Methods for Tree-Drawing

There are four ways to draw trees that you can insert into your assignments.

1. Hand-drawing
2. Using a desktop program
3. Using a web-based app
4. Using LaTeX

Below I briefly describe each of these methods, and give links to the various tools.

1 Hand-Drawing

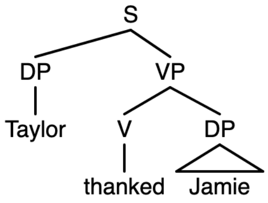
- ❖ If you hand-draw your trees, you need to find a way to add include drawings with your assignment
  - ❧ Your whole assignment could be hand-written/-drawn, in which case you can just scan (and convert-to-PDF) the entire assignment
  - ❧ If you want to write-up your solution on your computer, but hand-draw trees, you can {scan/take a picture of} your trees and insert them into your document
  - ❧ If you have a tablet or some other technology that allows digital drawing, you can draw your trees digitally and insert the drawings into your document

2 Drag-and-Drop Editor

- ❖ The easiest to learn method of building trees is a drag-and-drop editor
  - ❧ i.e., one where you can build trees through a point-and-click-interface
- ❖ The first (and easiest) tool mention is a web-app:
  - ❧ **Sapling** (https://groverburger.github.io/sapling/)

PRO:	there's no software to install
PRO:	usage is very intuitive
PRO:	also allows trees to be saved/uploaded as editable text (coded as JSON files)
CON:	triangle branches require a space somewhere in the label (i.e. you need to add a space in single-word labels)

  - ❧ Example output:



- ❖ Here's another drag-and-drop editor, which you can install on your computer (and use offline)
  - ❧ **TreeForm** (https://sourceforge.net/projects/treeform/)

PRO: very easy to use
CON: requires Java on your computer
CON: I have heard of frustrations with certain aspects/limitations of the program

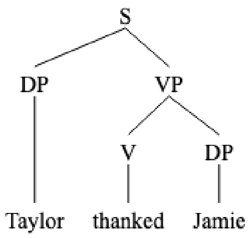
- Tutorial video here: http://www.ece.ubc.ca/ donaldd/TreeFormVideo.mov

3 Simple Bracket-Based Editors

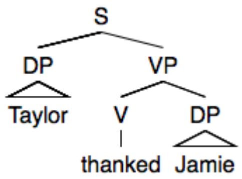
- The next easiest way to draw trees is with a tool that takes bracketed text and produces images of trees as output
- Some of the basic pros/cons of using this type of tree-drawing method:

PRO: there's no software to install
PRO: usage is very intuitive
CON: formatting options (e.g., italics/strikethrough) are limited
CON: some web-based apps don't support arrow drawing

- As far as I know, there are three bracket-based web apps of this type, which I list below
  - **jsSyntaxTree** (http://ironcreek.net/syntaxtree/)
    - Nice features: renders as you type, supports arrow-drawing, more tree-formatting options
    - Unique feature: has an 'align at bottom' option
    - Unique weakness: no way to draw triangles for a single-word
    - Example: `[S [DP Taylor ] [VP [V thanked ] [DP Jamie ]]]`

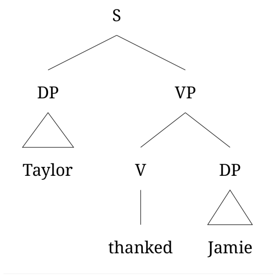


  - **Syntax Tree Generator** (http://mshang.ca/syntree/)
    - Nice features: renders as you type, supports arrow-drawing, more tree-formatting options
    - Unique feature: a unique shareable link can be generated for each tree you create
    - Example: `[S [^DP Taylor ] [VP [V thanked ] [^DP Jamie ]]]`



  - **RSyntaxTree** (https://yohasebe.com/rsyntaxtree/)
    - Nice features: deals best with formatting (e.g., superscript, cross-out, underline, ...), has an array of fonts, has tools for checking brackets
    - Unique feature: allows mutli-line labels
    - Unique weaknesses: doesn't render as you type

        - Example: `[S [DP ^Taylor ] [VP [V thanked ] [DP ^Jamie ]]]`



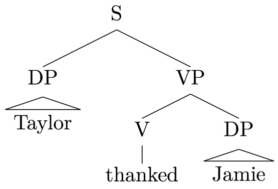
- HINT: If you want to draw arrows or do strikethrough, and your web-based app of choice cannot do this, you can always achieve this by adding these mark-ups to the tree image by using word processor's drawing tools or by using an image editor.

