What can we learn from natural and artificial dependency trees ?

Marine Courtin, LPP (CNRS) - Paris 3 Sorbonne Nouvelle
Chunxiao Yan, Modyco (CNRS) - Université Paris Nanterre

SyntaxFest
Paris, August 26-30 2019

Summary

- Introduce several procedures for generating random syntactic dependency trees with constraints
- Create artificial treebanks based on real treebanks
- Compare the properties of theses trees (real / random)
- Try to find out how these properties interact and to what extent the relationship between them is formally constrained and/or linguistically motivated.

What do we have to gain from comparing natural and random trees ?

Motivations

- Natural syntactic trees are nice but :
  - Very complex
  - It’s hard to understand how some property influences other properties
  - They mix formal and linguistic relationship between properties
- We want to find out why some trees are linguistically implausible ? i.e what makes these trees special compared to random ones

Motivations

- Natural languages have special syntactic properties and constraints that imposes limit on their variation.
- We can observe these properties by looking at natural syntactic trees.
- Some of the properties we observe might be artefacts : not properties of natural langages but properties of trees themselves (mathematical object).
  - → By also looking at artificial trees we can distinguish between the two

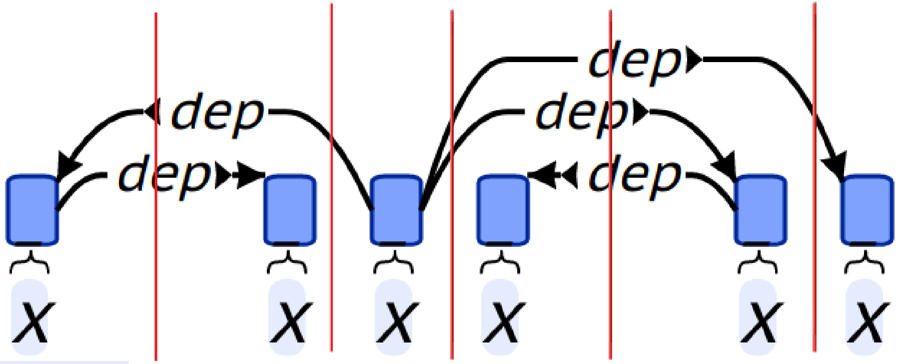
Methods and data preparation

Data

- Corpus : Universal Dependencies (UD) treebanks (version 2.3, Nivre et al. 2018) for 4 languages: Chinese, English, French and Japanese.
- We removed punctuation links.
- For every original tree we create 3 alternative trees.

Features

Feature name	Value
Length	6
Height	3
Maximum arity	3
Mean dependency distance (MDD)	(2+1+1+2+3)/5=1.8
Mean flux weight (MFW)	(1+1+1+2+1)/1.2

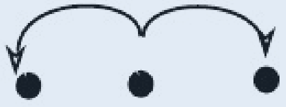


→ all related to syntactic complexity

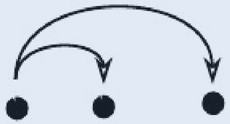
Typology of local configurations

25 possible trigram configurations : grouped into 4 types.

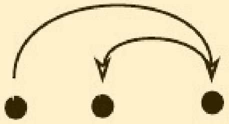
balanced



bouquet



a ← b → c



zigzag

chain

a → b → c

Introduces height in both directions

Introduces height in one direction

Hypotheses ?

- Tree length is positively correlated with other properties.
- Particularly interested in the relationship between mean dependency distance and mean flux weight.
  - As tree length increases ⇒ the number of possible trees increases
    ⇒ more complex trees with longer dependencies (higher MDD) and more nestedness (higher mean flux weight)
  - An increase in nestedness ⇒ more descendents between a governor and its direct dependents ⇒ increase in mean dependency distance.

