of approximately 500 documents spanning a diverse range of genres, including news, editorials, fiction, academic writing, and informal prose. In total, the dataset contains over one million tokens annotated with gold-standard Part-of-Speech labels.

The Brown Corpus is particularly well-suited for this study for several reasons. First, its genre diversity supports evaluation across varied linguistic contexts. Second, the availability of high-quality manual annotations enables precise offline supervision. Finally, the moderate dataset size allows controlled experimentation with Reinforcement Learning methods without the computational overhead associated with very large corpora.

4.2 Tagset Simplification

The original Brown tagset contains more than 80 fine-grained POS labels, encoding detailed syntactic and morphological distinctions. While this level of granularity is appropriate for traditional supervised learning, it significantly increases the action space in a Reinforcement Learning formulation, exacerbating exploration difficulty and reward sparsity.

To address this issue, the original tagset is mapped to the Universal POS Tagset, which consists of twelve coarse-grained categories: NOUN, VERB, ADJ, ADV, PRON, DET, ADP, NUM, CONJ, PRT, punctuation, and X (other). This mapping reduces the action space while preserving the core syntactic distinctions required for POS tagging, thereby improving learning stability and sample efficiency.

4.3 Lexical Dictionary Construction

A lexical dictionary is constructed from the training portion of the corpus to support action constraints during learning and inference. For each unique word type observed in the dataset, the dictionary records the set of POS tags with which the word appears in the annotations.

Formally, the dictionary defines a mapping from a word w to a subset of valid tags $\mathcal{T}(w) \subseteq \mathcal{A}$. During learning and inference, this mapping is used to restrict the set of admissible actions for a given token. The dictionary is constructed once by scanning the training data and is treated as fixed throughout all experiments.

4.4 Morphological Feature Extraction

To support generalization to rare and out-of-vocabulary words, each token is augmented with a set of morphological features derived from its surface form. These features are encoded as a fixed-length binary vector and capture orthographic patterns that are strongly correlated with syntactic function.

The extracted features include common suffixes (such as *-ing*, *-ed*, and *-ly*), capitalization patterns, the presence of digits, and the occurrence of special characters such as hyphens. By incorporating these features, the agent can infer plausible POS categories based on form-based cues rather than relying solely on lexical identity.

4.5 Embedding Representation

In addition to morphological features, tokens are represented using dense distributional word embeddings. Pretrained embeddings are employed to encode semantic information derived from large external corpora. For each time step, the embedding of the current word is included in the representation, providing contextual semantic information beyond surface-level features.

In some configurations, limited look-ahead information is incorporated by including the embedding of the subsequent word in the sequence. This design choice provides the agent with a restricted form of future context while preserving the left-to-right decision structure imposed by the Reinforcement Learning formulation.

4.6 Train–Test Split

The corpus is partitioned into disjoint training and evaluation sets. All learning is performed exclusively on the training split, while the evaluation split is used only for performance assessment. Because the learning paradigm is offline, the agent does not interact with the evaluation data during training, ensuring a clear separation between learning and testing phases.

This preprocessing pipeline produces a static dataset of state–action–reward transitions suitable for Offline Reinforcement Learning algorithms and enables reproducible experimentation across different learning configurations.

5 Reinforcement Learning Environments

This section defines the concrete Reinforcement Learning environments used in this study. While the previous sections introduced the abstract Markov Decision Process formulation, the environments specified here instantiate this formulation with concrete state representations, action spaces, and transition logic. Importantly, two distinct environments are defined and evaluated separately. Although they share common structural components, they differ in action availability and decision mechanisms.

5.1 Environment 1: Standard POS Tagging

The first environment models Part-of-Speech tagging as a purely autonomous sequential decision-making process. Each episode corresponds to a single sentence from the corpus, processed from left to right. The agent assigns a POS tag to each token without access to external supervision during inference.

At each time step t, the environment provides the agent with a state representation encoding the current position in the sentence and relevant contextual information. Based on this state, the agent selects an action corresponding to one of the POS tags in the Universal Tagset. Once an action is taken, the environment deterministically advances to the next token and updates the state accordingly.

The action space in this environment consists exclusively of POS tag assignments. No corrective or auxiliary actions are available. Consequently, the agent must commit to a tagging decision at every step, and all decisions directly influence subsequent states through the encoded context. This environment serves as the baseline setting for evaluating Reinforcement Learning methods applied to POS tagging under structured constraints.

5.2 Environment 2: Oracle-Augmented POS Tagging

The second environment extends the standard POS tagging setting by introducing an explicit Human-in-the-Loop Oracle mechanism. This extension is motivated by the inherent ambiguity of natural language, where certain tokens may admit multiple syntactically valid interpretations even under local context, making fully autonomous decisions unreliable in specific cases.

Unlike formulations in which the action space is expanded dynamically, the action space in this environment is fixed and consists of exactly three discrete actions:

- **KEEP**, which commits the agent's current autonomous POS prediction;
- **SHIFT**, which selects the 2nd most probable POS tag.
- **DICT**, which requests the correct POS tag from an external Oracle.

Selecting the **DICT** action triggers Oracle intervention, whereby the environment returns the ground-truth POS tag for the current token and advances the episode to the next state. Invoking the Oracle incurs a fixed negative reward, reflecting the cost of external supervision or human intervention. This cost is deliberately chosen to be smaller in magnitude than the penalty associated with an incorrect autonomous prediction, ensuring that Oracle usage is beneficial only under sufficient uncertainty.

