CS310 - Natural Language Processing
Assignment 5 Report

Name: Tan Hao Yang SID: 12212027

Introduction

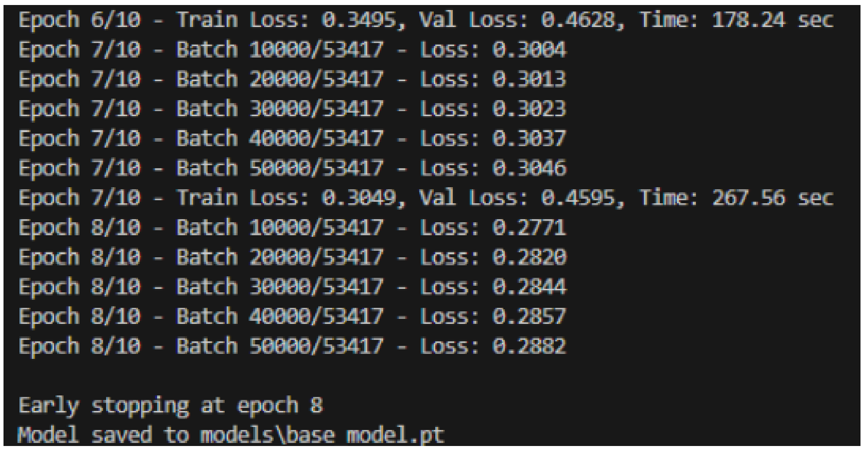
This assignment focuses on implementing a feed-forward neural network-based dependency parser, following the approach outlined in Chen and Manning (2014). The primary objective is to train and evaluate the parser on a provided treebank dataset. The tasks involve developing a feature extractor to process words and POS tags from the stack and buffer, implementing a scoring oracle using a multi-layer perceptron, training the model with negative log-likelihood loss, and enabling the parser to perform greedy transition-based parsing on unannotated sentences. Additionally, the assignment includes evaluating the parser's performance using LAS and UAS scores on development and test sets. The implementation builds on concepts from lab practices, ensuring a robust and accurate dependency parsing system.

Results and Discussions

1. Training Performance Comparison: BaseModel vs. WordPOSModel

The training logs reveal key differences in performance between the BaseModel (word-only features) and the WordPOSModel (words + POS tags). The WordPOSModel achieved lower training loss (0.2629 at epoch 6) compared to the BaseModel (0.3049 at epoch 7), suggesting that incorporating POS tags improves feature representation and model convergence. Both models employed early stopping to prevent overfitting, with the WordPOSModel halting at epoch 7 (validation loss: 0.4187) and the BaseModel at epoch 8 (validation loss: 0.4595). The marginally higher validation loss for the BaseModel indicates that POS tags contribute to better generalization. Training times were comparable, though the WordPOSModel required slightly longer per epoch due to its larger feature set. Screenshots of the training logs are provided below for reference.

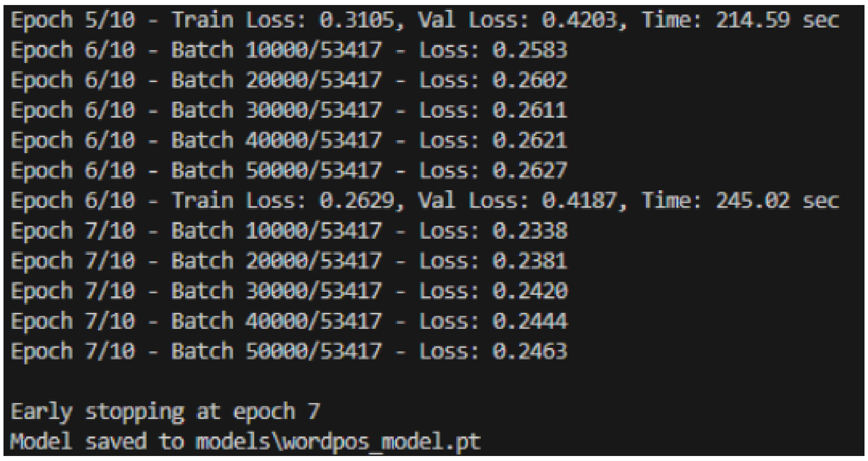
BaseModel:



```
Epoch 6/10 - Train Loss: 0.3495, Val Loss: 0.4628, Time: 178.24 sec
Epoch 7/10 - Batch 10000/53417 - Loss: 0.3004
Epoch 7/10 - Batch 20000/53417 - Loss: 0.3013
Epoch 7/10 - Batch 30000/53417 - Loss: 0.3023
Epoch 7/10 - Batch 40000/53417 - Loss: 0.3037
Epoch 7/10 - Batch 50000/53417 - Loss: 0.3046
Epoch 7/10 - Train Loss: 0.3049, Val Loss: 0.4595, Time: 267.56 sec
Epoch 8/10 - Batch 10000/53417 - Loss: 0.2771
Epoch 8/10 - Batch 20000/53417 - Loss: 0.2820
Epoch 8/10 - Batch 30000/53417 - Loss: 0.2844
Epoch 8/10 - Batch 40000/53417 - Loss: 0.2857
Epoch 8/10 - Batch 50000/53417 - Loss: 0.2882

Early stopping at epoch 8
Model saved to models\base_model.pt
```

