NLP Ass5 Dependency Parsing

Name: Ziheng Wang

Student ID: 12310401

Task 1-5

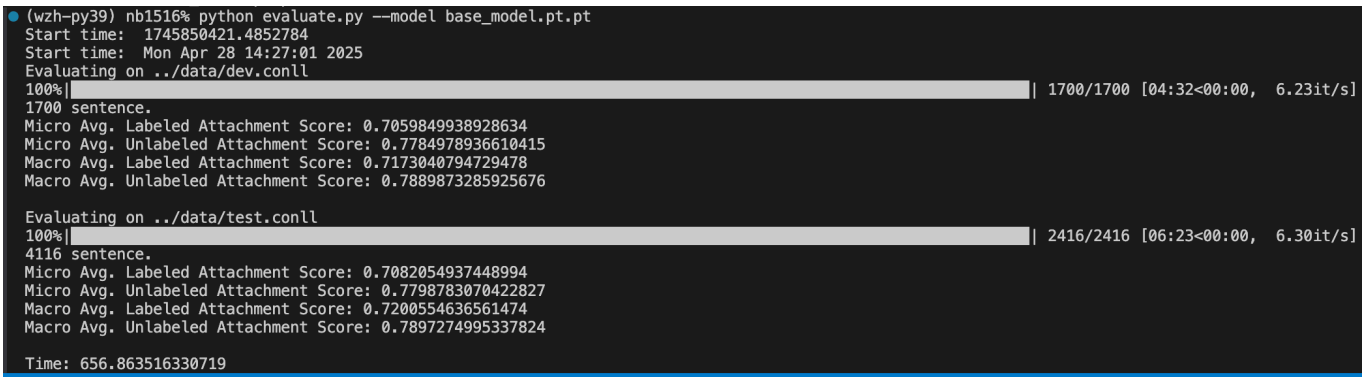
Base Model

Before training the model, please check the `get_train_data.py` and ensure use the `extractor.get_input_repr_word` function in `get_training_matrices`.

Then

```
python get_train_data.py $DATAPATH
python train.py --model_type base
python evaluate.py --model base_model.pt
```

And the evaluate results are



It meet the requirement that UAS >=70%

WordPos Model

Before training the model, please check the `get_train_data.py` and ensure use the `extractor.get_input_repr_wordpos` function in `get_training_matrices`.

Then

```
python get_train_data.py $DATAPATH
python train.py --model_type wordpos
python evaluate.py --model wordpos_model.pt
```

```
(wzh-py39) nb1516% python evaluate.py --model wordpos_model2.pt
Start time:  1745899813.7140656
Start time:  Tue Apr 29 04:10:13 2025
Evaluating on ../data/dev.conll
100%|| 1700/1700 [04:58<00:00,  5.70it/s]
1700 sentence.
Micro Avg. Labeled Attachment Score: 0.7336789889572999
Micro Avg. Unlabeled Attachment Score: 0.7931300944736646
Macro Avg. Labeled Attachment Score: 0.7430796456966959
Macro Avg. Unlabeled Attachment Score: 0.8025513369488261

Evaluating on ../data/test.conll
100%|| 2416/2416 [07:00<00:00,  5.74it/s]
4116 sentence.
Micro Avg. Labeled Attachment Score: 0.7346205101186971
Micro Avg. Unlabeled Attachment Score: 0.7940413838700013
Macro Avg. Labeled Attachment Score: 0.7428079488368885
Macro Avg. Unlabeled Attachment Score: 0.8015509160956362

Time: 723.7107930183411
```

Bonus Task 6

The Arc-Eager parsing algorithm is implemented in `parser_arc-eager.py`

And their different are as follows:

Arc-standard parser result:

```
(wzh-py39) nb1516% python parser.py --model base_model.pt
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       xcomp   _       _
17      specific        _       _       JJ      _       19      amod    _       _
18      congressional   _       _       JJ      _       19      amod    _       _
19      authorization   _       _       NN      _       16      dobj    _       _
20      .       _       _       .       _       3       punct   _       _
```

The Arc-Eager parser result:

```
(wzh-py39) nb1516% python parser.py --model base_model.pt
1       The     _       _       DT      _       2       det     _       _
2       bill    _       _       NN      _       3       nsubj   _       _
3       intends _       _       VBZ     _       0       root    _       _
4       to      _       _       TO      _       5       mark    _       _
5       restrict        _       _       VB      _       0       root    _       _
6       the     _       _       DT      _       7       det     _       _
7       RTC     _       _       NNP     _       10      nmod:poss       _       _
8       to      _       _       TO      _       10      case    _       _
9       Treasury        _       _       NNP     _       10      compound        _
10      borrowings      _       _       NNS     _       5       dobj    _       _
11      only    _       _       RB      _       10      advmod  _       _
12      ,       _       _       ,       _       16      punct   _       _
13      unless  _       _       IN      _       16      mark    _       _
14      the     _       _       DT      _       15      det     _       _
15      agency  _       _       NN      _       16      nsubj   _       _
16      receives        _       _       VBZ     _       0       root    _       _
17      specific        _       _       JJ      _       19      amod    _       _
18      congressional   _       _       JJ      _       19      amod    _       _
19      authorization   _       _       NN      _       16      nmod    _       _
20      .       _       _       .       _       19      punct   _       _
```

Bonus Task 7

The BiLSTM model is implemented in `model.py` and the training process is similar to the wordpos model.

```
python get_train_data.py $DATAPATH // same as wordpos train data
python train.py --model_type bilstm
python evaluate.py --model bilstm_model.pt
```

The evaluation result is

```
(wzh-py39) nb1516% python evaluate.py --model bilstm_model.pt
Start time:  1745891944.7258475
Start time:  Tue Apr 29 01:59:04 2025
Evaluating on ../data/dev.conll
100%| | 1700/1700 [08:38<00:00,  3.28it/s]
1700 sentence.
Micro Avg. Labeled Attachment Score: 0.7538699304534238
Micro Avg. Unlabeled Attachment Score: 0.8131465463519206
Macro Avg. Labeled Attachment Score: 0.7642068524114066
Macro Avg. Unlabeled Attachment Score: 0.8235288529964618

Evaluating on ../data/test.conll
100%| | 2416/2416 [12:09<00:00,  3.31it/s]
4116 sentence.
Micro Avg. Labeled Attachment Score: 0.7550438528527598
Micro Avg. Unlabeled Attachment Score: 0.814144481978492
Macro Avg. Labeled Attachment Score: 0.7625585455256031
Macro Avg. Unlabeled Attachment Score: 0.820991121273601
```

The UAS and LAS are higher than the wordpos model.

Model	dev UAS	dev LAS	test UAS	test LAS
base	0.789	0.717	0.790	0.720
wordpos	0.803	0.743	0.802	0.742
bilstm	0.823	0.764	0.821	0.763