Aside from the introduction of the **DICT** action, the structure of the environment remains consistent with the standard setting. Episodes correspond to individual sentences, transitions are deterministic, and the left-to-right processing order is strictly preserved. No additional stochasticity is introduced by the environment; all uncertainty arises from the agent's internal value estimates.

This environment enables the study of uncertainty-aware decision-making under a constrained action space. Rather than maximizing accuracy alone, the agent must learn a calibrated strategy that balances autonomous decision-making against selective external intervention. The fixed three-action formulation ensures tight alignment between the theoretical environment definition and the implemented system, allowing for precise analysis of Oracle efficiency and decision behavior.

5.3 Shared Properties and Design Rationale

Both environments operate on the same underlying dataset, share the same episode structure, and use compatible state representations. This design ensures that differences in performance and behavior can be attributed to the availability of the Oracle mechanism rather than to unrelated architectural factors.

By explicitly separating the environments, the framework supports a controlled comparison between autonomous tagging and human-assisted tagging within a unified Reinforcement Learning formulation. This separation also allows each environment to be analyzed independently, avoiding ambiguity in experimental interpretation.

6 Agent Architecture and Learning Algorithms

This section describes the architecture of the Reinforcement Learning agent and the learning algorithms employed to train it in an offline setting. The design choices are guided by the structured nature of the POS tagging task, the discrete action space, and the need for stable learning under fixed data distributions.

6.1 Agent Architecture

The agent is implemented as a value-based Reinforcement Learning model that approximates the state–action value function $Q(s, a)$. Given a state representation at time step t, the network outputs a scalar value for each admissible action, representing the expected cumulative reward of selecting that action in the current state.

The input to the network is a fixed-length vector obtained by concatenating multiple components of the state representation, including contextual encodings of previous decisions, morphological features of the current token, and dense word embeddings. This composite representation captures both syntactic and semantic information relevant to POS disambiguation.

The neural network architecture consists of a feed-forward multilayer perceptron with two hidden layers. Rectified Linear Unit (ReLU) activations are used to introduce non-linearity, while the output layer uses a linear activation to produce Q-values for each action. This relatively shallow architecture was selected to balance representational capacity with training stability, as deeper networks were observed to introduce unnecessary variance in preliminary experiments.

6.2 Value-Based Learning

Learning is performed using value-based Reinforcement Learning algorithms, with a primary focus on Deep Q-Networks (DQN). The DQN framework updates the Q-function by minimizing the temporal-difference error between predicted Q-values and target values computed from observed transitions in the dataset.

Because the learning setting is strictly offline, the replay buffer is populated entirely from pre-collected transitions derived from the annotated corpus. No new transitions are generated through interaction with the environment during training. This constraint necessitates careful algorithmic choices to avoid extrapolation beyond the support of the data distribution.

6.3 Offline Reinforcement Learning Considerations

To address the challenges associated with offline learning, conservative learning strategies are employed. These strategies aim to prevent the agent from assigning unrealistically high values to actions that are poorly represented in the dataset. Action masking based on lexical validity further restricts the set of actions considered during both training and inference, ensuring alignment between the learned policy and observed linguistic patterns.

In addition, prioritized sampling techniques are used to focus learning on informative or ambiguous transitions. By emphasizing transitions with higher prediction error, the agent allocates greater learning capacity to difficult cases, which are common in syntactically ambiguous contexts.

Overall, the combination of value-based learning, conservative estimation, and structured action constraints enables stable training in a high-dimensional discrete decision space without requiring online exploration.

7 Reward Design and Constraints

The reward function plays a central role in shaping the agent's behavior and guiding learning toward linguistically plausible solutions. In the context of POS tagging, naive reward formulations based solely on final accuracy tend to produce sparse and unstable learning signals. To mitigate these issues, a structured reward design is adopted, incorporating both correctness-based feedback and constraint-driven penalties.

7.1 Base Reward Signal

At each time step, the agent receives a base reward reflecting the correctness of the selected POS tag relative to the ground-truth annotation. Correct predictions are rewarded positively, while incorrect predictions incur a negative penalty. This immediate feedback encourages local accuracy and provides a dense learning signal across the sequence.

7.2 Lexical Constraints

A key component of the reward design is the enforcement of lexical constraints derived from the training corpus. For each token, only a subset of POS tags observed for that word in the dataset is considered lexically admissible. When the agent selects an action outside this admissible set, an additional penalty is applied.

This mechanism discourages linguistically implausible decisions, such as assigning a verb tag to a determiner, and substantially reduces the effective action space. Lexical constraints are implemented both as reward penalties and as hard action masks, ensuring that invalid actions are neither selected nor reinforced during learning.

7.3 Grammatical Transition Constraints

Beyond lexical validity, syntactic plausibility is modeled through penalties applied to unlikely tag transitions. Empirical transition statistics are estimated from the training data, capturing the frequency of consecutive POS tag pairs. Transitions with extremely low empirical probability are penalized to discourage syntactically inconsistent sequences.

These grammatical constraints introduce a form of structural bias into the learning process, guiding the agent toward tag sequences that are globally coherent rather than locally optimal but syntactically implausible.

