

# Using RSyntaxTree or Syntax Tree Generator to make trees

Alan Munn  
Michigan State University

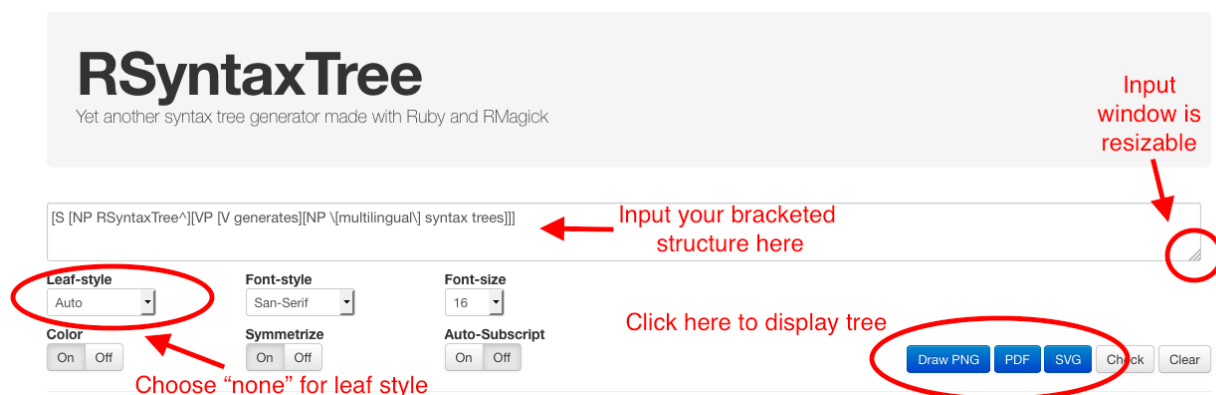
September 2020

## 1 Introduction

The [RSyntaxTree](#) and [Syntax Tree Generator](#) web sites provide a quick and easy tree drawing facility which is handy for simple trees. For more complex trees, you will be much better off using Overleaf. This guide is a quick introduction to using these sites (which are very similar to each other.) RSyntaxTree is nice because it allows you to save your image as a scalable PDF, which then allows you to resize the image in your document. Syntax Tree Generator only outputs PNG, but it is nice in other ways, however: it shows your tree as you draw it, which is very helpful, and it can even do arrows to connect nodes together. Both are very capable of drawing most kinds of trees.

## 2 RSyntaxTree

When you connect to [RSyntaxTree](#) you will see the following:



The site is very easy to use. You simply enter a bracketed structure and it turns it into a tree.

Here's an example which you can cut and paste into the site:

```
[TP
  [DP [D the ] [NP [N man ]]]
  [ 'T
    [T will ]
    [VP
      [V eat ]
      [DP [D a ] [NP [N burger ]]]
    ]
  ]
]
```

```

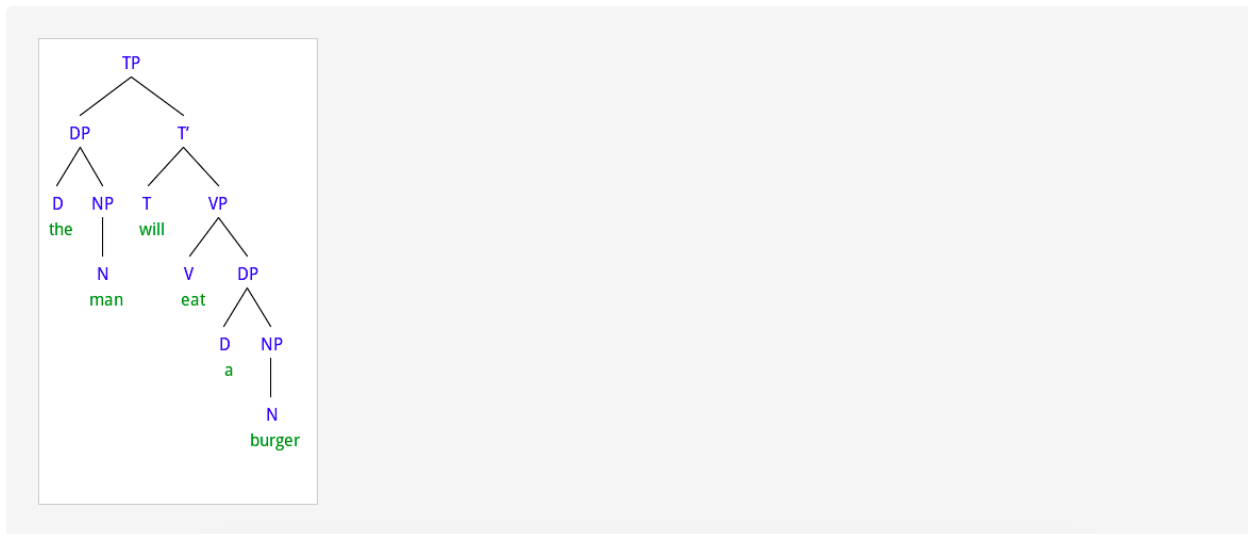
[TP
  [DP [D the ] [NP [N man ]]]
  [T'
    [T will ]
    [VP
      [V eat ]
      [DP [D a ] [NP [N burger ]]]
    ]
  ]
]