4 Using LaTeX

- If you already know the basics of how LaTeX works, I recommend using it to draw your trees
    - You do not need to have LaTeX installed on your own machine - you can use web-based apps like http://overleaf.com
- Some of the basic pros/cons of using this type of tree-drawing method:

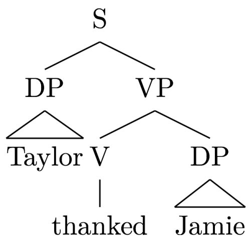
  PRO: LaTeX is powerful and lets you do a lot
  PRO: most any formatting/tree-style you want is possible
  CON: at least basic knowledge of LaTeX is required
  CON: troubleshooting mistakes can be painstakingly difficult, and might be as hard to find as a forgotten ";"

- There are several different packages for tree-drawing, which I list below – ***I prefer the 3rd option, `tikz-qtree`***
    - `qtree` https://www.ctan.org/pkg/qtree
        - Pro: standard tree drawing package
        - Con: arrow drawing is somewhat complex, and relies on an external pacakge, `tree-dvips`
        - Con: arrows will not compile with pdfLaTeX; you must use XeLaTeX or LaTeX→dvips→ps2pdf
        - Example:



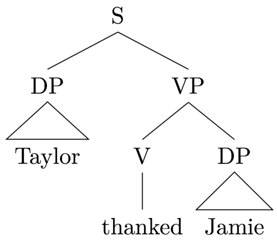
```
\Tree
[.S
\qroof{Taylor}.DP
[.VP
[.V thanked ]
\qroof{Jamie}.DP
]]
```

    - `pst-jtree` https://www.ctan.org/pkg/pst-jtree
        - Pro: has a very easy to read "colon construction" for building trees
        - Pro: arrow drawing is very easy to learn
        - Con: will not compile with pdfLaTeX; you must use XeLaTeX or LaTeX→dvips→ps2pdf
        - Example:



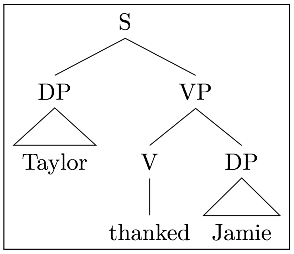
```
\jtree
\! = {S}
:({DP}<vartri>{Taylor}) {VP}
:({V}<vert>{thanked}) {DP}<vartri>{Jamie}
.
\endjtree
```

- `tikz-qtree` https://www.ctan.org/pkg/tikz-qtree
  - Pro: built off of `qtree` but improves in many ways including better formatting, and a built-in way to draw arrows
  - Pro: Compiles with any TeX engine
  - Con: All nodes must have labels in order for it to format correctly
  - See http://home.uni-leipzig.de/murphy/handouts/tikz-qtree%20tutorial.pdf for a nice how-to guide (including how to draw arrows)
  - Example:



```
\begin{tikzpicture}
\Tree
[.S
[.DP \edge[roof]; Taylor ]
[.VP
[.V thanked ]
[.DP \edge[roof]; Jamie ]
]]
\end{tikzpicture}
```

- **HINT**: you don't have to write your whole assignment/paper in LaTeX - you can use it to produce PDFs that are exactly the size of your tree, by using the documentclass `standalone`
  - This tree below is actually its own PDF (created with the code to its right, without any doing any cropping 'by hand') that I've re-inserted into this PDF:



```
\documentclass[12pt, border={0pt 0pt 0pt 0pt}]{standalone}
\usepackage{tikz-qtree}
\begin{document}
\begin{tikzpicture}
\Tree
[.S
[.DP \edge[roof]; Taylor ]
[.VP
[.V thanked ]
[.DP \edge[roof]; Jamie ]
]]
\end{tikzpicture}
\end{document}
```

  - Warning: if you use pst-jtree or tree-dvips to draw your arrows, they might get cut off using the `standalone` documentclass. You need to manipulate the `border` parameter, which is organized as `border={left bottom right top}`.