7.4 Oracle Cost in Extended Environments

In the Oracle-augmented environment, invoking external supervision incurs a fixed negative reward. This cost is intentionally smaller in magnitude than the penalty for an incorrect autonomous prediction, encouraging the agent to request assistance only when uncertainty is high. By associating a tangible cost with Oracle usage, the reward function enables the agent to learn a principled trade-off between independence and reliability.

7.5 Design Rationale

Collectively, these reward components transform POS tagging from a sparse-reward problem into a structured decision process with strong inductive biases. The reward design aligns naturally with Offline Reinforcement Learning principles by discouraging out-of-distribution actions and stabilizing value estimation under fixed data conditions.

8 Experimental Setup

This section describes the experimental configuration used to evaluate the proposed Reinforcement Learning framework. The goal of the experiments is to assess both tagging performance and agent behavior under the two previously defined environments, while ensuring a controlled and reproducible evaluation protocol.

8.1 Training Protocol

All agents are trained exclusively in an Offline Reinforcement Learning setting. Transitions are extracted from the training split of the corpus and stored in a fixed replay buffer. During training, the agent samples mini-batches of transitions from this buffer to update its value estimates. No interaction with the environment occurs during training, and no additional data is generated beyond the annotated corpus.

Training is performed for a fixed number of optimization steps, and model selection is based on validation performance. Hyperparameters such as learning rate, discount factor, and network architecture are held constant across experiments unless explicitly stated otherwise.

8.2 Evaluation Procedure

Evaluation is conducted on a held-out test set that is not accessed during training. Each sentence in the evaluation set is processed sequentially, and the agent assigns a POS tag to each token according to its learned policy. In the Oracle-augmented environment, Oracle calls are permitted during inference according to the learned decision strategy.

The primary evaluation metric is overall POS tagging accuracy, computed as the proportion of correctly predicted tags over all tokens. In addition to accuracy, auxiliary metrics are collected to analyze agent behavior, including action distribution statistics and Oracle invocation frequency.

8.3 Compared Configurations

Multiple learning configurations are evaluated within each environment, including baseline value-based methods and enhanced offline variants incorporating conservative learning strategies. All configurations share the same dataset, preprocessing pipeline, and evaluation protocol, ensuring that observed differences are attributable to algorithmic and environmental factors rather than experimental inconsistencies.

9 Results and Analysis

This section presents the empirical results obtained from our experimental evaluation, structured into two distinct phases. Phase 1 (Experiment 1) evaluates the feasibility of training an RL agent to perform POS tagging "from scratch," while Phase 2 (Experiment 2) investigates a hybrid correction-based approach with an optional Oracle mechanism.

9.1 Experiment 1.1: Learning Syntax via Active Environmental Interaction

In the first experimental phase, we trained a Deep Q-Network (DQN) agent for 50,000 episodes to predict POS tags without leveraging a pre-trained base tagger, ultimately achieving a test accuracy of **60.13%**. Achieving convergence in this setting required extensive engineering of both the observation space and the reward function. We determined experimentally that the most effective state representation combined raw embeddings for the current and subsequent word (lookahead) with a set of explicit morphological and lexical features. Furthermore, to guide the agent through the vast combinatorial space, we implemented a strict reward shaping strategy: the agent received a reward of **+1** for correct predictions, **-1** for standard errors, and a severe penalty of **-5** for lexical violations (selecting a tag impossible for the given word). The objective was to determine whether an RL agent could learn valid syntactic rules

9.1.1 Learning Stability and Convergence

The learning dynamics in this high-dimensional state space proved significantly challenging. As illustrated in Figure 1 and Figure 2, the agent exhibits a slow and unstable learning curve. While the accumulated reward increases over time, the loss remains volatile, indicating the difficulty of exploring a discrete action space (12 tags) with dense, complex observation vectors.

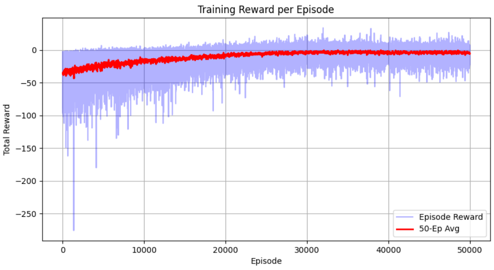


Figure 1: Experiment 1.1: Evolution of average reward per episode during training.

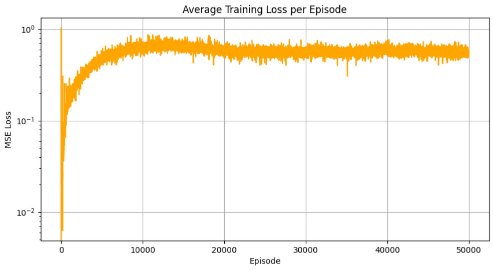


Figure 2: Experiment 1.1: Training loss over time showing volatility.

9.1.2 Performance relative to Baselines

The final accuracy of the "from-scratch" agent reached approximately **60.13%** on the test set, as shown in the training accuracy progression (Figure 3). While this performance significantly exceeds a random baseline (≈ 8.3%), it falls well short of uniform supervised baselines. To better understand the failure modes, we visualized the confusion matrix (Figure 4). The matrix reveals significant confusion between open-class words (e.g., NOUN vs. VERB) where distributional context is insufficient without pre-training.

