ANLP Tutorial Exercise Set 3 (for tutorial groups in week 6)

v1.4
School of Informatics, University of Edinburgh
Henry Thompson, Sharon Goldwater

This week's tutorial exercises focus on syntax, (English) grammar, and parsing, using both context-free grammar and dependencies. After working through the exercises and discussing some additional issues in your tutorial groups, you should be able to:

- Provide examples showing how syntactic structure reflects the semantics of a sentence, and in particular, semantic ambiguity. You should also be able to explain and illustrate how constituency parses and dependency parses differ with respect to this issue.
- Provide the syntactic parse for simple sentences using either Universal Dependencies or context-free grammar rules.
- Hand-simulate the CKY parsing algorithm and transition-based dependency parsing, and by doing so, better understand some of the computational issues involved.

1 CFGs and attachment ambiguity

When constructing a grammar and parsing with it, one important goal is to accurately reflect the meaning of sentences in the structure of the trees the grammar assigns to them. Assuming a compositional semantics, this means we would expect attachment ambiguity in the grammar to reflect alternative interpretations. The following two exercises aim to hone your intuitions about the syntax-semantics relationship.

Exercise 1.
In English, conjunctions often create attachment ambiguity, as in the sentence `I like green eggs and ham`. The ambiguity inside the noun phrase here could be captured by the following two context-free grammar rules, where `Nom` is a nominal (noun-like) category:

```
Nom → Adj Nom
Nom → Nom Conj Nom
```

a) Write down two paraphrases of `I like green eggs and ham`, where each paraphrase unambiguously expresses one of the meanings.

b) Draw two parse trees for just the `Nom` part of the sentence, illustrating the ambiguity. You'll also need to use a rule `Nom → N`. Which tree goes with which paraphrase?

Exercise 2.
Another common source of attachment ambiguity in English is from prepositional phrases. The relevant grammar rules include:

```
VP → V NP
VP → VP PP
NP → NP PP
PP → P NP
```

Here are five verb phrases:

(1) watched the comet from the roof with my telescope
(2) watched the comet from the park across the street
(3) watched a video by the BBC about the comet
(4) watched a video about the expedition to the comet
(5) watched a video about the comet on my mobile