```

**Leaf-style**    
**Font-style**    
**Font-size**

**Color**     
**Symmetrize**     
**Auto-Subscript**

When you click on the PNG button, the following should appear:



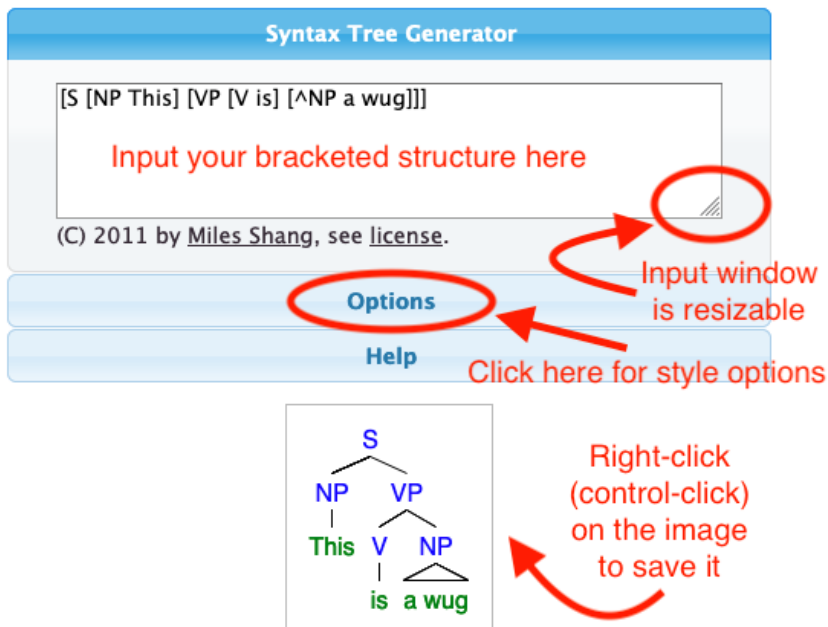
If you click on the PDF button, you can save the tree as a PDF image for insertion into a Word document (preferred method.)

## 2.1 Input options

Apart from the various font and colour options, RSyntax tree allows you to choose the format of the branches between the words. This option is called “Connector shape”. By default this is set to “Auto”, which automatically creates triangles for groups of words in a single pair of brackets. Unfortunately, this mode also puts a line between the word and its category label, which is not linguistically correct. You should therefore set the Connector shape to “None”. This removes removes the unwanted lines between words and their categories.

### 3 Using Syntax Tree Generator

When you connect to [Syntax Tree Generator](#) you will see the following:

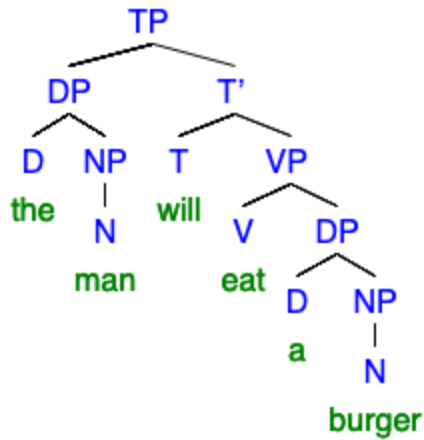


Like RSyntaxTree, the site simply turns a bracketed structure into a tree. Syntax Tree Generator is very forgiving, and will complete closing brackets for you, which allows you to see your tree as it is being built. For this reason, it is a very nice tree generator.

Here's an example which you can cut and paste into the site:

```
[TP
  [DP [D the ] [NP [N man ]]]
  ['T
    [T will ]
    [VP
      [V eat ]
      [DP [D a ] [NP [N burger ]]]
    ]
  ]
]
```

I've used an option to remove lines from connecting the words to their categories, so the output that I get looks like this:



### 3.1 Options

If you click on the Options you will see the following:

There are two important things to know about the options. First, you should make sure that the Terminal Lines button is not clicked (i.e., white, not blue). This removes lines connecting words to their categories. It is linguistically incorrect to have such lines, so you should not use them in your trees.

The second useful option removes the colours from the tree. This might be useful if you are inserting a tree into a document that needs to look more professional.

The options panel also allows you to change the fonts, and use sliders to change the size of the tree, both vertically and horizontally.

### 3.2 Input conventions: triangles and arrows

One of the nice things about Syntax Tree Generator is that it can represent groups of words using a triangle, and also can draw arrows between two elements to show movement. Here are some examples of how to do that.