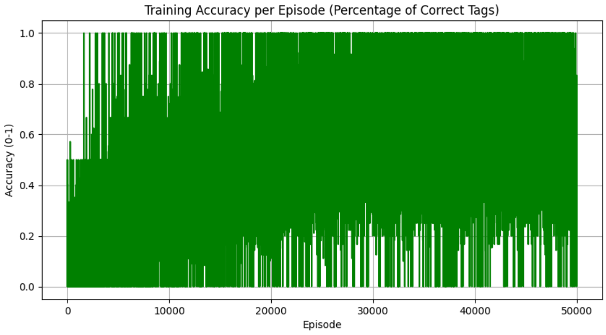
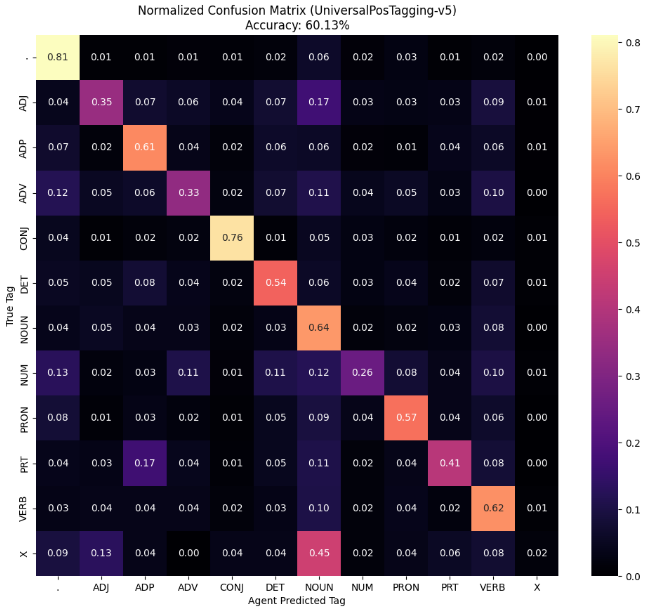


Figure 3: Experiment 1.1: Validation accuracy throughout the training process.



	.	ADJ	ADP	ADV	CONJ	DET	NOUN	NUM	PRON	PRT	VERB	X
.	0.81	0.01	0.01	0.01	0.01	0.02	0.06	0.02	0.03	0.01	0.02	0.00
ADJ	0.04	0.35	0.07	0.06	0.04	0.07	0.17	0.03	0.03	0.03	0.09	0.01
ADP	0.07	0.02	0.61	0.04	0.02	0.06	0.06	0.02	0.01	0.04	0.06	0.01
ADV	0.12	0.05	0.06	0.33	0.02	0.07	0.11	0.04	0.05	0.03	0.10	0.00
CONJ	0.04	0.01	0.02	0.02	0.76	0.01	0.05	0.03	0.02	0.01	0.02	0.01
DET	0.05	0.05	0.08	0.04	0.02	0.54	0.06	0.03	0.04	0.02	0.07	0.01
NOUN	0.04	0.05	0.04	0.03	0.02	0.03	0.64	0.02	0.02	0.03	0.08	0.00
NUM	0.13	0.02	0.03	0.11	0.01	0.11	0.12	0.26	0.08	0.04	0.10	0.01
PRON	0.08	0.01	0.03	0.02	0.01	0.05	0.09	0.04	0.57	0.04	0.06	0.00
PRT	0.04	0.03	0.17	0.04	0.01	0.05	0.11	0.02	0.04	0.41	0.08	0.00
VERB	0.03	0.04	0.04	0.04	0.02	0.03	0.10	0.02	0.04	0.02	0.62	0.01
X	0.09	0.13	0.04	0.00	0.04	0.04	0.45	0.02	0.04	0.06	0.08	0.02

Figure 4: Experiment 1.1: Confusion Matrix of the autonomous agent. Note the diagonal dominance but significant scattering for ambiguous classes like NOUN/VERB.

9.2 Experiment 1.2: Advanced Offline RL (CQL + BCQ)

In the second phase of autonomous tagging, we employed advanced Offline Reinforcement Learning techniques, specifically **Conservative Q-Learning (CQL)** combined with **Batch-Constrained Q-learning (BCQ)**. This approach addresses the instability observed in standard DQN by penalizing Q-values for out-of-distribution actions and constraining the policy to the support of the dataset.

9.2.1 Learning Stability and Convergence

The introduction of conservative constraints resulted in markedly more stable training dynamics. As shown in Figure 5, the offline agent (DQN + BCQ + CQL) demonstrates a steady increase in episode rewards and tagging accuracy, with the loss decreasing consistently compared to the volatile baseline in Experiment 1.1.

9.2.2 Performance Improvement

The enhanced offline formulation, further augmented by Prioritized Experience Replay (PER), yielded a substantial performance leap, achieving a final test accuracy of **93.99%**. This result is competitive with supervised baselines and represents a major improvement over the $\approx 60\%$ accuracy of the unconstrained agent.

Figure 6 details the recall per tag, highlighting that the model achieves near-perfect recognition for closed classes (DET, PRON, .) and robust performance on open classes (NOUN, VERB). The corresponding confusion matrix (Figure 7) confirms the reduction in off-diagonal errors. Notably, the introduction of **Batch-Constrained Q-learning (BCQ)** was the primary driver of the $\approx 30\%$ accuracy leap by effectively eliminating out-of-distribution actions, while the conservative loss function (CQL) further refined the policy to reach the final **93.99%** performance benchmark.