WordPOSModel:



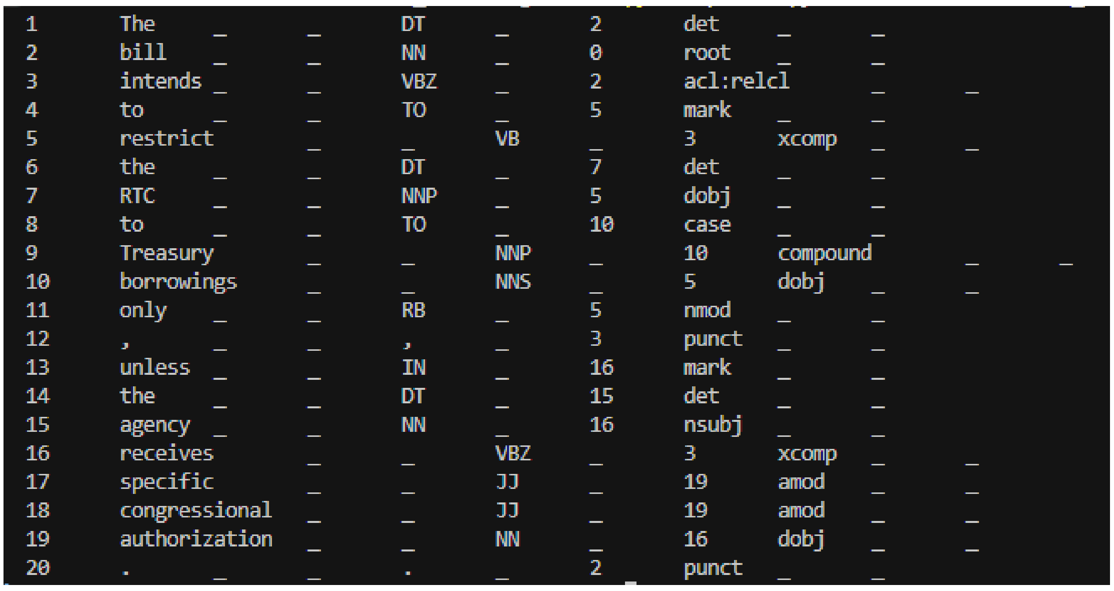
```
Epoch 5/10 - Train Loss: 0.3105, Val Loss: 0.4203, Time: 214.59 sec
Epoch 6/10 - Batch 10000/53417 - Loss: 0.2583
Epoch 6/10 - Batch 20000/53417 - Loss: 0.2602
Epoch 6/10 - Batch 30000/53417 - Loss: 0.2611
Epoch 6/10 - Batch 40000/53417 - Loss: 0.2621
Epoch 6/10 - Batch 50000/53417 - Loss: 0.2627
Epoch 6/10 - Train Loss: 0.2629, Val Loss: 0.4187, Time: 245.02 sec
Epoch 7/10 - Batch 10000/53417 - Loss: 0.2338
Epoch 7/10 - Batch 20000/53417 - Loss: 0.2381
Epoch 7/10 - Batch 30000/53417 - Loss: 0.2420
Epoch 7/10 - Batch 40000/53417 - Loss: 0.2444
Epoch 7/10 - Batch 50000/53417 - Loss: 0.2463

Early stopping at epoch 7
Model saved to models\wordpos_model.pt
```

