Comparative Study of Decoding Techniques and Adversarial Training in Hidden Markov Model-Based Part-of-Speech Tagging

Kayvan Shah
University of Southern California
kpshah@usc.edu

Abstract

This document explores the utilization and re-implementation of Hidden Markov Models (HMMs) to understand the sequential structure inherent in language, specifically associating concealed states, such as Part-of-Speech (POS) tags, with observable words (Manning, 1992). HMMs serve as a foundational framework that captures linguistic sequences' underlying grammar and syntax. Focused on both theory and practice, this project emphasizes the hands-on implementation of HMMs for Part-of-Speech (POS) tagging tasks. It includes thoroughly comparing and re-implementing decoding techniques like the Viterbi algorithm and greedy decoding within the HMM framework. Additionally, the study delves into augmenting model accuracy and robustness by exploring adversarial training strategies introducing perturbations during training to enhance resilience in sequence labeling tasks.

1 Introduction

The quest for structurally parsing language led to the inception of Hidden Markov Models (HMMs) as a cornerstone in Part-of-Speech (POS) tagging (Manning, 1992). HMMs provided a probabilistic framework to associate hidden states—POS tags—with observable words, pioneering the modeling of sequential linguistic data.

While HMMs exhibit an accuracy range of 85-90% (Manning, 1992), recent advancements in the field of Natural Language Processing have witnessed the emergence of more sophisticated models achieving scores surpassing 95%, notably when trained on extensive and diverse linguistic datasets (Lample et al., 2016; Devlin et al., 2018). These advancements propel the quest for enhancing accuracy and robustness in sequence labeling tasks.

Traditionally, HMMs formed the bedrock of POS tagging methodologies, utilizing algorithms like the Baum-Welch algorithm for parameter estimation (Jurafsky and Martin, 2000; Manning and Schütze, 1999). Perturbations or adversarial training strategies were not conventionally integrated into the standard training regimen of HMMs for POS tagging.

However, the landscape of POS tagging has evolved significantly with the rise of more intricate models (Lample et al., 2016; Devlin et al., 2018). These models, leveraging neural architectures and transformer-based approaches, have set new benchmarks in accuracy and performance, prompting a relentless pursuit for heightened accuracy and robustness in POS tagging tasks.

In the wake of these advancements, this study revisits the efficacy of traditional HMMs in the realm of POS tagging (Jurafsky and Martin, 2000; Manning and Schütze, 1999). While historically devoid of perturbation-based strategies, this research investigates potential adaptations or extensions to HMMs, exploring the integration of perturbations or adversarial techniques to fortify model accuracy and resilience in the nuanced domain of POS tagging (Jurafsky and Martin, 2000; Manning and Schütze, 1999).

2 Methodology

Our methodology aims to enhance our sequence labeling model by re-implementing core components like the Hidden Markov Model (HMM) and Greedy and Viterbi decoders (Manning, 1992; Jurafsky and Martin, 2000). We also integrate adversarial-based perturbation techniques, including controlled noise addition, limited perturbation strategies, and adversarial training with clean and perturbed data samples. These methods assess the model's resilience to data variations, fortifying its adaptability and optimizing sequence generation for reduced errors.

The code-base is made available on GitHub[1].

[1] https://github.com/KayvanShah1/pos-tagging-hmm-adversarial-learning

2.1 Re-implemention

2.1.1 Hidden Markov Models

This process encompassed the construction of three crucial matrices: Prior probabilities, Transition probabilities, and Emission probabilities, using the Wall Street Journal Dataset (Marcus et al., 1993; Jurafsky and Martin, 2000), specifically the PENN Treebank.

$$t(s'|s) = \frac{\text{count}(s \to s')}{\text{count}(s)}$$

$$e(x|s) = \frac{\text{count}(s \to x)}{\text{count}(s)}$$

$$\pi(s) = \frac{\text{count}(\text{null} \to s)}{\text{count}(\text{num\_sentences})}$$

Where t is the transition parameter, e is the emission parameter, and π represents the initial state or prior probabilities. In this context, s and s' denote states in the model, x represents observations, and *num_sentences* denotes the total number of sentences.