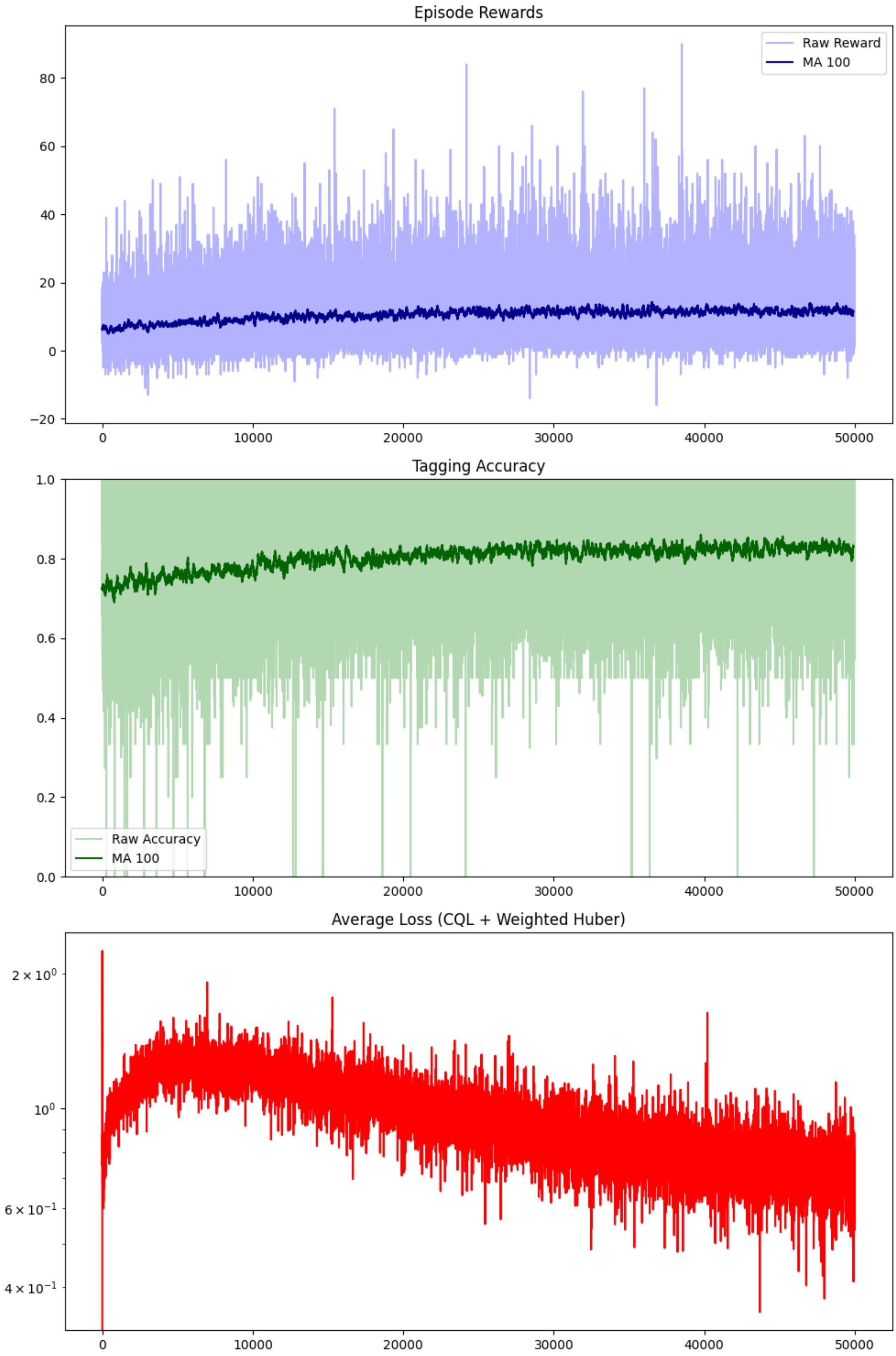


Figure 5: Experiment 1.2: Training dynamics for the CQL + BCQ agent. Top: Episode rewards show consistent improvement. Middle: Validation accuracy climbs steadily. Bottom: Loss convergence is smoother than the standard DQN baseline.

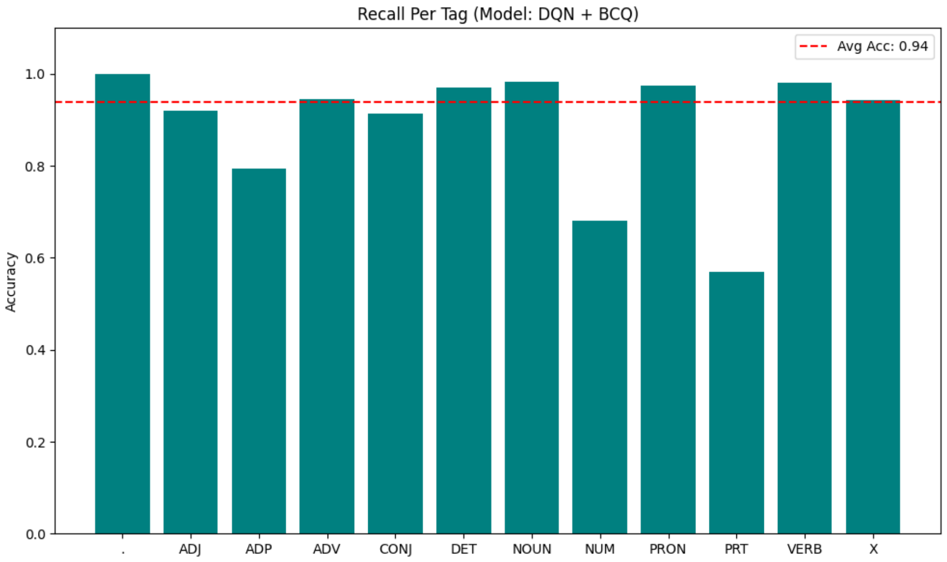


Figure 6: Experiment 1.2: Recall per POS tag. The Offline RL agent achieves >90% accuracy on most categories, with only function-specific ambiguities (e.g., PRT vs ADP) remaining.