To make a triangle, simply prefix the node label with the ^ character:

```
[DP[D the ][^NP very hungry caterpillar]]
```

This produces:

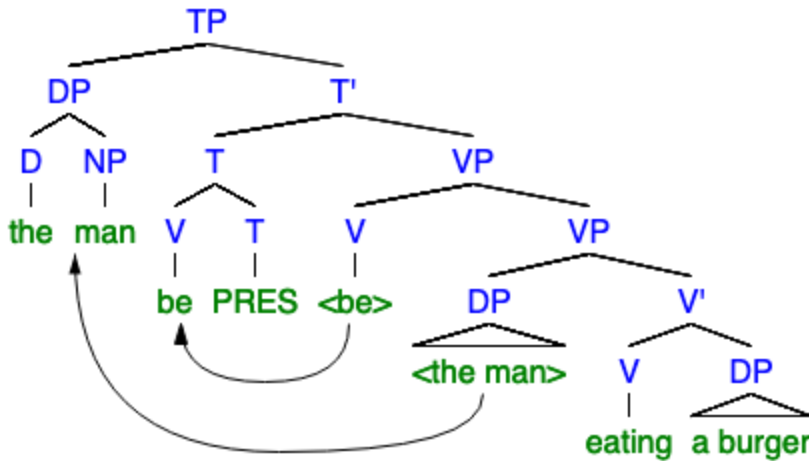


The syntax for connecting two elements with an arrow is slightly more complicated, but when you have learned about movement its syntax should be quite intuitive:

The node label that the arrow points to receives a suffix consisting of the \_ character and a unique name (which you give it). This name is used inside the node containing the place where the arrow points from. Here's an example with two movements marked. This example also shows how you can use the \_ character to mark node labels with subscripted numbers.

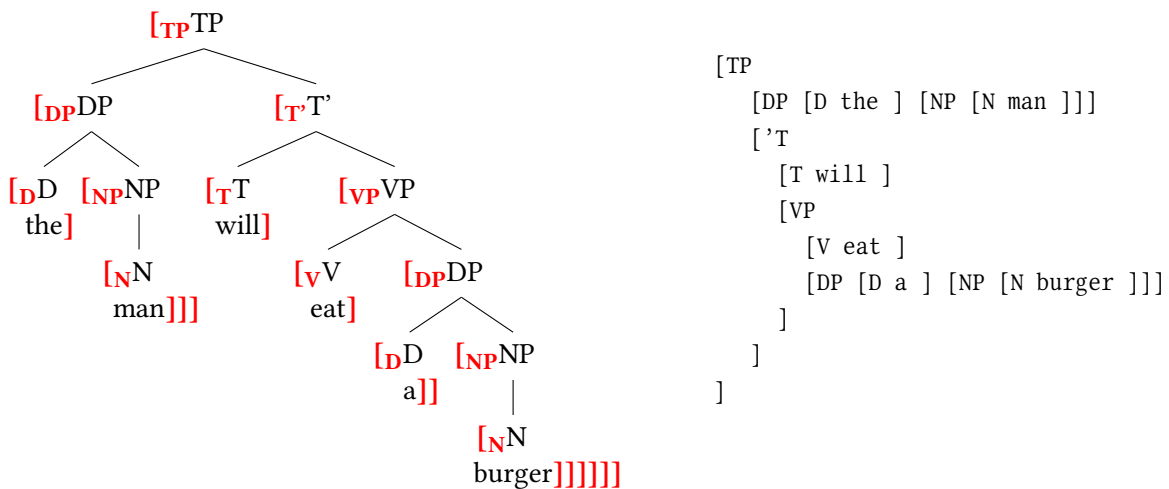
```
[TP [DP_subj [D the ] [NP man ]]
  [T' [T [V_aux be ] [T PRES ]]
    [VP [V <aux> <be>]
      [VP [^DP <the man> <subj>] [V' [V eating ] [^DP a burger ]]]]]]
```

In the example above, we have used the node label DP\_subj to say that this DP will be named “subj” and have an arrow pointing to it. Inside the VP, we have a corresponding tag <subj> which is the position where the arrow will originate. The node label V\_aux is similar: it is named be and in the corresponding lower V node there is a tag <aux> which marks where the arrow originates. Here's the output of this code:



#### 4 How to convert a tree to brackets

It's helpful to understand how to convert a tree into a bracketed structure. The basic idea is that every node in the tree gets a pair of brackets, and the label for the node is the label for the left bracket of the pair. So you should draw your tree by hand and then starting from the top, create brackets around the beginning and ends of nodes. For the sample tree shown above, we could do the following. In this tree, I've shown the bracketing in red as well as the tree.



#### 5 How to insert PDF images into MSWord documents

Once you've created your tree, you can easily insert it into a Microsoft Word document.

- On a Mac you can simply drag the PDF file onto your document or choose "Insert Picture from File".
- On a PC you can choose "Insert Object" and then select Adobe Acrobat Document and choose the file.