Generating simple Dependency Trees with XeLaTeX

Johannes Heinecke
johannes.heinecke@orange.fr

February 8, 2022

1 Introduction

This package generates simple dependency trees (in contrast to dependency graphs produced by `tikz-dependency package`). It depends on `tikz` and some of its libaries (`arrows`, `positioning`)

The package is loaded by `\usepackage{deptree}`. Do not forget to load `\usepackage{xcolor}` to be able to redefine colours

Dependency tries are technically speaking a `tikzpicture`, so all tree commands must be in an `tikzpicture`-environment. The size of the horizontal and vertical steps can be given by the `tikz`-options `x=` and `y=`:

```
\begin{tikzpicture}[x=20mm,y=20mm]
\end{tikzpicture}
```

2 Declaring dependency relations

The root is declared by `\root{`*`word position`*`}{`*`form`*`}{`*`POS`*`}`. The POS field can be split into several lines (e.g.to give UPOS and XPOS)) by `\\`. Non root word forms are given by

`\dep{`*`head pos`*`}{`*`word pos`*`}{`*`vertical pos`*`}{`*`form`*`}{`*`POS`*`}{`*`dep rel`*`}`.

- `head pos` is the horizontal position of the head in the sentence
- `word pos` is the horizontal position of the dependent in the sentence

- `vertical pos` is the vertical position of the dependent (counted from the root)

A bottom line which repeats the words of the sentence, can be added by specifying the vertical position of this line `\setbottom{`*`vert pos`*`}`. This value will be used for all following trees. So it either has to be set to an correct value for each sentence or to 0 to deactivate the bottom line.

For instance, the code in figure 1 results in figure 2:

```
\setbottom{4} % set to 0 to hide bottom line of forms
\begin{tikzpicture}[x=20mm,y=20mm]
\root{2}{gefais}{VERB}
\dep{2}{1}{2}{Mi}{PART}{advmod}
\dep{2}{3}{2}{i}{PRON}{nsubj}
\dep{2}{5}{2}{ngheni}{NOUN}{ccomp}
\dep{2}{7}{2}{Nghaerdydd}{PROPN}{obl}
\dep{2}{8}{2}{.}{PUNCT}{punct}
\dep{5}{4}{3}{fy}{PRON}{nmod:obj}
\dep{7}{6}{3}{yng}{ADP}{case}
\end{tikzpicture}
```

Figure 1: dependency tree definition

The CoNLL-U editor (https://github.com/Orange-OpenSource/conllueditor) generates the tree definitions automatically from any CoNLL-U file.

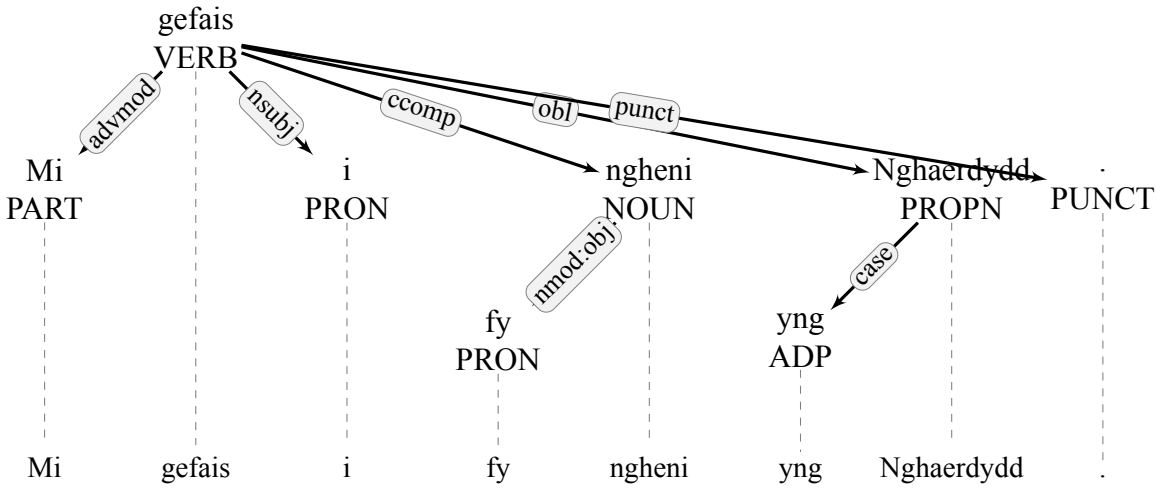

Figure 2: Default dependency tree

3 Customisation

The package has limited configuration options:

- `\setdeprelcolor{blue}` color of the lines linking head and dependant
- `\setdeprellabelcolor{red}` colour of the rectangle containing the dependency relation name
- `\setdeprelbgcolor{blue!15}` color of the background of the dependency relation name
- `\setdepreltextcolor{violet}`colour of the dependency relation name
- `\setdepreltextfont{\sf}` font for the dependency relation name
- `\setwordfont{\sf}` font for word nodes (form and POS, unless the latter is overriden)
- `\setposfont{\sf}` font for POS in word nodes
- `\setbottomwordfont{\sf}` font for words on the bottom line
- `\setmwtfont{\it\scriptsize}` font for multiword token (bottom line)

- `\setmwtlabelcolor{blue!10}` background colour for multiword token

The `set*font` commands accept any font command, including new fonts declared with XeLaTeX' `fontspec` package.