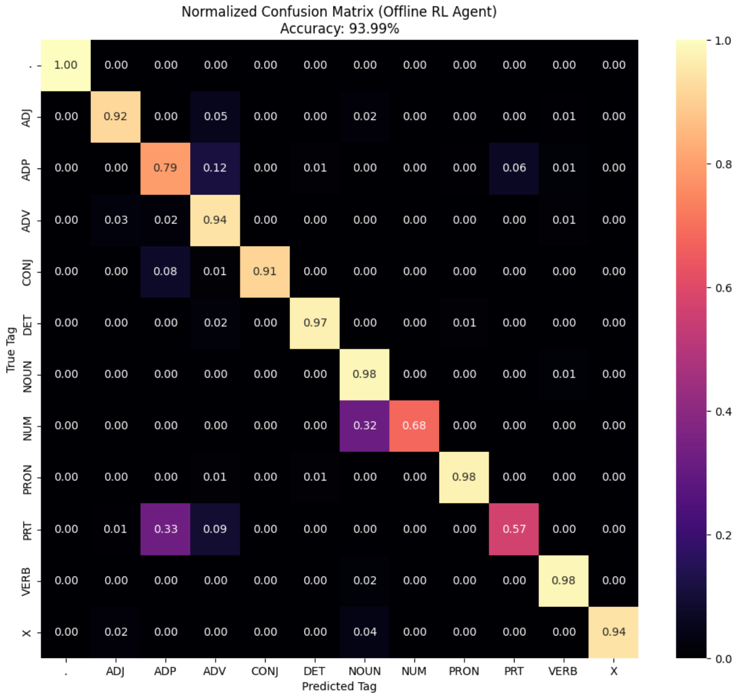


Figure 7: Experiment 1.2: Normalized Confusion Matrix. The diagonal structure is sharp, indicating that the agent has successfully internalized the syntactic rules of the language.

9.3 Experiment 2: Hybrid Correction Strategy

The second phase shifted the paradigm from *learning to tag* to *learning to correct.* Here, the RL agents (Q-Learning, DQN, REINFORCE) acted as meta-controllers over a fine-tuned DistilBERT base tagger, with the ability to **KEEP** existing predictions, **SHIFT** to the second-best model prediction, or consult an **Oracle (DICT)**.

9.3.1 Overall Accuracy Comparison

By leveraging the pre-trained knowledge of the base tagger, the RL agents achieved state-of-the-art performance for this specific dataset. As shown in Figure 8 and Table 1, both Q-Learning and DQN agents successfully improved upon the already high baseline of the DistilBERT model.

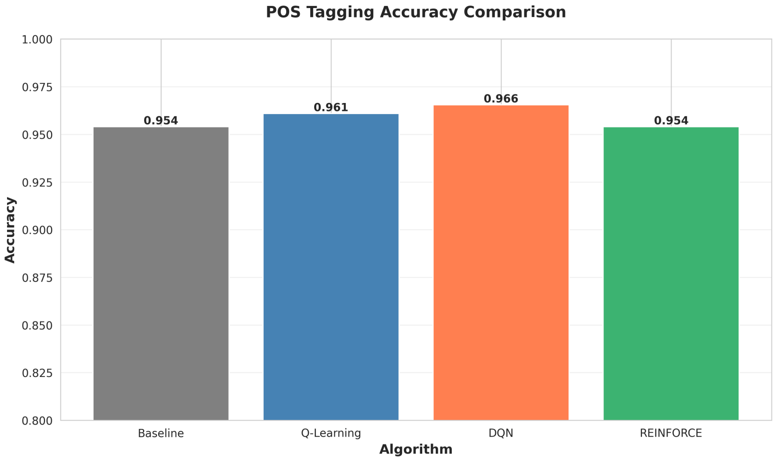


Figure 8: POS tagging accuracy comparison. The DQN agent achieves the highest performance, effectively correcting mistakes made by the base DistilBERT tagger.

Algorithm	Accuracy	Change	Strategy Profile
Base Tagger (DistilBERT)	95.4%	—	Static Prediction
DQN (Ours)	**96.6%**	**+1.20%**	**Strategic Correction**
Q-Learning	96.1%	+0.72%	Aggressive Correction
REINFORCE	95.4%	+0.00%	Passive (Risk Averse)

Table 1: Comparative accuracy of RL correction strategies against the supervised baseline on the evaluation set.

The **DQN agent** achieved the highest overall accuracy (96.6%), demonstrating that the inclusion of a value-based correction mechanism can correct edge cases that supervised models miss. Conversely, policy gradient methods (REINFORCE) failed to improve over the baseline, converging to a purely passive policy (always selecting KEEP) to avoid the penalty accumulation associated with exploration.

