LING3401 Linguistics and Information Technology

Tutorial: NLP for linguistic analysis

Yige Chen

The Chinese University of Hong Kong

March 19, 2025

Introduction

- Before LLMs emerge, texts are usually processed morphologically, syntactically, and semantically before they are fed to downstream tasks
- Tokenization
- Part-of-speech (POS) tagging
- Parsing
- Semantic role labeling
- ...

Sequence labeling with BIO scheme

- Sequence labeling is a task where a label is assigned to each token in a sequence
- The BIO scheme is commonly used
  - B (Begin) indicates the start of an entity
  - I (Inside) indicates a continuation of the entity
  - O (Outside) indicates no entity
- Example:
  - Sentence: "John lives in New York"
  - Labels: B-PER O O B-LOC I-LOC

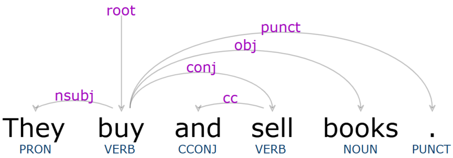
Universal Dependencies (UD)

- Universal Dependencies (UD) is a framework for consistent annotation of grammar across languages
- It provides a unified scheme for morphological, syntactic, and dependency annotation
- UD consists of:
  - Part-of-Speech (POS) tags (e.g., `NOUN`, `VERB`, `ADJ`)
  - Dependency relations (e.g., `nsubj`, `obj`, `root`)
  - Morphological features (e.g., `Number=Plur`, `Tense=Pres`)
- https://universaldependencies.org/

Example: CoNLL-U Format

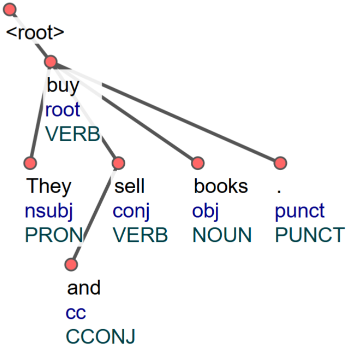
- The CoNLL-U format is used to represent linguistic annotations as in Universal Dependencies
- Example sentence: *They buy and sell books.*

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS
1	They	they	PRON	PRP	Case=Nom|Number=Plur	2	nsubj	2:nsubj|4:nsubj
2	buy	buy	VERB	VBP	Number=Plur|Person=3|Tense=Pres	0	root	0:root
3	and	and	CCONJ	CC	-	4	cc	4:cc
4	sell	sell	VERB	VBP	Number=Plur|Person=3|Tense=Pres	2	conj	0:root|2:conj
5	books	book	NOUN	NNS	Number=Plur	2	obj	2:obj|4:obj
6	.	.	PUNCT	.	-	2	punct	2:punct



Visualizing parsed trees

- Check Universal Dependencies and its visualization tools
  - https://universaldependencies.org/conllu_viewer.html
  - https://weblicht.sfs.uni-tuebingen.de/Tundra/
- The input needs to be in the CoNLL-U format
- Try removing the enhanced dependencies before visualization



POS tagging

- Part-of-speech (POS) tagging assigns a grammatical category to each word
- Universal POS (UPOS) tags can be found at https://universaldependencies.org/u/pos/
- Language-specific POS (XPOS) tags differ from treebanks to treebanks, and you will need to consult the documentation for each treebank

Try it out!

- In today's Colab notebook, we are using the POS taggers from spaCy and Stanza for POS tagging
- Try providing your own sentences and see if the two taggers assign different labels onto the tokens!
- Also, pay attention to how UPOS and XPOS are different!

Parsing and parsers

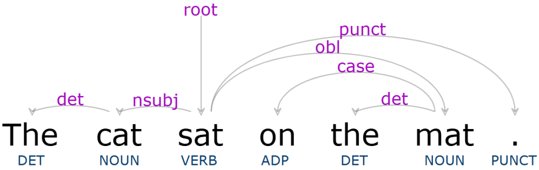
- A parser analyzes the syntactic structure of a sentence by determining relationships between words
- Helps break down sentences into constituency (phrase structure) or dependency (word relationships) trees
- The parsers utilize the token as their input. Some utilize lemma, POS tags as well
- We will use two dependency parsers, one from spaCy `https://spacy.io/` and the other being Stanza `https://stanfordnlp.github.io/stanza/`

Do I want to train my own POS tagger or parser?

- Short answer: not recommended
- Yes, you can train your own POS tagger or parser using a treebank, with HMM or CRF or neural/transformer models (e.g., BERT; but you still need, for instance, a CRF or LSTM classifier for sequence labeling)
- But I won't recommend doing this
  - Existing parsers are usually well-trained with abundant treebank data
  - Unless you have a new treebank (for example, you have collected and annotated your own treebank in a low-resource language) or your own data format (for example, as in my paper), training your own parsers are not recommended
  - If that day really comes, shoot me an email and I'll let you know how to do that

Dependency parsing

- Dependency parsing represents grammatical relations between words
- Words are connected based on dependency relations
- The head word governs the dependent words
- Example:
- Sentence: "The cat sat on the mat"



Constituency parsing

- Constituency parsing analyzes the hierarchical structure of sentences
- It represents sentences using a tree structure
- Each phrase corresponds to a subtree
- Example:
- Sentence: "The cat sat on the mat"
- (S (NP (DT The) (NN cat)) (VP (VBD sat) (PP (IN on) (NP (DT the) (NN mat)))))

Try it out!

- In today's Colab notebook, I have provided parsers from spaCy and Stanza for POS tagging and dependency parsing, and Berkeley Neural Parser for constituency parsing
- This is a good chance for you to compare dependency grammar and constituency grammar and how a single sentence can be parsed in two different ways

Using LLMs for POS tagging and parsing

- Yes, we can fine-tune BERT for sequence labeling
- But how about we just use generative models like GPT with prompting?
- Try asking LLMs to provide POS tags and dependency/constituency relations!
- Note: Decoder-only LLMs like GPT series may not outperform word embeddings + CRF or encoder-only models like BERT!

Miscellaneous

- Please do not hesitate to ask questions
- We enjoy feedback from you, so please let us know if you feel there's anything we could have done better
- It would be great if you'd bring your laptop to the class every week