Generating random trees

Generating random trees

We test 3 possibilities :

- *Original-random* : original tree, random linearisation
- *Original-optimized* : original tree, « optimal » linearisation
- *Random-random* : random tree, random linearisation

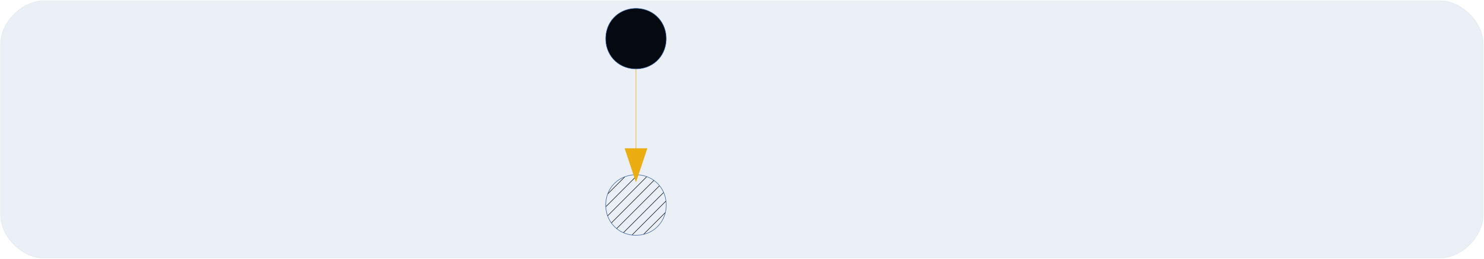
One more constraint : we only generate projective trees.

→ We expect that natural trees will be the furthest away from random-random and somewhere between original-random and original-optimized.

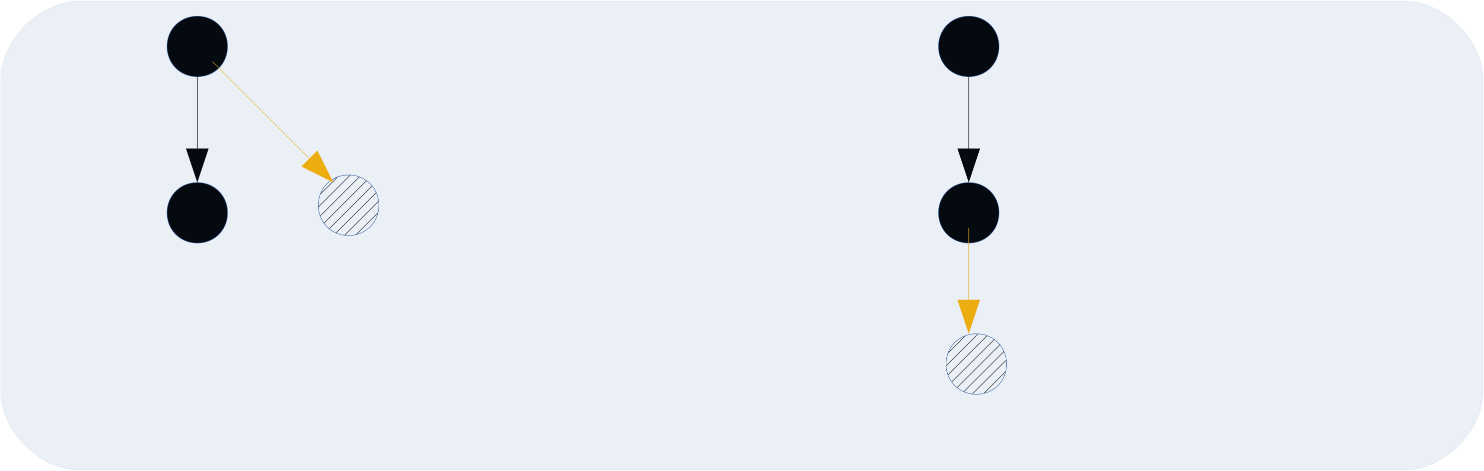
Random tree

1.

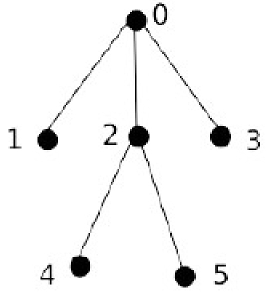
2.



3.