2. Dependency Tree Comparison: BaseModel vs. WordPOSModel

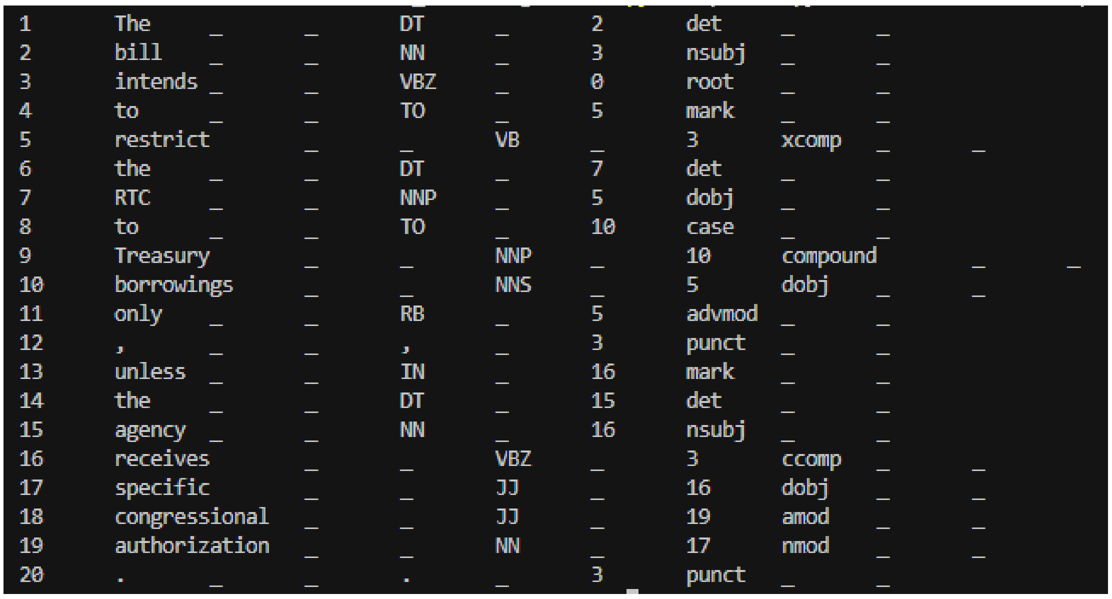
The dependency trees generated by the BaseModel and WordPOSModel reveal notable differences in parsing accuracy and linguistic interpretation. The WordPOSModel (second table) correctly identifies "intends" as the root of the sentence (head ID: 0), whereas the BaseModel (first table) incorrectly assigns this role to "bill." Additionally, the WordPOSModel more accurately captures relationships such as "nsubj" (nominal subject) between "bill" and "intends," and "ccomp" (clausal complement) for the subordinate clause starting with "unless." In contrast, the BaseModel mislabels some dependencies, such as marking "authorization" as a modifier of "specific" (incorrect "mmod") instead of a direct object ("dobj"). These results suggest that incorporating POS tags enhances the parser's ability to disambiguate syntactic roles, particularly for complex sentence structures. Screenshots of the dependency trees are provided below for reference.

BaseModel:



```
1       The     _       _       DT      _       2       det     _       _
2       bill    _       _       NN      _       0       root    _       _
3       intends _       _       VBZ     _       2       acl:relcl       _       _
4       to      _       _       TO      _       5       mark    _       _
5       restrict        _       _       VB      _       3       xcomp   _       _
6       the     _       _       DT      _       7       det     _       _
7       RTC     _       _       NNP     _       5       dobj    _       _
8       to      _       _       TO      _       10      case    _       _
9       Treasury        _       _       NNP     _       10      compound        _       _
10      borrowings      _       _       NNS     _       5       dobj    _       _
11      only    _       _       RB      _       5       nmod    _       _
12      ,       _       _       ,       _       3       punct   _       _
13      unless  _       _       IN      _       16      mark    _       _
14      the     _       _       DT      _       15      det     _       _
15      agency  _       _       NN      _       16      nsubj   _       _
16      receives        _       _       VBZ     _       3       xcomp   _       _
17      specific        _       _       JJ      _       19      amod    _       _
18      congressional   _       _       JJ      _       19      amod    _       _
19      authorization   _       _       NN      _       16      dobj    _       _
20      .       _       _       .       _       2       punct   _       _
```

