TreeCloud & Unitex: an increased synergy
Claude Martineau

To cite this version:
Claude Martineau. TreeCloud & Unitex: an increased synergy. ECLAVIT Workshop, Nov 2017, Champs-sur-Marne, France. 2017. hal-01702091

HAL Id: hal-01702091
https://hal-upec-upem.archives-ouvertes.fr/hal-01702091
Submitted on 19 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
TreeCloud & Unitex: an increased synergy
Claude Martineau
Projet PEPS CNRS/UPE Eclatvi

TreeCloud is a tree cloud visualization of a text
TreeCloud builds a tree visualization of a text, which looks like a tag cloud where the tags are displayed around a tree to reflect the co-occurrence distance between the words in the text.

The adjacent grammar tree gives an overview of Barack Obama’s 2008 presidential campaign speeches.

Tree construction process

- Input text
- Remove empty words
- Word selection
- Co-occurrence search
- Define distance between words
- Build the tree
- Display the tree

The stepword list contains grammatical words or auxiliary words in a language. For example, in English: the, a, an, this, that, be, have, may, etc. The stepwords are usually the most common words in a language. To build a meaningful distance matrix, all these words must be firstly removed from the input text.

Whenever a sliding window of analysis is used, its size (i.e. a number of words) must be given as parameter.

Co-occurrence and distance calculation

Given two words A and B:
- \( O_{\text{in}} \): observed number of sliding windows containing both A and B
- \( O_{\text{AB}} \): observed number of sliding windows containing A and B
- \( O_{\text{A}} \): observed number of sliding windows containing A
- \( O_{\text{B}} \): observed number of sliding windows containing B

The following variables are defined:
- \( K = O_{\text{AB}} - O_{\text{A}} \text{x} O_{\text{B}} \): number of sliding windows containing A and B
- \( E = O_{\text{A}} \text{x} O_{\text{B}} \): expected number of sliding windows containing A and B

The definitions of co-occurrence formulas are the following:

- \( C_{\text{in}} = \frac{2 \times K}{E} \): observed number of sliding windows containing both A and B
- \( C_{\text{AB}} = \frac{K}{E} \): observed number of sliding windows containing A and B
- \( C_{\text{A}} = \frac{O_{\text{A}}}{E} \): observed number of sliding windows containing A
- \( C_{\text{B}} = \frac{O_{\text{B}}}{E} \): observed number of sliding windows containing B
- \( C_{\text{AB}} = \frac{O_{\text{AB}}}{E} \): observed number of sliding windows containing A and B

Several versions of TreeCloud

Downloadable version in Python

- TreeCloud 1.3 for Windows, Linux, Mac developed by Philippe Gambette and Fabrice Teligny.
- Use of SplitsTree 4.10 to draw the tree

On-line version in C

- 2009: 2nd version developed by Jean-Baptiste Bonnemps
- 2012: Transition to Unix developed by Claude Martineau
- 2014: 1st implementation of Unitex developed by Claude Martineau

UniteX/GramLab is a corpus analyser and annotation tool

- Based on Automata and RTNs with outputs
- Multilingual: Up to 22 languages (French, English, ..., Greek, ... Korean, Thai)
- Unicode: 3.0 (UTF16, UTF64, UTF848)
- Cross-platform: Linux, macOS, Windows

Open source: https://github.com/UnitexGramLab
Website and binary installations: http://unitexgramlab.org

How and Why to plug Unitex into TreeCloud?

1) UniteX transforms the input text into a new text with all the forbidden/stopwords replaced by the XXX word
2) The new text is sent to TreeCloud with XXX as the unique forbidden word (the unique word in the stoplist)

Get a larger and more accurate coverage of forbidden words

Under development since 2001 by a group of passionate volunteers

Unitex/GrampLab uses linguistic resources:

- DELA (ADL electronic dictionaries)
- LXX2 (formalization of the lexicon)
- NodoLaME (LADL electronic dictionaries)
- ANTIpERSO (lexical mask in the last box) matches Multilingual oddsratio
- DELA (lado) electronic dictionaries

Several ways to use UniteX/GramLab

- Two interfaces written in JAVA:
  - UniteX IDE (classic)
  - GramLab IDE (project-oriented)

- Command line or system calls with Perl, Python, etc.

- Use the API C and JAVA (JNI) that provides access to:
  - a virtual file system
  - a persistence layer for resources (alphabets, dictionaries and corpora)

- Consequences and locators for several applications

Some examples of trees in different languages

The 45 most frequent compound and person nouns in a Latin article

The 30 most frequent words in a Serbian article

Conclusions

Plug-in of UniteX into TreeCloud provides:

- A more accurate representation of forbidden words
- All kinds of multilinguals to be recognized in the text and presented in the tree
- A visual representation of some grammatical or semantic categories of the words
- A faster construction of the tree (via a careful use of the UniteX API)


Take advantage of the work already done by UniteX

Unitex/GrampLab analysis steps

program called
created files

Normalization

Tokenize
tokens, lexems, text
...

Dico
Locate
Concord
Annotated text

Consequences

To get the view root, we retrieve the text with matched sequences of the concord file in the new tokens of the text. New text and concord files are created. This process prevents double reading of the text and double division into words.

Thanks to the UniteX API and virtual file system, all this work is done in memory.