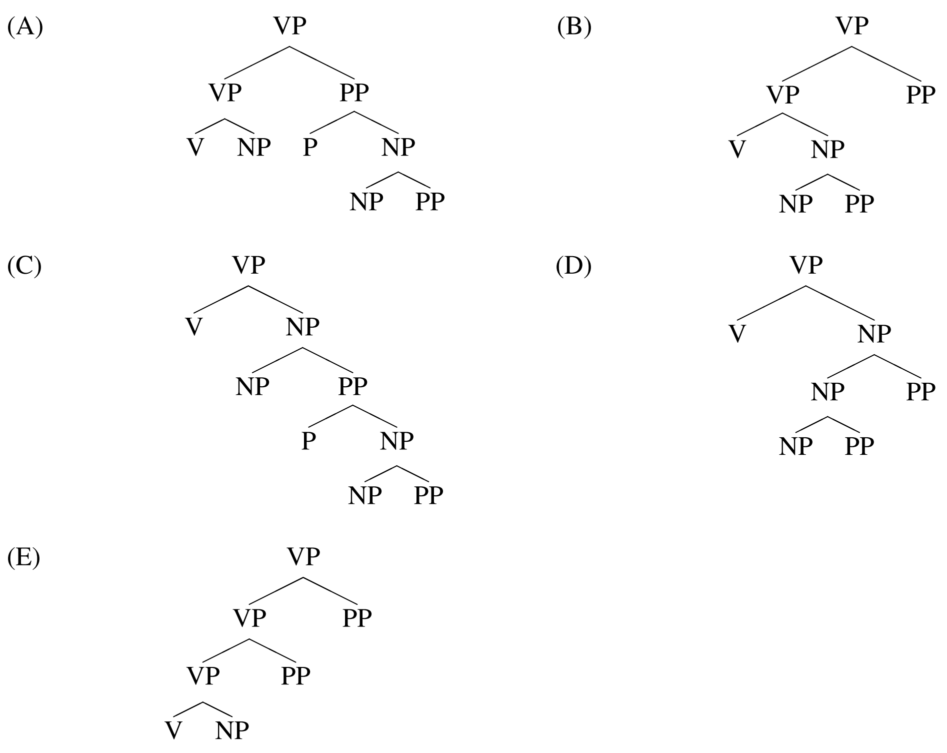


Figure 1: Trees for exercise 2

Figure 1 shows five partial trees. Match the phrases to the trees which best capture their meanings. You may find it helpful to ask yourself questions such as “where did this event happen?”, “how was it done?”, “what was watched?”. You may also want to try out (in pencil!) different ways of writing in phrases under the leaves of the various trees.

2 CKY parsing

Exercise 3.
Assume we are using the following grammar:

```
S → NP VP          V → swam | ran | flew
VP → V NP          VP → swam | ran | flew
VP → VP PP         D → the | a | an
NP → D N           N → pilot | plane
NP → NP PP         NP → Edinburgh | Glasgow
PP → P NP          P → to
```

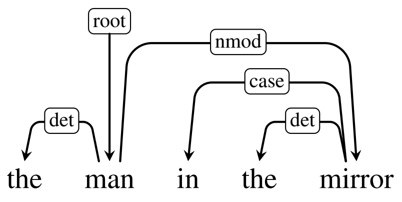
a) Draw a 7x7 chart for the sentence `the pilot flew the plane to Glasgow` and fill it in using the CKY algorithm. Number the symbols you put in the matrix in the order they would be computed, assuming the grammar is searched top-to-bottom.

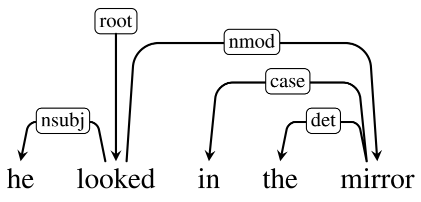
b) How is the attachment ambiguity present in this sentence reflected in the chart at the end?

3 Dependency syntax and parsing

Exercise 4.

a) Draw dependency parses for verb phrases (2) and (3) from exercise 2, using UD labels for the relations as illustrated in JM3. You shouldn't need to know any more labels than: `nsubj, dobj, iobj, det, case, nmod, amod`. You should be able to figure out most of the labels by looking at examples from the textbook. For prepositional phrases, use the `nmod` relation, as in these examples:[1]



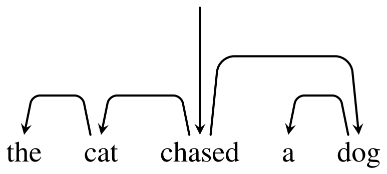


Note that by convention, all dependency parses have a root, whether or not it is the head of a full sentence.

b) Now try to draw a dependency parse for the sentence `I like green eggs and ham`. You will need to use the `cc` and `conj` labels (see examples in JM3, Fig 15.3). Do you run across any problems? Is it clear what the dependency structure should be? Is the ambiguity in this sentence represented in the dependency structure (or multiple structures), and if so how?

Exercise 5.

Consider the following dependency-annotated sentence. (For simplicity, we leave out the relation labels in this exercise).



By hand-simulating the algorithm for arc-standard transition-based parsing, show that there is more than one sequence of transitions that can lead to the correct parse of this sentence. How does this fact motivate the need for the procedure described in JM3 section 15.4.1 (generating the training oracle)? What is the sequence produced by the training oracle?

[1]The textbook and this tutorial more or less follow the UD v1 guidelines; UD guidelines have now been updated for v2, so if you look online you may find discrepancies.