9.3.2 Oracle Efficiency Analysis

A critical contribution of this work is the analysis of *when* agents choose to use the costly Oracle (DICT) action. An intelligent agent should only pay the cost for an Oracle call when the base model is likely wrong.

We define **Necessity Rate** as the percentage of DICT calls where the base model actions would have been incorrect. A high necessity rate implies efficient resource usage.

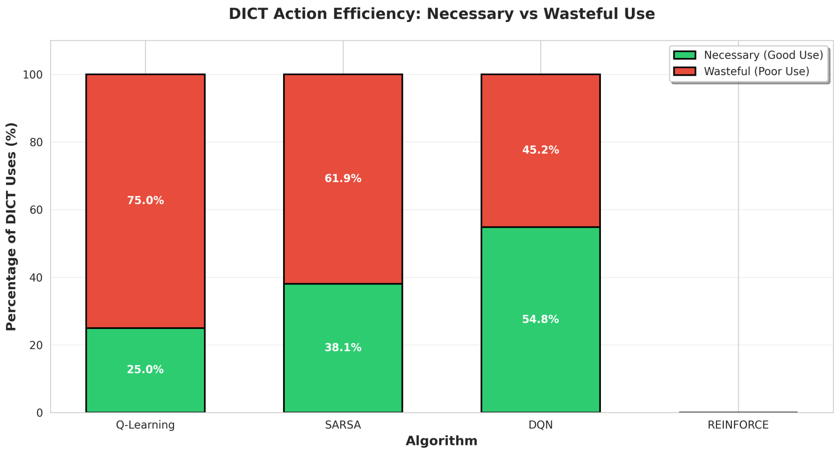


Figure 9: Necessity vs. Waste analysis for Oracle (DICT) usage. The DQN agent demonstrates superior efficiency, using the Oracle primarily when necessary (54.8% necessity), whereas Q-Learning frequently "wastes" budget on already correct tokens.

The empirical breakdown of Oracle usage is detailed in Table 2.

Agent	Total DICT Uses	Necessary (Valid Fixes)	Wasteful (Redundant)	Efficiency
Q-Learning	52	13 (25.0%)	39 (75.0%)	Low
DQN	**31**	**17 (54.8%)**	**14 (45.2%)**	**High**
REINFORCE	0	0 (0.0%)	0 (0.0%)	N/A

Table 2: Detailed statistics on Oracle usage. The DQN agent learns a more calibrated uncertainty estimate, reducing wasteful calls by nearly half compared to tabular Q-Learning.

These results suggest that the deep reinforcement learning approach (DQN) generalizes better to the continuous state representation (embeddings + confidence scores), allowing it to identify "risky" tokens more accurately than the tabular approach, which suffers from discretization errors.

Oracle usage statistics indicate controlled and selective intervention rather than over-reliance, as illustrated in Figure 10.

10 Error Analysis and Limitations

Despite the overall effectiveness of the proposed framework, several limitations and sources of error remain. These limitations arise from both the intrinsic properties of natural language and the design choices imposed by the Reinforcement Learning formulation.

10.1 Sequential Processing Limitations

The left-to-right processing constraint prevents the agent from revising earlier decisions based on future context. Although limited look-ahead information is incorporated in some configurations, long-range dependencies and global sentence structure remain challenging. As a result, certain syntactic ambiguities cannot be resolved reliably within the current framework.

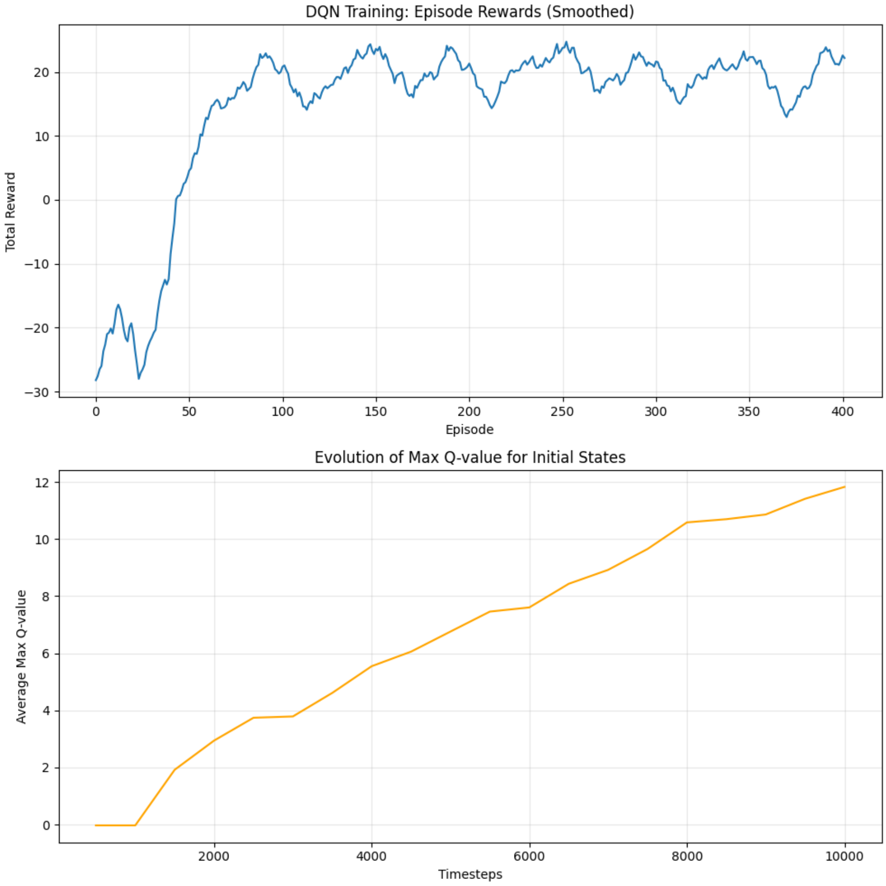


Figure 10: Frequency and efficiency of DICT (Oracle) action usage.

