Drawing syntactic trees with `tikz-qtree`

Andrew Murphy
andrew.murphy@uni-leipzig.de

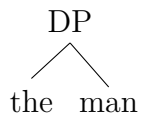
LaTeX for Linguists – Doktorandenforum 2015

03.03.15

1 The basics

- `tikz-qtree` takes the standard tree drawing package (`qtree`) and incorporates it into the powerful graphics program TikZ. This makes drawing arrows etc. much easier and gives your far more control over your trees.
- *Note*: The basic syntax of `tikz-qtree` can be used with the standard `qtree` package. This means you could use `\usepackage{qtree}` and use the `\Tree` commands we will learn here without the `tikzpicture` environment. However, drawing arrows requires an additional PS package (e.g. `pst-node`) and a much more complicated compiling procedure (LaTeX→ dvips → ps2pdf). `tikz-qtree` can do all of this (and more) with one step.
- The basic syntax of `tikz-qtree` corresponds to bracket notation also used in the syntactic literature. so [$_{\text{DP}}$ the man] is written as follows:

```
\begin{tikzpicture}
\Tree [.DP the man ]
\end{tikzpicture}
```

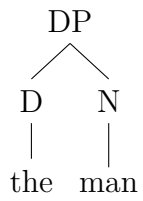


- Open `trees.tex` and add this code in between the `tikzpicture` environment provided.
- It is **very** important to note that there must be a space between a closing bracket and anything else. For example, the following code would cause LaTeX to crash:

```
\begin{tikzpicture}
\Tree [.DP the man]
\end{tikzpicture}
```

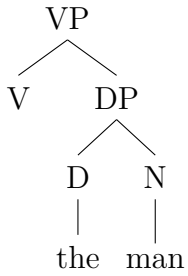
- The label of each node is written after a dot (.) following the opening bracket. We can modify the example above to add D and N nodes for *the* and *man*:

```
\begin{tikzpicture}
\Tree [.DP [.D the ] [.N man ] ]
\end{tikzpicture}
```



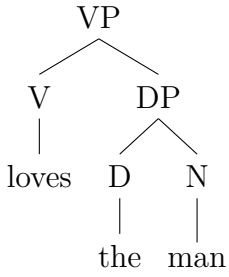
- In the `.tex` file, modify the code accordingly.
- Now, we can add to this structure by adding another set of brackets around what we have to form a VP:

```
\begin{tikzpicture}
\Tree [.VP V [.DP [.D the ] [.N man ] ]]
\end{tikzpicture}
```



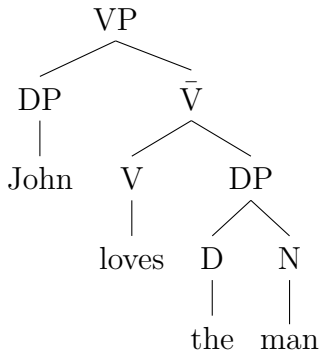
- Then we can add brackets around the V node we created and turn this into the verb *loves*.

```
\begin{tikzpicture}
\Tree [.VP [.V loves ] [.DP [.D the ] [.N man ] ]]
\end{tikzpicture}
```



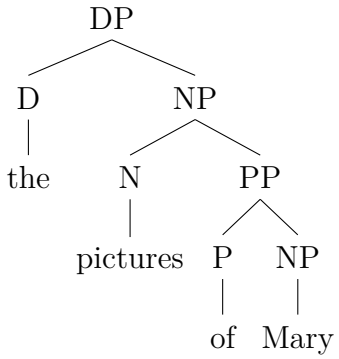
- Add a subject in the VP (this will involve creating an $\bar{\text{X}}$ or X′ level.

```
\begin{tikzpicture}
\Tree [.VP [.DP John ] [.{\=V} [.V loves ] [.DP [.D the ] [.N man ] ]]]
\end{tikzpicture}
```



- If you prefer V′, then use `V$'$` instead of `{\=V}`.

  Exercise: Try to draw the following tree:

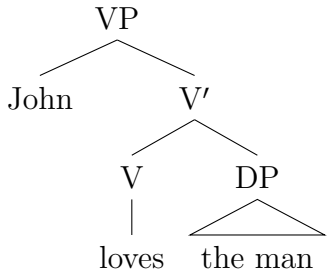


2 Options for trees

2.1 Options for `tikz-qtree-compat`

- The package `tikz-qtree-compat` allows you to use a number of commands from the `qtree` package.
- The command `\qroof{text under triangle}.XP` will draw a triangle under a node:

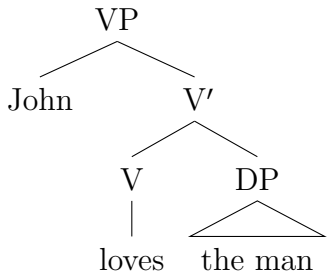
```
\begin{tikzpicture}
\Tree [.VP John [.V$'$ [.V loves ] \qroof{the man}.DP ]]
\end{tikzpicture}
```



  Exercise: Take the tree you just drew and draw a triangle under the PP!