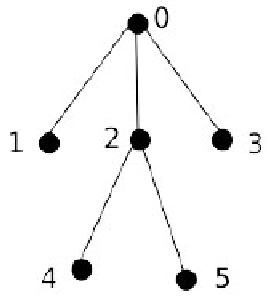


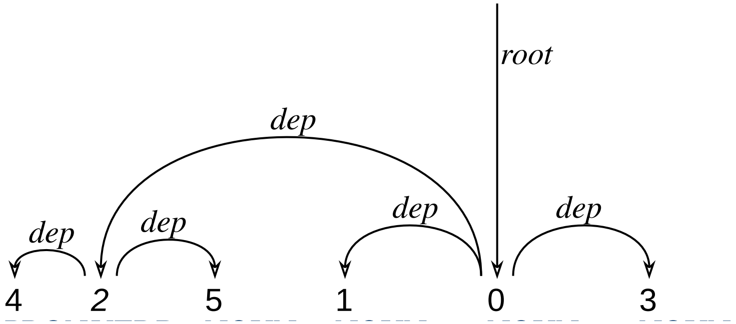
Random projective linearisation



1. Start at the root
2. Randomly order direct dependents → [2,1,3]
3. Select a random direction for each → [« left », « left », « right »] → [1203]
4. Repeat steps 2-3 until you have a full linearization → [124503]

Optimal linearisation





1. Start at the root
2. Order direct dependents by their decreasing number of descendant nodes → [1,3,2]
3. Linearize by alternating directions (eg. left, right, left) → [2103]
4. Repeat until all nodes are linearized → [425103]

Generating random trees

- Why this particular algo ?
  - Separates generation of the unordered structure and of the linearisation → this allows us to change only of the two steps.
  - Easily extensible, we have the possibility to add constraints :
    - Set a parameter for the probability of a head-final edge
    - Set a limit on lenth, height, maximum arity for a node..
    - Set a limit on flux weight (can we actually do this?)

Results

Results on correlations

- Non surprising results :
  - length/height :
    - **strong** in both artificial and real → formal relationship, slightly intensified in non-artificial trees
    - Zhang and Liu (2018) : the relation can be described as a powerlaw function in English and Chinese ; interesting to look if the same thing can be found in artificial trees
  - MDD/MFW :
    - Strong in both real and artificial treebanks.
- Interesting results :
  - MDD/height is **stronger** in artificial than real treebanks.
  - MDD/MFW **is stronger** in artificial than real treebanks.

Distribution of configurations

Non-linearized case :

Potential explanations for the original distribution ?

- b ← a → c is favoured because it contains the « balanced » configuration, i.e the optimal one for limiting dependency distance.
- a → b → c is disfavoured because it introduces too much height.

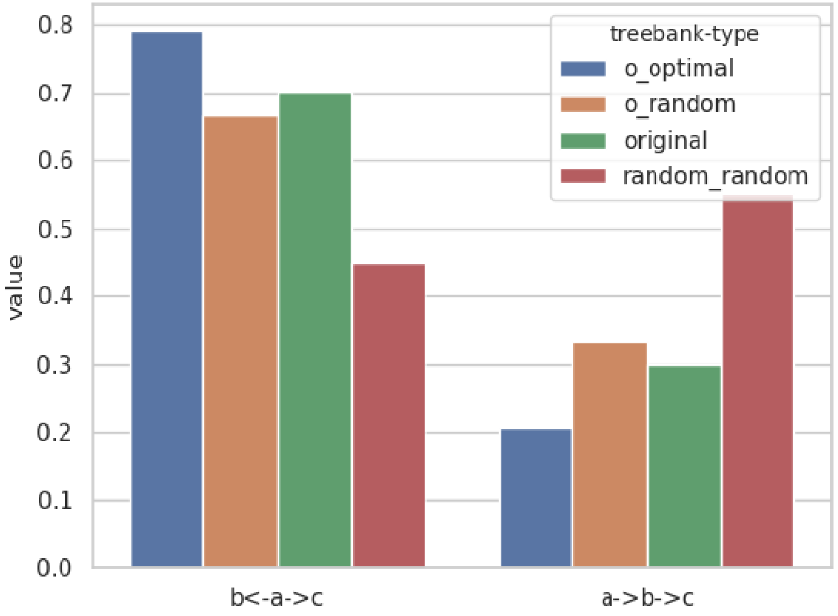


Figure 3: Non-linearized trigram configurations distribution for French

Distribution of configurations

- *Random random :*
  - slight preference for “chain” and “zigzag” : this is probably a by-product of the preference for b←a→c configurations rather than a → b → c.
  - inside each group (“chain” and “zigzag” / “bouquet” and “balanced”) the distribution is equally divided.
- *Original optimal* :
  - very marked preference for “balanced”.