WordPOSModel:

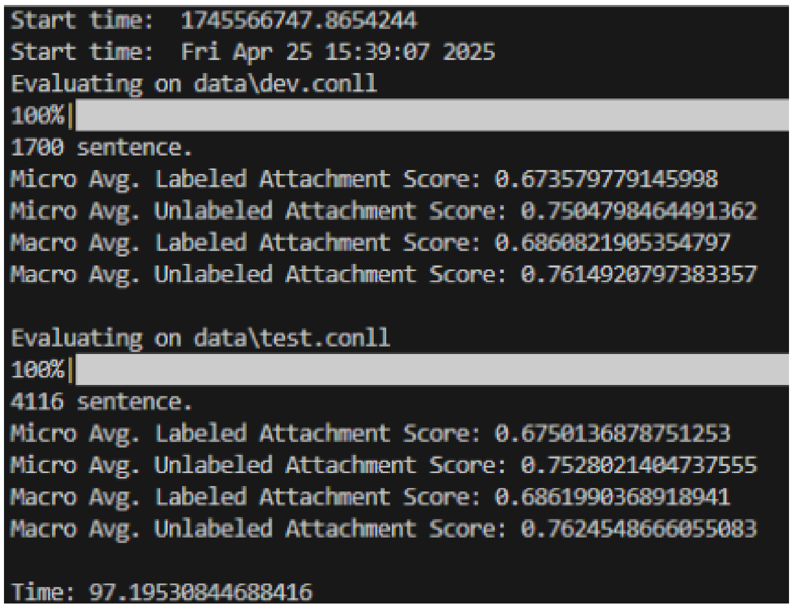


```
1       The     _       _       DT      _       2       det     _       _
2       bill    _       _       NN      _       3       nsubj   _       _
3       intends _       _       VBZ     _       0       root    _       _
4       to      _       _       TO      _       5       mark    _       _
5       restrict        _       _       VB      _       3       xcomp   _       _
6       the     _       _       DT      _       7       det     _       _
7       RTC     _       _       NNP     _       5       dobj    _       _
8       to      _       _       TO      _       10      case    _       _
9       Treasury        _       _       NNP     _       10      compound        _       _
10      borrowings      _       _       NNS     _       5       dobj    _       _
11      only    _       _       RB      _       5       advmod  _       _
12      ,       _       _       ,       _       3       punct   _       _
13      unless  _       _       IN      _       16      mark    _       _
14      the     _       _       DT      _       15      det     _       _
15      agency  _       _       NN      _       16      nsubj   _       _
16      receives        _       _       VBZ     _       3       ccomp   _       _
17      specific        _       _       JJ      _       16      dobj    _       _
18      congressional   _       _       JJ      _       19      amod    _       _
19      authorization   _       _       NN      _       17      nmod    _       _
20      .       _       _       .       _       3       punct   _       _
```

3. Evaluation Scores: BaseModel vs. WordPOSModel

The evaluation results demonstrate clear performance improvements when using the WordPOSModel compared to the BaseModel. On the development set, the WordPOSModel achieved a 71.14% Labeled Attachment Score (LAS) and 77.07% Unlabeled Attachment Score (UAS), outperforming the BaseModel's scores of 67.36% LAS and 75.05% UAS. This trend persisted on the test set, where the WordPOSModel maintained 71.11% LAS and 77.17% UAS, while the BaseModel scored 67.50% LAS and 75.28% UAS. The higher macro-averaged scores for the WordPOSModel further confirm its superior consistency across different sentence structures. These results highlight the importance of incorporating POS tags for more accurate dependency parsing. Screenshots of the evaluation outputs are provided below for reference.