- The `\1` command will give you bar levels:

```
\begin{tikzpicture}
\Tree [.VP John [.V\1 [.V loves ] \qroof{the man}.DP ]]
\end{tikzpicture}
```

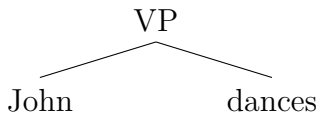


- If you want to have multiple dashes (X″) use `X\2`.

2.2 Tree options

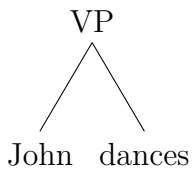
- It is possible to customize your tree in a number of ways by adding options in square brackets after `\begin{tikzpicture}[...]`.
- Alternatively, these can specified separately in a `\tikzset{...}` environment.
- For example, the distance between branches can be set with `sibling distance=72pt`:

```
\begin{tikzpicture}[sibling distance=50pt]
\Tree [.VP John dances ]
\end{tikzpicture}
```



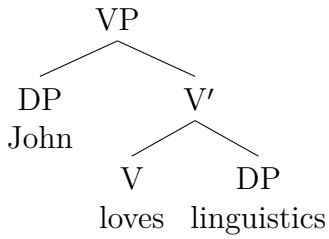
- The same can be done with `level distance`:

```
\begin{tikzpicture}[level distance=50pt]
\Tree [.VP John dances ]
\end{tikzpicture}
```



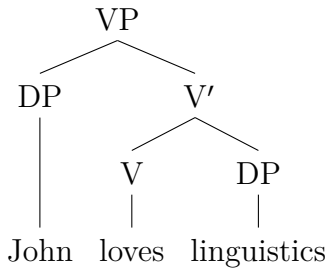
- If you do not want have the lines going from terminal nodes, it is necessary to add the following code either in the options (or tikzset as below). You can then simply use `\\` for a new line, i.e. `V\\{loves}`.

```
\begin{tikzpicture}
\tikzset{every tree node/.style={align=center,anchor=north}}
\Tree [.VP DP\\{John} [.V\1 V\\{loves} DP\\{linguistics} ]]
\end{tikzpicture}
```



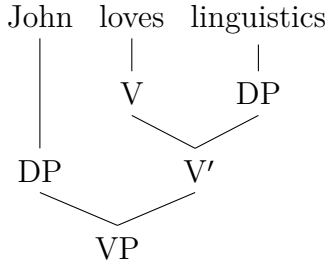
- `tikzset` allows you to modify trees so that the leaves of the tree are aligned in a row (this is done with `anchor=base`):

```
\begin{tikzpicture}
\tikzset{every tree node/.style={align=center,anchor=base}}
\tikzset{level 1+/.style={level distance=2\baselineskip}}
\tikzset{frontier/.style={distance from root=6\baselineskip}}
\Tree [.VP [.DP John ] [.V\1 [.V loves ] [.DP linguistics ] ]]
\end{tikzpicture}
```



- It is also possible to make trees grow upside-down by adding `\begin{tikzpicture}[grow'=up]` (other options are `down`, `left` and `right`):