For instance, adding the the customization shown in figure 3 *before* the tree definition, results in figure 4:

```
\setdeprelcolor{blue}
\setdeprellabelcolor{black}
\setdepreltextcolor{blue!50!black}
\setdeprelbgcolor{blue!15}
\setdepreltextfont{\it\scriptsize}
\setwordfont{\large\sf}
\setposfont{\rm\small}
\setbottomwordfont{\footnotesize\sf}
\setmwtfont{\it\scriptsize}
\setmwtlabelcolor{blue!10}
```

Figure 3: customization

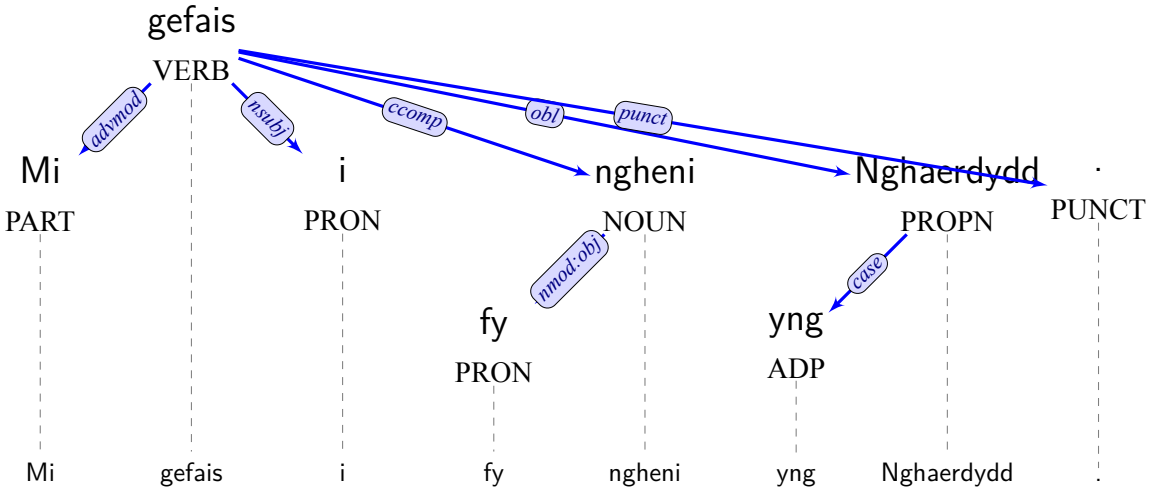

Figure 4: Customized dependency tree

Setting the label background color to an empty value, prints the dependency labels above the edges without box nor background (cf. figure 6)

```
\setdeprelbgcolor{}
\setdepreltextfont{\footnotesize}
```

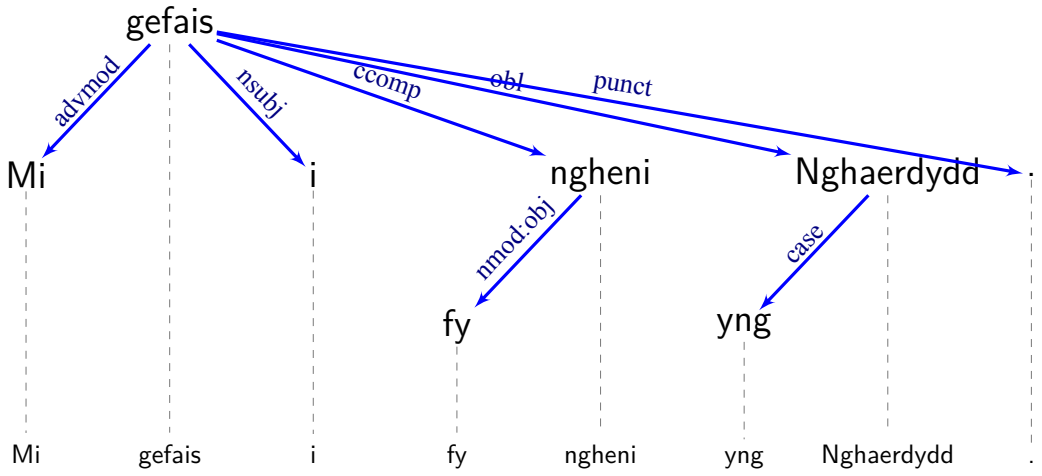

Figure 5: Customized dependency tree

The vertical position can be used to modify the layout of the trees. For instance if we change the line from figure 1 to `\dep{5}{4}{2.3}{fy}{PRON}{nmod:obj}` will lower the node of word *fy* slightly (cf. figure 6).

