Lab 5. Assignment guidelines & Building/training dependency parser 2

LING-581-Natural Language Processing 1

Instructor: Hakyung Sung

September 25, 2025

Building/training dependency parser

Goal

Train a dependency parser on the provided CoNLLU-formatted training data and generate predictions—heads (*HEAD*) and dependency labels (*DEPREL*)—for the test set.

Overview

- **Write your own parser** using the stack, buffer, and shift/reduce mechanisms discussed in class (more guidelines on the Colab)
- **Use an existing library/toolkit** to train a parser (you'll need to research options).
- **Propose another reasonable method** not covered in class.
- **Tip.** If it feels difficulty, vibe-coding is encouraged; make sure your work is reproducible (fixed seeds, documented environment) and well documented/commented.

Shared thoughts in class

https://docs.google.com/document/d/1UAloZzdCr9JJo1nVU5sbTIszM91085ELTaGNFrqX_qM/edit?usp=sharing

Data

Use the provided **train** / **dev** / **test** splits.

Link:*https://github.com/hksung/Fall25_PythonTutorial/tree/main/corpus/dep_parse*

Data Format (10 tab-separated fields, CoNLL-U)

1. *ID* (word index, starting at 1; may be a range for multiword tokens)
2. *FORM* (word form or punctuation mark)
3. *LEMMA* (lemma or stem; _ if not available)
4. *UPOS* (universal part-of-speech tag)
5. *XPOS* (language-specific part-of-speech tag; _ if not available)
6. *FEATS* (list of morphological features; _ if not available)
7. *HEAD* (ID of the head word; 0 if the current word is the root)
8. *DEPREL* (universal dependency relation to the HEAD; or subtype)
9. *DEPS* (enhanced dependency graph; _ if not available)
10. *MISC* (miscellaneous annotation; _ if not available)

Submission

1. .ipynb file
2. predicted test.conllu (with dependency label and relations)

```
# sent_id = file01141.txt_26
# text = I have a little audio set .
1       I       _       PRON    PRP     _       0       _       _       _
2       have    _       VERB    VBP     _       0       _       _       _
3       a       _       DET     DT      _       0       _       _       _
4       little  _       ADJ     JJ      _       0       _       _       _
5       audio   _       NOUN    NN      _       0       _       _       _
6       set     _       NOUN    NN      _       0       _       _       _
7       .       _       PUNCT   .       _       0       _       _       _
```

Grading (1/2)

1. **.ipynb notebook** *(7 points)*
    - Clearly describe your approach (algorithm/model), libraries, and key hyperparameters. (4 points)
    - Ensure the notebook runs top-to-bottom without errors (*Before submitting your file, all the codes should be run - the grader needs to check the output). (3 point)

Grading (2/2)

2. **Predictions** *(3 points)*
   - Submit ***`test.conllu`*** (make sure that your output has the same format!) with your **predicted HEAD and DEPREL** (match the input format; preserve IDs/tokenization; use ***`0`*** for ROOT if applicable).

   Scoring rubric (by LAS):
   - $\geq 80.0\% \rightarrow$ **3 pts**
   - **70.0–79.9%** $\rightarrow$ **2 pts**
   - **60.0–69.9%** $\rightarrow$ **1 pt**

   ***Extra credit:** The **single highest LAS** in the class earns **+1 point**.