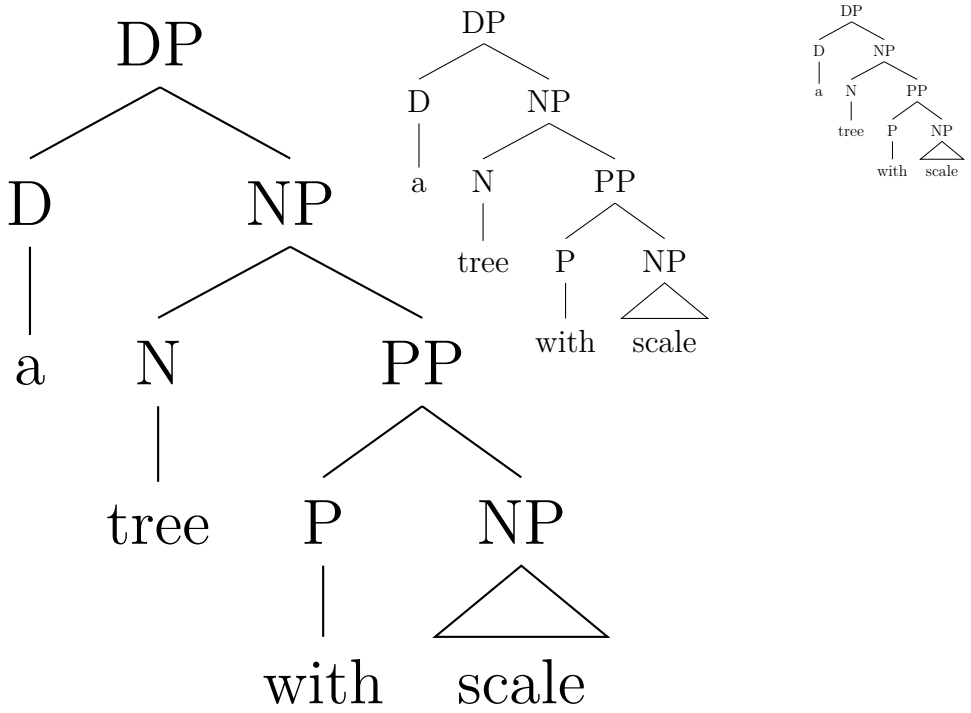
```
\begin{tikzpicture}[grow'=up]
\tikzset{every tree node/.style={align=center,anchor=base}}
\tikzset{level 1+/.style={level distance=2\baselineskip}}
\tikzset{frontier/.style={distance from root=6\baselineskip}}
\Tree [.VP [.DP John ] [.V\1 [.V loves ] [.DP linguistics ] ]]
\end{tikzpicture}
```



- The size of trees can be manipulated easily by adding the option `scale=`x. The value `1` is the standard size, so `2` will be twice as big and `0.5` half as big and so on.[1]

[1] I have added a `multicols` environment here for multiple columns (requires the package `multicol`).

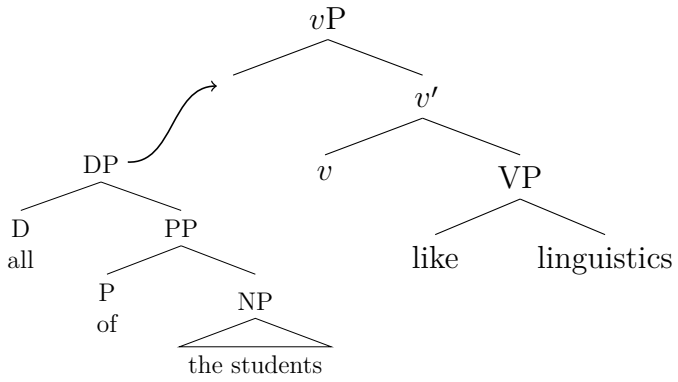
```
\begin{multicols}{3}

\begin{tikzpicture}[scale=2]
\Tree [.DP [.D a ] [.NP [.N tree ] [.PP [.P with ] \qroof{scale}.NP ] ]]
\end{tikzpicture}

\columnbreak

\begin{tikzpicture}
\Tree [.DP [.D a ] [.NP [.N tree ] [.PP [.P with ] \qroof{scale}.NP ] ]]
\end{tikzpicture}

\columnbreak

\begin{tikzpicture}[scale=0.5]
\Tree [.DP [.D a ] [.NP [.N tree ] [.PP [.P with ] \qroof{scale}.NP ] ]]
\end{tikzpicture}

\end{multicols}
```



- It is also possible to grow have multiple trees in a single `tikzpicture` using `\begin{scope}`:
- The options `xshift` and `yshift` define the position of the second tree (the one in scope environment). A positive value for xshift will move it further to the right of the other tree, and a negative one will shift it to the left.

```
\begin{tikzpicture}
\tikzset{every tree node/.style={align=center,anchor=north}}
\tikzset{level 1+/.style={sibling distance=1.5\baselineskip}}
\Tree [.$v$P \node(e){~}; [.$v$\1 $v$ [.VP like linguistics ]]]
```

```
\begin{scope}[xshift=-3cm,yshift=-2cm, scale=0.8]
\Tree [.\node(d){DP}; D\\{all} [.PP P\\{of} \qroof{the students}.NP ]]
\draw[semithick,->] (d) to [out=0, in=180] (e);
\end{scope}

\end{tikzpicture}
```