10.2 Ambiguity-Driven Errors

A significant portion of residual errors is attributable to genuine linguistic ambiguity. Words with multiple syntactic roles may remain ambiguous even under local contextual cues. In such cases, incorrect predictions do not necessarily reflect model deficiencies but rather limitations of the available information.

10.3 Credit Assignment Challenges

A significant limitation lies in the nature of the feedback signal relative to supervised learning. In the RL formulation, optimizing the cumulative return can obscure specific errors, as the global reward at the end of the episode lacks individual word context. Consequently, the agent may struggle to identify exactly which particular word caused a drop in performance ("which word got wrong"), making it difficult to correct specific local errors without highly engineered, dense reward signals.

11 Discussion

This section discusses the implications of the experimental findings and situates the proposed approach within the broader context of Reinforcement Learning and Natural Language Processing. Rather than focusing on raw performance alone, the discussion emphasizes qualitative behavior, modeling choices, and practical relevance.

A central observation of this work is that framing POS tagging as a sequential decision-making problem enables explicit control over linguistic constraints and decision dependencies. Unlike static supervised models, the Reinforcement Learning formulation exposes the internal decision process, making it possible to analyze how individual tagging decisions influence subsequent predictions.

The separation between the standard and Oracle-augmented environments highlights two complementary operating regimes. Fully autonomous tagging encourages the agent to internalize syntactic regularities and rely on learned value estimates, while the Oracle-augmented setting introduces a mechanism for managing uncertainty in ambiguous cases. Importantly, the Oracle is not used as a corrective post-processing step, but as an integrated decision option whose cost is explicitly modeled within the learning objective.

From a practical perspective, the results suggest that constraint-aware Offline Reinforcement Learning can achieve competitive performance on structured linguistic tasks without requiring large-scale pretrained models. Although the proposed approach does not aim to outperform state-of-the-art Transformer-based systems, it offers advantages in interpretability, modularity, and the ability to incorporate external decision logic.

Overall, the framework demonstrates that Reinforcement Learning serves as a potent optimization tool suitable for refining high-performance supervised baselines and flagging uncertain situations where human intervention is beneficial. Simultaneously, it provides a flexible modeling paradigm for structured prediction, where constraints, uncertainty, and interaction play a central role.

12 Future Work

The proposed Reinforcement Learning framework opens several directions for future research and extension. While the current study focuses on Part-of-Speech tagging, the underlying principles are applicable to a broader class of structured prediction problems in Natural Language Processing.

One natural extension involves incorporating richer contextual representations. While the current formulation relies on local context and limited look-ahead, future work could integrate recurrent or attention-based components into the state representation. Such extensions would allow the agent to capture longer-range dependencies while preserving the sequential decision-making structure.

Another promising direction is the integration of pretrained language models. Contextual embeddings produced by Transformer-based encoders could serve as high-quality state inputs, while Reinforcement Learning would remain responsible for decision-making under constraints and uncertainty. This hybrid approach could combine the strengths of large pretrained models with the flexibility of RL-based control.

The Oracle mechanism itself can also be extended. More sophisticated uncertainty estimation techniques, such as ensemble methods or Bayesian value estimation, could provide more reliable signals for triggering external intervention. Additionally, the Oracle could be generalized beyond ground-truth access to include expert heuristics or secondary models with higher computational cost.

Finally, the framework can be applied to other sequence labeling tasks, including named entity recognition, chunking, or shallow parsing. Exploring these applications would help assess the generality of the approach and further clarify the role of Reinforcement Learning in structured NLP tasks.

13 Conclusions

This work presented a Reinforcement Learning-based approach to Part-of-Speech tagging, reformulating a classical Natural Language Processing task as a sequential decision-making problem. By modeling tagging as a Markov Decision Process, the proposed framework enables explicit representation of action dependencies, structured constraints, and uncertainty-aware decision strategies.

The study demonstrated that naive Online Reinforcement Learning is ill-suited for high-dimensional linguistic domains, motivating the adoption of an Offline Reinforcement Learning paradigm. Through the use of lexical constraints, grammatical biases, and conservative learning strategies, stable training becomes feasible even in large discrete action spaces.

A key contribution of this work is the explicit separation and analysis of two environments: a standard autonomous tagging environment and an Oracle-augmented environment that incorporates Human-in-the-Loop decision support. This separation allows a clear examination of the trade-offs between autonomy and reliability and avoids ambiguity in experimental interpretation.

While the proposed approach does not seek to replace state-of-the-art supervised models, it demonstrates that Reinforcement Learning can provide a viable and flexible alternative for structured prediction tasks. Beyond POS tagging, the framework offers a foundation for integrating constraints, interaction, and uncertainty into sequential NLP systems.