2.1.2 Greedy and Viterbi Decoder

Two decoding techniques, Greedy Decoding and Viterbi Decoding, were implemented to extract the most probable sequence of hidden states from observed data. The Greedy Decoding technique makes locally optimal decisions at each step (Rabiner, 1989). Viterbi Decoding aims to find the globally optimal sequence by considering the entire observation sequence and associated probabilities (Viterbi, 1967; Forney, 1973).

Greedy Decoding:

for each observation **do**

Choose the most probable state based on local probabilities at each step.

Update the sequence based on these decisions.

end for

Viterbi Decoding:

Initialize trellis with probabilities.

for each observation **do**

Calculate probabilities for each state and the most probable path.

Update the sequence considering global probabilities and transition rules.

end for

2.2 Adversarial Perturbation Strategies in Hidden Markov Models

Adversarial strategies in HMMs for POS tagging involve variations such as Laplace smoothing with varying constants, controlled noise addition to emission probabilities, and random perturbations within selected ranges to assess model robustness and sensitivity (Mitrophanov et al., 2005).

Laplace Smoothing with Variation: Apply Laplace smoothing with varying smoothing constants to transition and emission probabilities. This experiment aims to assess the impact of different levels of smoothing on the model's performance in Part-of-Speech (POS) tagging tasks (Johnson, 2007).

Noise Addition: Introduce controlled noise to the emission probabilities matrix. This test evaluates the model's resilience to noise-induced disturbances in input data, simulating real-world scenarios.

Random Ranged Perturbations: Introduce variations in probabilities by modifying them slightly within a selected range. This experiment aims to gauge the model's sensitivity to subtle variations without significantly altering the structure.

3 Experiments

3.1 Dataset

We have utilized the Wall Street Journal dataset, specifically utilizing the POS tagging dataset from the PENN Treebank. This corpus consists of text excerpts from the Wall Street Journal (Marcus et al., 1993), widely recognized for its comprehensive coverage of various domains, making it a valuable resource for Natural Language Processing (NLP) tasks, particularly in Part-of-Speech (POS) tagging. The dataset utilized in this study was a sampled version of the PENN TREE bank dataset, consisting of 38,218 samples in the training set and 5,527 samples in the validation set.

3.2 Data Processing

Our sequence-to-sequence labeling process begins with meticulous data preparation steps. This includes ensuring case insensitivity, generating the vocabulary, mapping words to indices, and enhancing sequences with specialized tokens for context. A threshold of 2 was applied to identify tokens as unknown for creating the vocabulary.

3.3 Implementation

3.3.1 HMMs

Calculating the transition, emission, and prior probabilities involves initializing the HMM with parameters and creating matrices. Constants, N (distinct tags) and M (observable symbols), are determined from the data. This includes $N \times N$ transition probabilities, $N \times M$ emission probabilities, and $N \times 1$ prior probabilities. Mapping dictionaries ensures tag and word association consistency.

Computing Priors: Iterate over sentences, extract first tags and indices, and calculate occurrences. Normalize occurrences by total sentences for tag priors.

Transition Probabilities: Extract label indices from sentences, increment corresponding transition matrix entries, and normalize.

Emission Probabilities: Iterate over sentences, words, and labels to compute emission probabilities. Increment emission matrix entries and normalize.

3.3.2 Greedy Decoder

Determines the most probable tag for each word, emphasizing maximizing likelihood at each step independently. It begins decoding from the first word, selecting one state at a time without considering the overall sequence context. Inputs are those that are generated after training an HMM.

Precomputing scores: Precompute scores for word-tag pairs by multiplying prior probabilities with emission probabilities.

Decoding: Then, iterate through sentences, computing scores for each word based on priors and emissions. For subsequent words, combine transition probabilities from the previous state with emission probabilities for the current word. Predict the tag with the highest combined score for each word and store it for the next iteration.

3.3.3 Viterbi Decoder

The algorithm uses dynamic programming to break sequences into smaller subproblems, exploring all potential paths to identify the most probable sequence. It employs a trellis structure to pinpoint the optimal sequence of hidden states and utilizes a backtracking mechanism to retrace the path of maximum probabilities for sequence identification. Inputs are those that are generated after training an HMM.

Precomputing scores: The Viterbi decoding process commences by precomputing scores for word-tag pairs, amalgamating prior probabilities with emission probabilities. Essential variables such as the Viterbi and Path matrices are initialized before decoding.