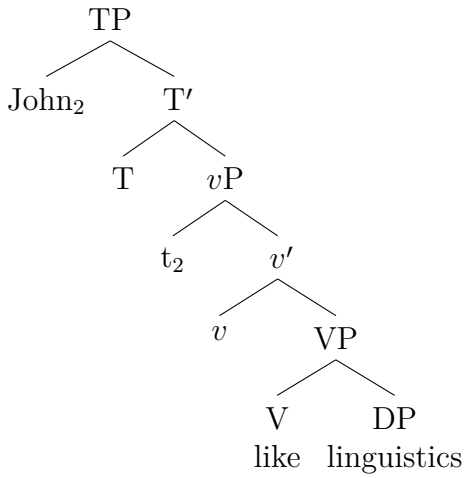


3 Tree ornaments: arrows, labels

- Now that we have our basic trees, this section will show you how to add extra 'ornaments' to them such as arrows, annotations and any other things you might want to draw.
- Drawing arrows involves defining the start and end points of the arrows (nodes). The syntax involved is `\node(name of node){Text};`. In the round brackets (), you type the name of the node so you can refer to it when drawing lines. In the curly brackets {}, you write the actual text.
- **Note**: Do not forget the ; it will not work without it!
- Let us take the following tree in the file `trees.tex` as a working example:[2]
- Delete the `\end{document}` command in between CUT POINT 1.

```
\begin{tikzpicture}
\tikzset{every tree node/.style={align=center,anchor=north}}
\Tree [.TP John\sub{2} [.T\1 T [.$v$P t\sub{2}
[.$v$\1 $v$ [.VP V\\{like} DP\\{linguistics}  ] ] ] ] ]
\end{tikzpicture}
```

[2]To make things simpler, I defined a command `\sub` that creates subscripts. This can be found in the preamble of your tex file.



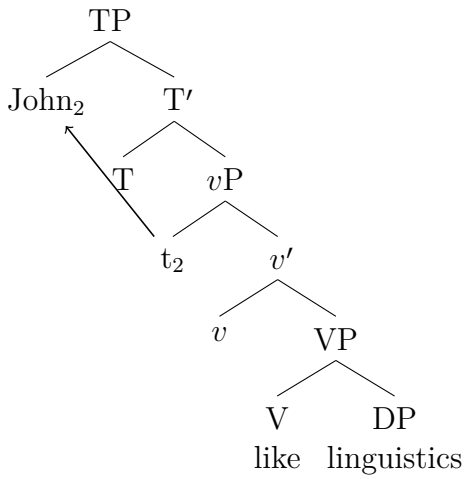
- The first arrow we want to draw is from the trace t_2 to $John_2$. So, we need to define these two items in the tree as nodes.
- Following the format above, `John$_2$` will be replaced by `\node(x){John\sub{2}};` and `t$_2$` will be replaced by `\node(y){t\sub{2}};`. Your code should look like this:

```
\begin{tikzpicture}
\tikzset{every tree node/.style={align=center,anchor=north}}
\Tree [.TP \node(x){John\sub{2}}; [.T\1 T [.$v$P \node(y){t\sub{2}};
[.$v$\1 $v$ [.VP V\\{like} DP\\{linguistics}  ] ] ] ] ]
\end{tikzpicture}
```

- If you recompile at this point, you will see that nothing has changed. Now, we need to tell `tikz-qtree` to draw an arrow between the two.
- The basic syntax for drawing arrows involves the command

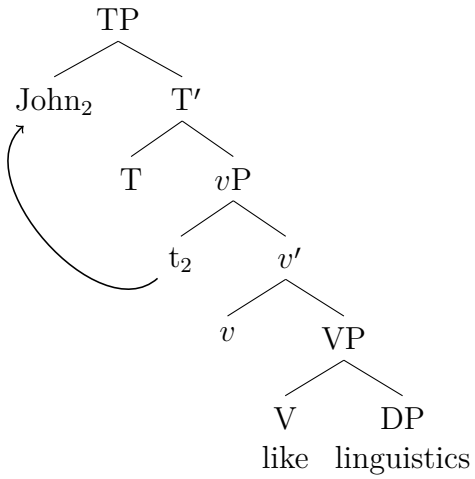
```
\draw[semithick, <-] (node1) to (node2);
```

```
\begin{tikzpicture}
\tikzset{every tree node/.style={align=center,anchor=north}}
\Tree [.TP \node(x){John\sub{2}}; [.T\1 T [.$v$P \node(y){t\sub{2}};
[.$v$\1 $v$ [.VP V\\{like} DP\\{linguistics}  ] ] ] ] ]
\draw[semithick, <-] (x) to (y);
\end{tikzpicture}
```