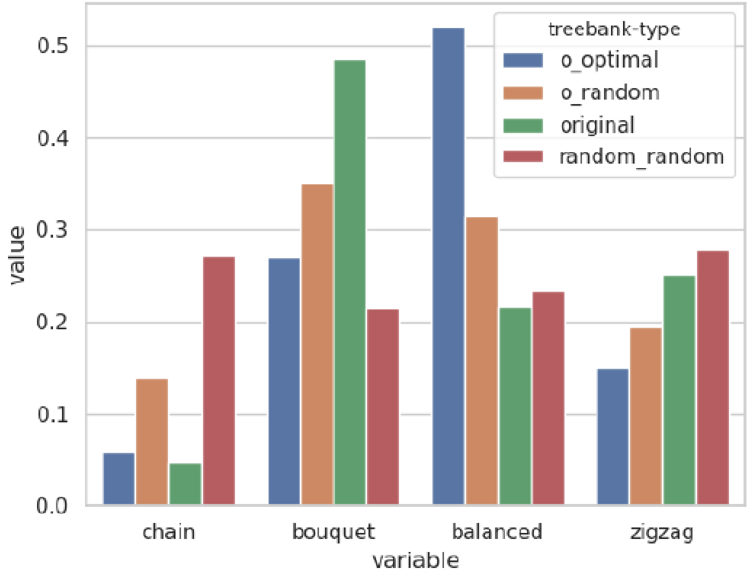


Figure 4: Trigram configurations distribution for French

Distribution of configurations

- *Original trees* :
  - Contrary to the potential explanation we advanced for the high frequency of b←a→c configurations, “balanced” configurations are not particularly frequent in the original trees.
  - The bouquet configuration is the most frequent, and it is much more frequent in the original trees than in the artificial ones.

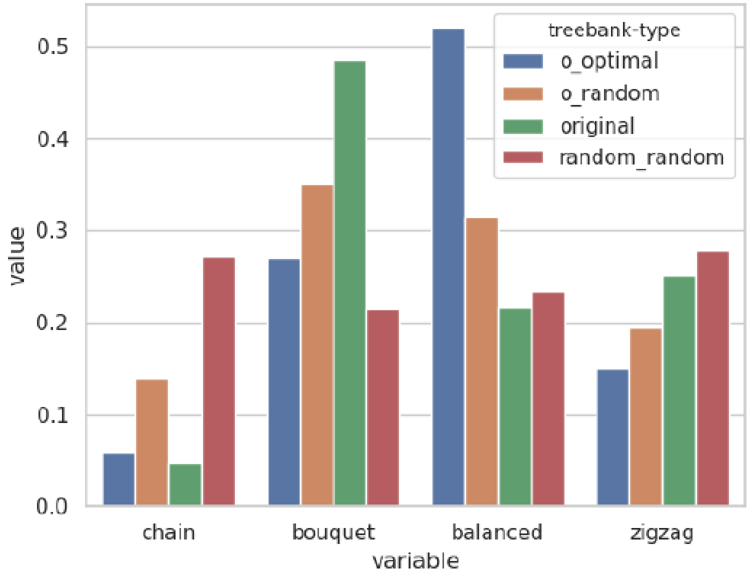


Figure 4: Trigram configurations distribution for French

Limitations

- We only generated projective trees.
- We looked at local configurations instead of all subtrees.
- Linear correlation may not be the most interesting observation :
  - The relationship between properties of the tree is probably not linear
  - We can directly look at the properties themselves and compare groups to see where original trees fit compared to all random groups.

Future work

- Compare directly the properties of the trees from the different groups. Which groups are more distant / similar ?
- Build a model to predict features of the tree
  - Which features can we predict from which combinations of features ?
  - Are natural trees more predictible ? They represent a smaller subset, so they should (?)
- Study the effects of the annotation scheme
  - How will our results be affected if we repeat the same process using an annotation scheme with functional heads ?