*Decoding:*During the decoding phase, it starts by computing the score for the initial word using the precomputed matrices. It proceeds by iterating through subsequent time steps, beginning from the second word in the sentence. At each time step, scores are calculated based on prior Viterbi values, transition, and emission probabilities. The Viterbi and Path matrices are updated according to the maximum scores obtained.

Backtracking: Following the decoding steps, the process entails backtracking from the last word to derive the best sequence of tags. This backtracking mechanism iterates backward, selecting the tag with the highest score at each step to determine the optimal sequence.

3.3.4 Adversarial Strategies

Perturbations are introduced in the following ways:

Laplace Smoothing: To handle zero probabilities, apply Laplace smoothing by adding a small constant (1×10^{-10}) to all probabilities. This prevents the issues caused by zero probabilities in the calculations.

Noise Addition to Transition/Emission Params: Introduces deliberate variations to emission probabilities by tweaking them with small, predetermined increments, strategically altering probabilities.

Random Ranged Perturbations: Generates controlled noise within predefined ranges, subtly adjusting transition or emission parameters to simulate slight disturbances without major structural alterations (Drive5, 2021).

3.3.5 Evaluation

When evaluating the performance of the sequence labeling model, a tag-by-tag comparison is conducted between the true sequences (ground truth) and the sequences predicted by the model. This comparison involves assessing each individual tag within the predicted sequence against the corresponding tag in the true sequence (Manning and Schütze, 1999). The model-predicted tag matches the actual tag from the reference or labeled data for every position in the sequence.

3.4 Implementation differences

Greedy Decoding: Our Greedy Decoding implementation differs by utilizing direct word-tag scores precomputed from prior and emission probabilities. This approach bypasses iterative multiplication, enhancing efficiency in the decoding process. Additionally, the algorithm simplifies sequence iteration and tag selection, reducing computational complexity and streamlining the decoding workflow.

Viterbi Decoding: Our Viterbi decoding implementation optimizes efficiency by employing precomputed scores, enhancing initialization, and streamlining backtracking techniques. Leveraging precomputed word-tag scores and optimized memory usage it significantly accelerates decoding speed compared to traditional methods, striking a balance between memory utilization and computational efficiency. Furthermore, it showcases streamlined backtracking operations and leverages NumPy for vectorized operations, enhancing overall efficiency and performance.

4 Results & Analysis

In this section, we present the results of our experiments as a table (Table 1).

4.1 Results

Strategies	Greedy Decoding	Viterbi Decoding
Standard	0.9156	0.9321
Laplace Smoothing	0.9155	0.9323
Noise Addition to Transition Params	0.9124	0.0907
Random Ranged Perturbations	0.7031	0.0945

Table 1: Accuracy Comparison Using Different Adversarial Perturbation-based Strategies

1. Comparable Performance of Default and Laplace Smoothing: Using default and Laplace smoothing strategies demonstrates remarkably similar accuracies in Part-of-Speech (POS) tagging. Despite their distinct approaches to handling unseen or rare events, both methods yield comparable results, showcasing their proximity in accuracy levels.

2. Viterbi Algorithm Superiority over Greedy Decoding with Standard and Laplace Smoothing: The Viterbi algorithm consistently outperforms the greedy decoding method in evaluations with both standard and Laplace smoothing strategies. This superiority highlights the Viterbi algorithm's ability to capitalize on the sequential dependencies within the data, enhancing accuracy in POS tagging tasks compared to the more simplistic greedy approach (Mitrophanov et al., 2005).

3. Greedy Decoding's Advantage in the Presence of Perturbations: Intriguingly, when perturbations are introduced, the tables turn, and the greedy decoding algorithm showcases superior performance over the Viterbi algorithm. The simplicity and robustness of the greedy approach appear to mitigate the disruptive effects of perturbations, resulting in better accuracy in the presence of these perturbations.

4. Impact of Perturbations on Accuracy: Introducing perturbations induces a significant drop in accuracy across both decoding methods. This decline underscores the vulnerability of the models to subtle variations or noise in the input data, emphasizing the need for enhanced robustness strategies in handling such adversarial scenarios.

4.2 Analysis

The analysis provided below delves into the interpretation of the obtained results and the comparison of decoding algorithms.