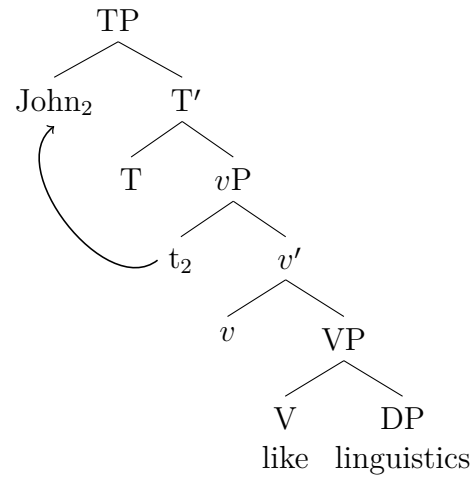
- This does not look too nice, so let us make a curved line. There are a number of ways to do this in TikZ, I prefer the following:

```
\draw[semithick, <-] (x) to [bend right=90] (y);
```



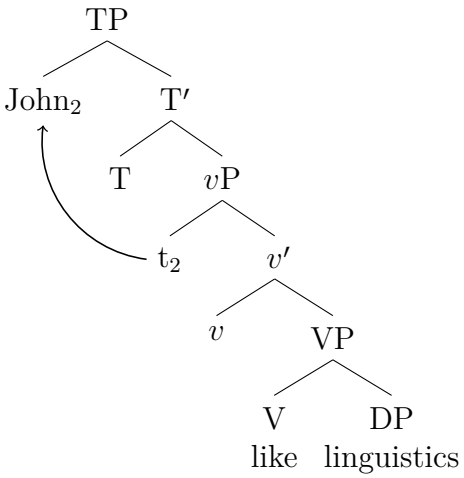
- We can add or modify the arrows in a number of ways. For example, the position of the start and end points of the arrows can be set using compass directions (e.g. north, south).

```
\draw[semithick, <-] (x.south) to [bend right=90] (y.west);
```



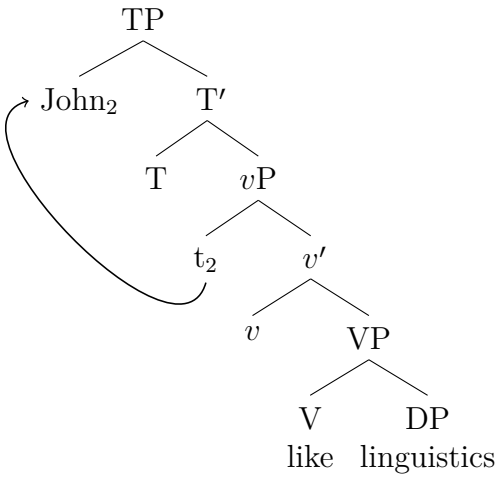
- The angle on the arrow seems a bit wrong here to let's modify it by reducing the value of `bend right`:

```
\draw[semithick, <-] (x.south) to [bend right=45] (y.west);
```



- Here is another option just to compare:

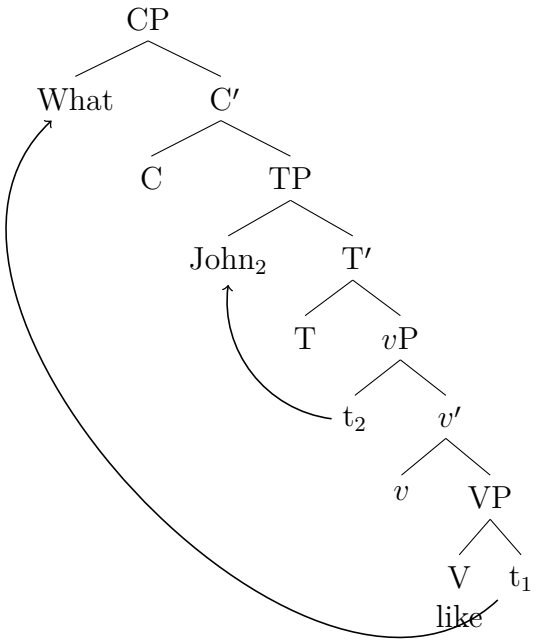
```
\draw[semithick, <-] (x.west) to [bend right=120] (y.south);
```



- The paths that arrows can take can also be modified. Delete CUT POINT 2 and compile.
- This tree contains two new nodes `wh` and `t1` let us draw a line between them by writing:

