Dependency Tree

December 12, 2017

```
In [ ]: from datascience import *
        from urllib.request import urlopen
        import numpy as np
        from datascience import *
        import spacy
        % matplotlib inline
```

```
In [ ]: !python -m spacy download en
```

```
In [ ]: UN_articles_term="Legal Non-Legal Non-Legal Both Non-Legal Legal Both Legal Both Legal
        Legal Legal Legal Legal Legal Legal\
        Legal Legal Legal Legal Legal Legal Legal Legal Both Legal Both"
        UN_articles_terms_accurate = UN_articles_term.split()
```

```
In [ ]: UN_articles="Article1 Article2 Article3 Article4 Article5 Article6 Article7 Article8 Ar
        Article13 Article14 Article15 Article16 Article17 Article18\
        Article19 Article20 Article21 Article22 Article23 Article24 Article25 Article26 Article
        UN_articles_list = UN_articles.split()
```

```
In [ ]: comparision_table= Table().with_column('Names',UN_articles_list, 'Correct Identificati
```

```
In [ ]: my_features = make_array('shall','should','may','recognize','will','undertaking','enhan
```

```
In [ ]: article1_final2= """For the purpose of this Agreement, the definitions contained in Ar
        the Convention shall apply. In addition : \
        - 2 - \
        (a) " Convention " means the United Nations Framework Convention on\
        Climate Change, adopted in New York on 9 May 1992 ;\
        (b) " Conference of the Parties " means the Conference of the Parties to the \
        Convention ; \
         (c) Party means a Party to this Agreement."""
        UN_article1_list = article1_final2.split()
```

```
In [ ]: parsed_text = en_nlp(article1_final2)
        parsed_text
```

```
In [ ]: from spacy.symbols import nsubj, VERB
```

```
        SV = []
        for possible_subject in parsed_text:
            if possible_subject.dep == nsubj and possible_subject.head.pos == VERB:
                SV.append((possible_subject.text, possible_subject.head))

In [ ]: print (SV)

In [ ]: def tablefy(parsed_text):
            toks_tab = Table()
            toks_tab.append_column(label="Word", values=[word.text for word in parsed_text])
            toks_tab.append_column(label="POS", values=[word.pos_ for word in parsed_text])
            toks_tab.append_column(label="Lemma", values=[word.lemma_ for word in parsed_text]
            toks_tab.append_column(label="Stop Word", values=[word.is_stop for word in parsed_
            toks_tab.append_column(label="Punctuation", values=[word.is_punct for word in parse
            toks_tab.append_column(label="Space", values=[word.is_space for word in parsed_tex
            toks_tab.append_column(label="Number", values=[word.like_num for word in parsed_te
            toks_tab.append_column(label="OOV", values=[word.is_oov for word in parsed_text])
            toks_tab.append_column(label="Dependency", values=[word.dep_ for word in parsed_te
            return toks_tab

In [ ]: from spacy.symbols import nsubj, VERB

        SV = []
        for possible_subject in parsed_text:
            if possible_subject.dep == nsubj and possible_subject.head.pos == VERB:
                SV.append((possible_subject.text, possible_subject.head))

In [ ]: import spacy
        from nltk import Tree

In [ ]: import spacy
        from nltk import Tree


        en_nlp = spacy.load('en')

        doc = en_nlp("""For the purpose of this Agreement, the definitions contained in Articl
        the Convention shall apply. In addition : \
        - 2 - \
        (a) " Convention " means the United Nations Framework Convention on\
        Climate Change, adopted in New York on 9 May 1992 ;\
        (b) " Conference of the Parties " means the Conference of the Parties to the \
        Convention ; \
         (c) Party means a Party to this Agreement.""" )

        def to_nltk_tree(node):
            if node.n_lefts + node.n_rights > 0:
                return Tree(node.orth_, [to_nltk_tree(child) for child in node.children])
            else:
                return node.orth_


        [to_nltk_tree(sent.root).pretty_print() for sent in doc.sents]
```

```
In [ ]: article1_draft1="""For the purpose of this Agreement, the definitions contained in Arti
        COP 21 - Draft text agenda item 4 (b) - version 2\
        17 of 27\
        1. "Convention" means the United Nations Framework Convention on Climate Change, adopte
        May 1992;\
        2. "Conference of the Parties" means the Conference of the Parties to the Convention;\