Greedy Algorithm: The Greedy algorithm demonstrates exceptional speed in processing. It exhibits rapid and independent decision-making per token, enabling quick results. However, its speed efficiency may lead to suboptimal outcomes due to its localized decision-making process, sacrificing global optimality for speed (Manning et al., 2008; Manning and Schütze, 1999).

Viterbi Algorithm: In contrast, the Viterbi algorithm focuses on attaining optimal results by considering the entire observation sequence and its associated probabilities (Manning et al., 2008). This approach ensures the best possible tags for a given sequence of words. However, its pursuit of optimality involves higher computational complexity than the Greedy algorithm.

Disruptive Nature of Random Perturbations: Random perturbations within Hidden Markov Models (HMMs) inherently disturb learned patterns within the model. Particularly impactful on emission probabilities, these disruptions significantly distort the decoding process, introducing disturbances that affect the model's integrity and accuracy.

Viterbi Decoding Sensitivity to Perturbations: The Viterbi decoding process demonstrates high sensitivity to changes in emission probabilities induced by perturbations. This heightened sensitivity results in a drastic drop in accuracy, affecting the model's ability to maintain the sequential integrity of the output (Mitrophanov et al., 2005).

5 Conclusion

In conclusion, this project explored various adversarial perturbation strategies within Hidden Markov Models (HMMs) for Part-of-Speech (POS) tagging. The study aimed to bolster the model's robustness and adaptability through experiments employing techniques such as Laplace smoothing variations, controlled noise addition, and focused perturbations. Findings revealed that while adversarial training incorporating perturbed data improved the model's resilience, limiting perturbation strategies safeguarded crucial learned patterns without compromising adaptability. Focused perturbation strategies proved effective in refining the model's handling of ambiguous observations. Yet, it's crucial to acknowledge that the efficacy of these strategies is context-dependent, and their implementation requires a careful balance between adaptation and preserving core learning. However, due to time constraints, not all objectives were fully realized. While the decoders and Hidden Markov Model were successfully re-implemented, the scope for integrating various adversarial strategies was limited, and only a few could be implemented within the stipulated timeframe. Future research could delve deeper into optimizing these strategies for specific linguistic contexts or expanding their application to other sequence modeling tasks, thereby advancing the field's understanding of adversarial robustness in NLP.

6 Future Work

Incorporating adversarial training techniques involves exposing the model to a combination of clean and perturbed data during the training phase, aiming to fortify its adaptability to variations and uncertainties in real-world input data sources. To mitigate potential disruptions to the model's core learned patterns, a strategy of limited perturbation is employed, confining disturbances to a subset of the emission matrix or less critical probabilities (Kurakin et al., 2016). This controlled approach allows the model to adjust to variations while preserving essential information. Additionally, employing a focused perturbation strategy targets specific subsets of emission probabilities, directing perturbations to infrequent or ambiguous observations. This strategy seeks to refine the model's handling of challenging data instances, potentially enhancing its accuracy in handling uncertain or less frequent data points.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Drive5. 2021. Hidden markov model perturbations. https://drive5.com/muscle5/manual/hmm_perturbations.html.

George D Forney, Jr. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Richard Johnson. 2007. *Introduction to Probability and Statistics*, 2nd edition edition. Wiley. Chapter 5: Smoothing Techniques.

Daniel Jurafsky and James H Martin. 2000. *Speech and Language Processing*. Prentice Hall.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. 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*.

Christopher D Manning. 1992. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Proceedings of the 12th international conference on computational linguistics - Volume 1*. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. In *Computational linguistics*, volume 19, pages 313–330. MIT Press.

Alexander Yu. Mitrophanov et al. 2005. Sensitivity of hidden markov models. *Journal of Applied Probability*, 42(3):632–642.

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Andrew J Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transaction on Information Theory*, 13(2):260–269.

Contributions

- Re-implementation on HMMs
- Re-implementation on Greedy Decoder
- Re-implementation on Viterbi Decoder
- Implementing adversarial perturbation-based strategies
- Data Processing and Inference Pipeline
- Project presentation
- Final report

Acknowledgement

I acknowledge the guidance received from the course staff of DSCI 599, Fall 2023, at the University of Southern California.