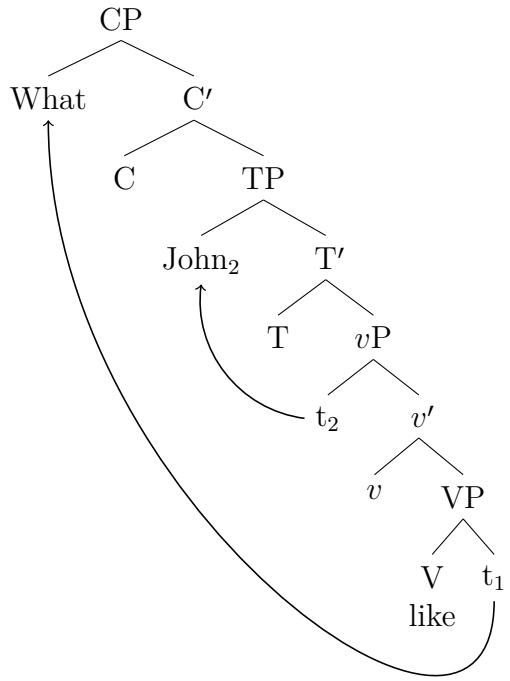
```
\draw[semithick, <-] (wh) to [bend right=90] (t);
```

- As you can see, the result is not good so we need to define the path of movement a little more.



- We can do this using `controls` that either specify specific coordinates in the picture (see below) or specifying compass commands again. These controls act as magnets pulling the line in different directions to create a curve. It is complicated to understand fully — Just play around with it!

```
\begin{tikzpicture}
\tikzset{every tree node/.style={align=center,anchor=north}}
\Tree [.CP \node(wh){What}; [.C\1 C [.TP \node(x){John\sub{2}}; [.T\1 T
[.$v$P \node(y){t\sub{2}};
[.$v$\1 $v$ [.VP V\\{like} \node(t){t\sub{1}};  ] ] ] ] ] ]]
\draw[semithick, <-] (x.south) to [bend right=45] (y.west);
\draw[semithick, <-] (wh.south)..controls +(south:5) and +(south:3)..(t.south);
\end{tikzpicture}
```

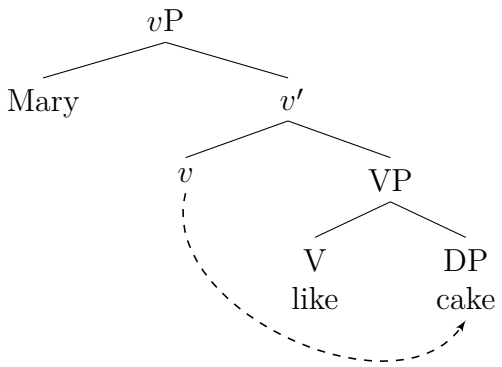


Additional exercise: Modify this tree using some of the options in the basics section such as `sibling distance` etc. You may have to modify the lines as a result!

3.1 Customizing arrows

- There are a number of additional ways to customize your arrows.
- If you want a nicer arrowhead, add `arrows` to `\usetikzlibrary{}` in the preamble and then add the option `\begin{tikzpicture}[>=latex']` to your tree.
- To get a dashed line add the option `dashed` to the options following the `\draw` command as follows `\draw[semithick, <-,dashed]`.
- There are a vast number of other options (e.g. changing colours) – explore for yourself!
- One last important part of arrow-drawing is adding annotations to arrows. Remove CUT POINT 3.
- Imagine we want to draw an arrow from *v* to *cake* indicating accusative case assignment.

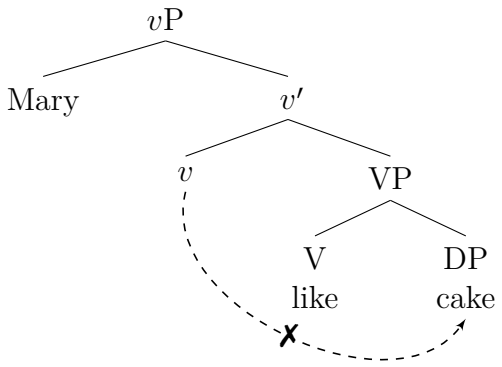
```
\begin{tikzpicture}[>=latex']
\tikzset{every tree node/.style={align=center,anchor=north}}
\tikzset{level 1+/.style={sibling distance=2\baselineskip}}
\Tree  [.$v$P Mary
[.$v$\1 \node(v){$v$}; [.VP V\\{like} \node(c){DP\\cake};  ] ] ]
\draw[semithick,dashed,->] (v.south) to [bend right=80] (c.south);
\end{tikzpicture}
```



- Assume that we first want to add a cross to the line, indicating that this operation is not possible. This is very simple, we simply add `node {\ding{55}}` before the final (node):[3]

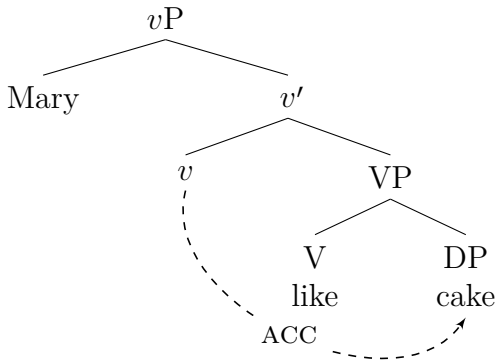
```
\draw[semithick,dashed,->] (v.south) to [bend right=80] node {\ding{55}}
(c.south);
```

[3]Note: this command requires the package `pifont`.