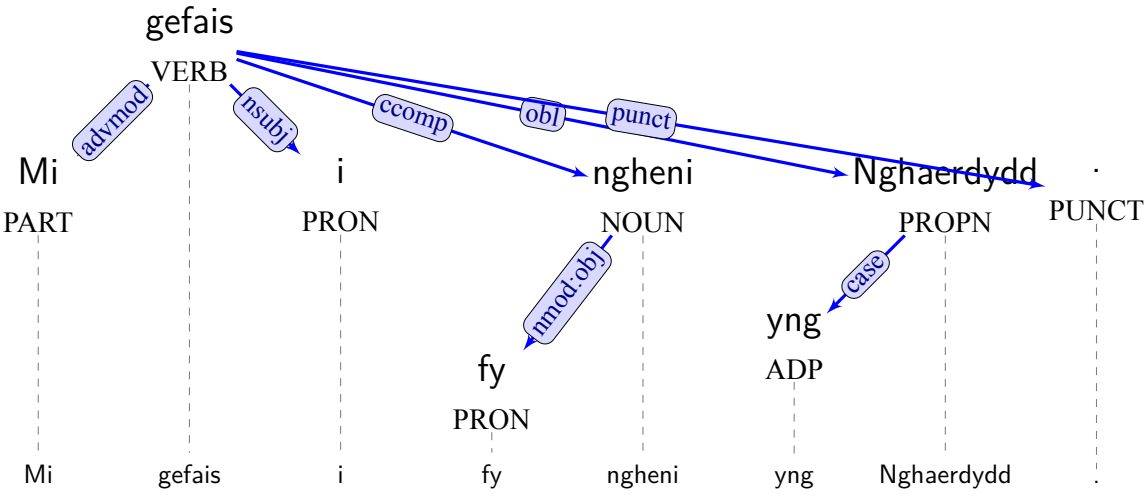

Figure 6: Vertical position modified

4 Multiword tokens

Multiword tokens (MWT) can be indicated by a `\mwt` command (e.g. figure 8):

`\mwt{`*`first word pos`*`}{`*`last word pos`*`}{`*`MWT`*`}`

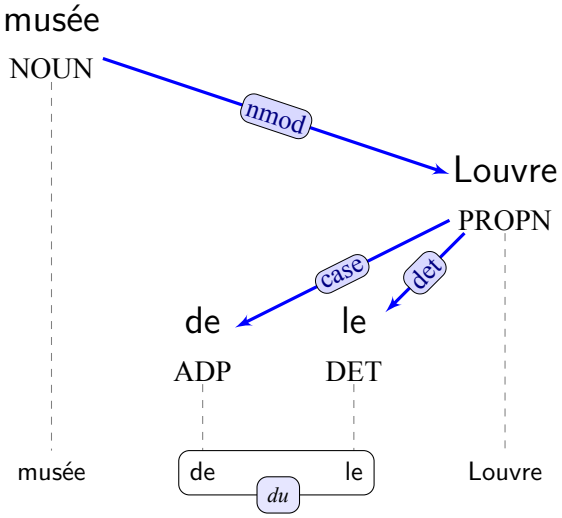

Figure 7: Multiword tokens

Small trees can be obtained by modifying the `x=/y=` options. Do not chose values to small (as I did here) without lowering font size accordingly, since the deprel labels will overlap with nodes.

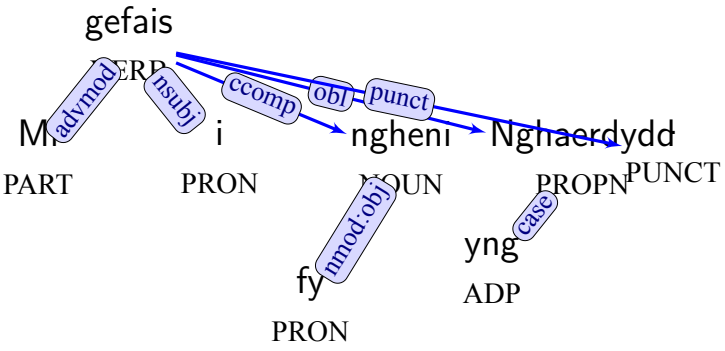

Figure 8: Small tree