BaseModel:

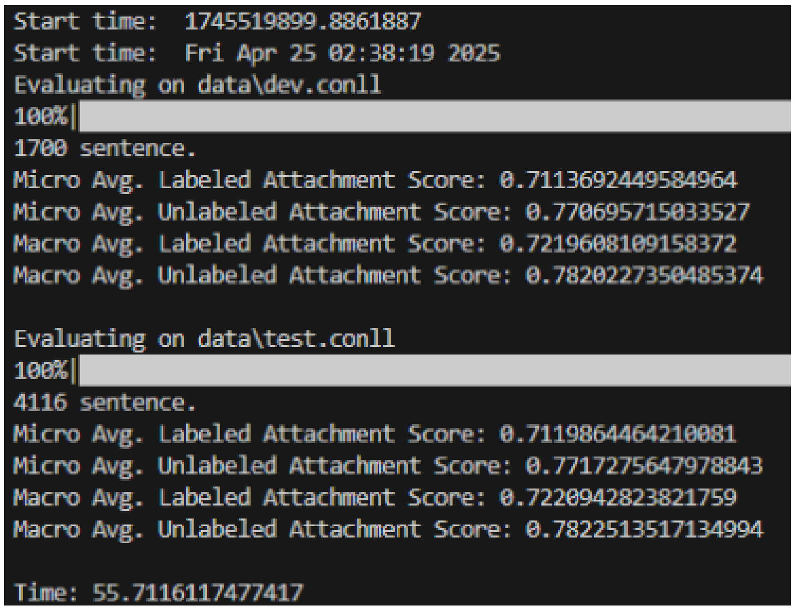


```
Start time:  1745566747.8654244
Start time:  Fri Apr 25 15:39:07 2025
Evaluating on data\dev.conll
100%|
1700 sentence.
Micro Avg. Labeled Attachment Score: 0.673579779145998
Micro Avg. Unlabeled Attachment Score: 0.7504798464491362
Macro Avg. Labeled Attachment Score: 0.6860821905354797
Macro Avg. Unlabeled Attachment Score: 0.7614920797383357

Evaluating on data\test.conll
100%|
4116 sentence.
Micro Avg. Labeled Attachment Score: 0.6750136878751253
Micro Avg. Unlabeled Attachment Score: 0.7528021404737555
Macro Avg. Labeled Attachment Score: 0.6861990368918941
Macro Avg. Unlabeled Attachment Score: 0.7624548666055083

Time: 97.19530844688416
```

WordPOSModel:



```
Start time:  1745519899.8861887
Start time:  Fri Apr 25 02:38:19 2025
Evaluating on data\dev.conll
100%|
1700 sentence.
Micro Avg. Labeled Attachment Score: 0.7113692449584964
Micro Avg. Unlabeled Attachment Score: 0.770695715033527
Macro Avg. Labeled Attachment Score: 0.7219608109158372
Macro Avg. Unlabeled Attachment Score: 0.7820227350485374

Evaluating on data\test.conll
100%|
4116 sentence.
Micro Avg. Labeled Attachment Score: 0.7119864464210081
Micro Avg. Unlabeled Attachment Score: 0.7717275647978843
Macro Avg. Labeled Attachment Score: 0.7220942823821759
Macro Avg. Unlabeled Attachment Score: 0.7822513517134994

Time: 55.7116117477417
```

Conclusion

This assignment successfully implemented a neural transition-based dependency parser using feed-forward networks, achieving reasonable performance (70+% UAS) while demonstrating the value of POS tags through the superior WordPOSModel. Future work could explore more advanced transition systems like the arc-eager parser (which handles non-projective trees more efficiently), incorporate contextualized word embeddings for better feature representation, or implement dynamic oracle training to improve the model's decision-making during parsing. Additionally, the parser could be enhanced with graph-based or ensemble methods to further boost accuracy, while error analysis on specific dependency types (like long-distance attachments) could reveal targeted areas for improvement. These extensions would help bridge the performance gap with state-of-the-art parsers while maintaining the model's efficiency.