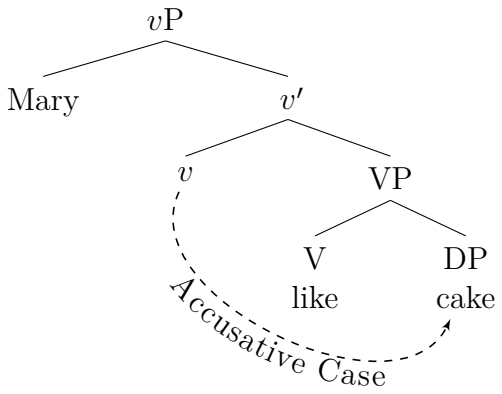
- Now, what if we want to indicate accusative case assignment by adding ACC to the arrow so that it breaks the line?
- Again, there are a number of ways to achieve this. I will show you the simplest. You can add some options after the `node` command that will create a white background around the text (`[midway,fill=white]`)

```
\begin{tikzpicture}[>=latex']
\tikzset{every tree node/.style={align=center,anchor=north}}
\tikzset{level 1+/.style={sibling distance=2\baselineskip}}
\Tree  [.$v$P Mary
[.$v$\1 \node(v){$v$}; [.VP V\\{like} \node(c){DP\\cake};  ] ] ]
\draw[semithick,dashed,->] (v.south) to [bend right=80] node [midway,fill=white]
{\textsc{acc}} (c.south);
\end{tikzpicture}
```



- Using `[above]` or `[below]` after `node` will place the label either above or below the line.
- What about if we want bend text so that it follows a curved line? This is a little more complicated and requires `\usetikzlibrary{decorations.text}` and the code below.

```
\begin{tikzpicture}[>=latex']
\tikzset{every tree node/.style={align=center,anchor=north}}
\tikzset{level 1+/.style={sibling distance=2\baselineskip}}
\Tree  [.$v$P Mary [.$v$\1 \node(v){$v$}; [.VP V\\{like}
\node(c){DP\\cake};  ] ] ]
\def\myshift#1{\raisebox{1ex}}
\draw [semithick,dashed,->,postaction={decorate,decoration={text along path,
text align=center,text={|\sffamily\myshift|\textit{Accusative case}}}}]
(v) to [bend left=45] (c);
```

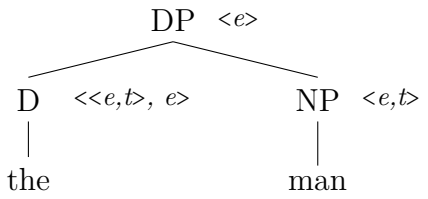


3.2 Node options

- If you want to add semantic denotations to your nodes, this is done as follows:
- Rather than writing `[.DP` as the label, use

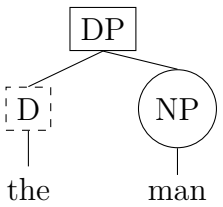
```
[. \node[label={right:Text to go next to your node}]{DP};
```

```
\begin{tikzpicture}
\tikzset{level 1+/.style={sibling distance=2\baselineskip}}
\Tree [. \node[label={right:\textit{\footnotesize $<$e$>$}}]{DP};
[. \node[label={right:\textit{ \footnotesize $<$$<$e,t$>$, e$>$}}]{D}; the ]
[.\node[label={right:\textit{\footnotesize $<$e,t$>$ }}]{NP}; man ]]
\end{tikzpicture}
```



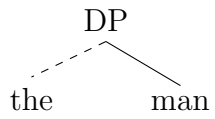
- Drawing a box around a node is done in a similar fashion using the `\node[]` format. The option `draw` will draw a box around the node in question and adding `circle` will draw a circle. Colours and `dashed` can also be added.

```
\begin{tikzpicture}
\tikzset{level 1+/.style={sibling distance=2\baselineskip}}
\Tree [. \node[draw]{DP};  [. \node[draw,dashed]{D}; the ]
[.\node[draw,circle]{NP}; man ]]
\end{tikzpicture}
```