        3. Party means a Party to this Agreement."""
```

```
In [ ]: en_nlp=spacy.load('en')

        doc =en_nlp("""For the purpose of this Agreement, the definitions contained in Article
        COP 21 - Draft text agenda item 4 (b) - version 2\
        17 of 27\
        1. "Convention" means the United Nations Framework Convention on Climate Change, adopte
        May 1992;\
        2. "Conference of the Parties" means the Conference of the Parties to the Convention;\
        3. Party means a Party to this Agreement.""")

        def to_nltk_tree(node):
            if node.n_lefts + node.n_rights > 0:
                return Tree(node.orth_, [to_nltk_tree(child) for child in node.children])
            else:
                return node.orth_


        [to_nltk_tree(sent.root).pretty_print() for sent in doc.sents]
```

```
In [ ]: """1. Parties recognize the importance of averting, minimizing and addressing loss\
        and damage associated with the adverse effects of climate change, including\
        extreme weather events and slow onset events, and the role of sustainable\
        development in reducing the risk of loss and damage.\
        2. The Warsaw International Mechanism for Loss and Damage associated with\
        Climate Change Impacts shall be subject to the authority and guidance of the\
        Conference of the Parties serving as the meeting of the Parties to this Agreement\
        and may be enhanced and strengthened, as determined by the Conference of the\
        Parties serving as the meeting of the Parties to this Agreement.\
        3. Parties should enhance understanding, action and support, including through\
        the Warsaw International Mechanism, as appropriate, on a cooperative and\
        facilitative basis with respect to loss and damage associated with the adverse\
        effects of climate change.\
        4. Accordingly, areas of cooperation and facilitation to enhance understanding,\
        action and support may include:\
        5. The Warsaw International Mechanism shall collaborate with existing bodies\
        and expert groups under the Agreement, as well as relevant organizations and \
        expert bodies outside the Agreement. """

In [ ]: en_nlp=spacy.load('en')

        doc =en_nlp("""1. Parties recognize the importance of averting, minimizing and addressi
        and damage associated with the adverse effects of climate change, including\
        extreme weather events and slow onset events, and the role of sustainable\
        development in reducing the risk of loss and damage. """ )

        def to_nltk_tree(node):
            if node.n_lefts + node.n_rights > 0:
                return Tree(node.orth_, [to_nltk_tree(child) for child in node.children])
            else:
                return node.orth_


        [to_nltk_tree(sent.root).pretty_print() for sent in doc.sents]

In [ ]: en_nlp=spacy.load('en')

        doc =en_nlp(""" The following text to be part of Article 4 alongside adaptation provisi
        1. [The Warsaw International Mechanism for Loss and Damage associated with Climate Chan
        shall serve this Agreement.\
        2. Upon the completion of its review at the twenty-second session of the COP,9 the Wars
        through the Financial Mechanism of the Convention.\
        3. A climate change displacement coordination facility shall be established under the
        displacement, migration and planned relocation.\
        4. A process to develop approaches to address irreversible and permanent damage resulti
        with a view to completing this process within four years.""")

        def to_nltk_tree(node):
            if node.n_lefts + node.n_rights > 0:
                return Tree(node.orth_, [to_nltk_tree(child) for child in node.children])
            else:
                return node.orth_


        [to_nltk_tree(sent.root).pretty_print() for sent in doc.sents]

In [ ]: from IPython.display import IFrame

In [ ]: IFrame('https://demos.explosion.ai/displacy/?text=For%20the%20purpose%20of%20this%20Ag

In [ ]: import spacy
        nlp = spacy.load('en')
        doc = nlp('First, I wrote some sentences. Then spaCy parsed them. Hooray!')
```