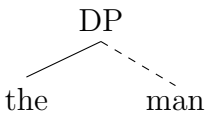


- It is also possible to modify individual branches using the `\edge` command. Placing `\edge[dashed];` before or after the first daughter will one of the branches dashed:

```
\Tree [.DP \edge[dashed]; the man ]
```




```
\Tree [.DP  the \edge[dashed]; man  ]
```



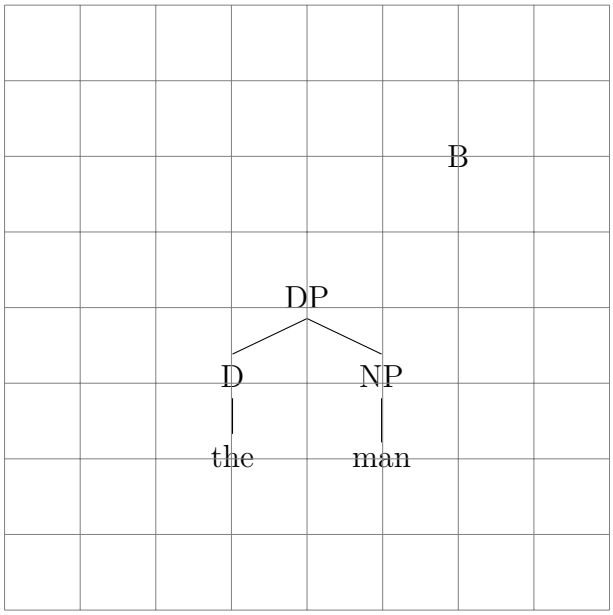
Appendix: More tips and tricks with TikZ

- TikZ allows one to draw almost anything. Everything is based is on a set of coordinates this can be made explicit by adding the a grid using

  `\draw[help lines] (coordinate) grid (coordinate)!.`

- You can then specify nodes to draw things at various points on the tree.

```
\begin{tikzpicture}
\tikzset{level 1+/.style={sibling distance=2\baselineskip}}
\Tree [.DP  [.D the ] [.NP man ]]
\node (x) at (2,2) {B} ;
\draw[help lines] (-4,-4) grid (4,4)
\end{tikzpicture}
```

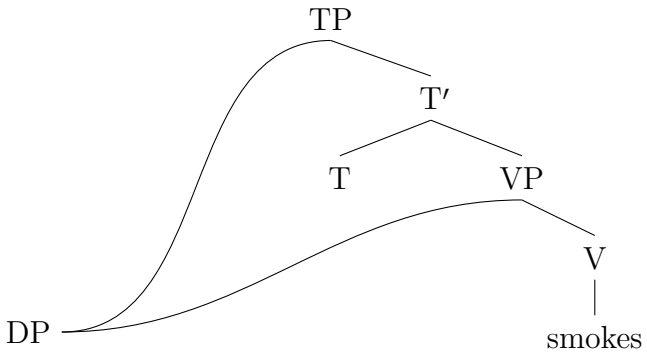


- Note that `tikz-qtree` draws trees at the coordinate (0,0). You can draw things around your tree by defining nodes.
- For example, adding a node at (2,2) using `\node (label) at (coordinate) {name-of-node};`.
- This opens up a number of further possibilities to draw trees and figures exactly the way you want — Be creative!

Some ideas of what is possible

```
\ex.
\leavevmode\vadjust{\vspace{-\baselineskip}}\newline
\begin{tikzpicture}
\tikzset{level 1+/.style={sibling distance=2\baselineskip}}
\Tree [.\node(z){TP}; \edge[white]; ~  [.T\1 T [.\node(y){DP}; \edge[white];
~ [.V smokes ]]]]
\node (x) at (-4,-4) {DP} ;
\draw (x.east) to [out=0, in=180] (y.south);
\draw (x.east) to [out=0, in=180]  (z.south);
%\draw[help lines] (-4,-4) grid (4,4)
\end{tikzpicture}
```

(1)




```
\begin{tikzpicture}
\tikzset{every tree node/.style={align=center,anchor=base}}
\tikzset{level 1+/.style={level distance=2\baselineskip}}
\tikzset{frontier/.style={distance from root=6\baselineskip}}
\Tree [.VP [.NP John ] [.V\1 [.V ate ] [.NP meat  ] ]]

\begin{scope}[grow'=up,yshift=-6.11cm]
\Tree [.VP [.NP John ] [.V [.V ate ] [.N meat ] ]]
\end{scope}
\end{tikzpicture}
```