

# 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.science/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 Eclavit



- jaccard:  $1 O_{11} / (O_{11} + O_{12} + O_{21})$
- liddell: 1 (O<sub>11</sub>O<sub>22</sub> O<sub>12</sub>O<sub>21</sub>) / (C<sub>1</sub>C<sub>2</sub>)
- dice:  $1 2O_{11} / (R_1 + C_1)$
- hyperlex: 1 max(O<sub>11</sub> / R<sub>1</sub>, O<sub>11</sub> / C<sub>1</sub>)
- poissonstirling:  $O_{11}(\log O_{11} \log E_{11} 1)$
- chisquared: 1000  $N(O_{11} E_{11})^2 / (E_{11}E_{22})$
- zscore: 1  $(O_{11} E_{11}) / sqr(E_{11})$
- **ms**: 1 min( $O_{11} / R_1 O_{11} / C_1$ )
- oddsratio: 1 log((O<sub>11</sub>O<sub>22</sub>) / (O<sub>12</sub>O<sub>21</sub>))
- loglikelihood: 1 2( $O_{11} \log(O_{11}/E_{11}) + O_{12} \log(O_{12}/E_{12}) + O_{21} \log(O_{21}/E_{21}) + O_{22} \log(O_{22}/E_{22})$ )
- gmean:  $1 O_{11}/sqr(R_1C_1) = 1 O_{11}/sqr(NE_{11})$
- **mi** (mutual information): 1 log(O11/E<sub>11</sub>)
- ngd (normalized Google distance):  $(\max(\log R_1, \log C_1) \log O_{11}) / (N \min(\log R_1, \log C_1))$

#### Several versions of TreeCloud

#### Downloadable version in Python

• 2009: TreeCloud 1.3 for Windows, Linux, Mac developed by Philippe Gambette Use of SplitsTree 4.10 to draw the tree

#### **On-line version in C**

- 2009: 1<sup>st</sup> C version developed by Jean-Charles Bontemps
- 2012: Transition to Unicode developed by Claude Martineau
- 2014: 1<sup>st</sup> implementation of Unitex developed by Claude Martineau
- Use the API C and JAVA (JNI) that provides access to

Text dictionary

Several ways to use Unitex/GramLab

Two interfaces written in JAVA:

Unitex Core written in C/C++

GramLab IDE (project-oriented)

• Unitex IDE (classic)

They refer to

- a virtual file system · a persistence layer for resources (alphabets, dictionaries and corpora)
- Command lines or system calls with Perl, Python, etc.
- Unitex Core

by one who has <<u>Pattern>unwittingly fallen</Pattern></u> under your disple

.nce John, "his <<u>Pattern>heart is sinking</Pattern></u>; I am jealous lest

irchers, having <<u>Pattern>previously determined</Pattern></u> by lot their o

Notice that Unitex/GramLab can also produce an annotated text that includes all the tagged patterns.

Unitex IDE

Sample of concordances

#### GramLab IDE

- Conco

To get the *«new text»*, we **retokenize** the text with **matched sequences** of the *concord.ind* file as the new tokens of the text. New token.txt and text.cod 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.



#### LABORATOIRE D'INFORMATIQUE **GASPARD-MONGE**

Sous la co-tutelle de CNRS ÉCOLE DES PONTS PARISTECH ESIEE PARIS **UPEM • UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE** 

# Some examples of trees in different languages

# γκρουπ εξαίρεσ ομάδες Ηπειρο κλήρωση Νικίτα\_Σιμονιά <sup>θρύλος</sup> FIFA Μαρία Κομαντάνια Tree of the 50 most frequent words in a Greek article



Tree of the 20 most frequent compound and person nouns in an English article







Tree of the 30 most frequent words in a Serbian article

# Conclusion

Plug-in of Unitex into TreeCloud provides:

- A more accurate representation of forbidden words
- All kinds of multiwords 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)

http://treecloud.univ-mlv.fr

# How and Why to plug Unitex into TreeCloud?

### Construction of the tree with Unitex

- 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 stop list)

#### Get a larger and more accurate coverage of forbidden words



• The greyed box represents a call to a subgraph that recognizes *auxiliary* and *modal verbs*. • The second box uses some lexical masks (<ADV>, <CONJ>, ...) to match grammatical words. • The <ANTIPERSO> lexical mask in the last box matches a list of stopwords given by the user.

The adjacent local grammar contains three paths/boxes:

Each word of this list is labelled/tagged with ANTIPERSO. <**NB**> matches numbers.

#### Insert multiwords into the tree



If a dictionary contains compound words, these words can be kept in the tree but all multiwords cannot be listed in a dictionary.

The adjacent grammar contains a path that recognizes person names. The variable \$P\$ contains the name of a person captured by the subgraph Name of Person.

The .{Pers} label is added to the output (2017 online version)

#### Make a strong selection of the words kept in the tree



Since the number of words kept in the tree is very low (50 up to 150) and depends on their frequency, if a special category should be displayed, their selection should be made carefully. The adjacent grammar keeps only compound nouns.



In order to display verbs into the tree, it could be useful to get their LEMMA (since they have many inflected forms). The adjacent grammar is designed for this goal. A similar process is used for languages XXX with nouns or adjectives with cases (e.g. Serbian or Greek).

# 2017 on-line version of TreeCloud: an improved implementation

#### Introduction of the concept of file processing

In the online version of 2014, there is only a single Unitex processing for each language. The necessary resources were hard-coded into the program.

In the new 2017 version, there can be several processing procedures for each language. Furthermore, the Serbian language (Latin and Cyrillic) has been added. In order to manage these pairs (language, processing) a concept of processing file has been set up.

For example, in the processing file below, the first line indicates the path for French resources, then three dictionaries (French general dictionary, first name dictionary, toponym dictionary) have to be applied to the text. In the end, the local grammar is applied in the replace mode.

REP:TreeCloud_WS/French/src NB DICOS=3				
	FICHIER=Dela/dela-fr-public.bin			
	FICHIER=Dela/prenom-s.bin			
	FICHIER=Dela/Prolex-Unitex_1_2_TOPONYMES.bin			
GRAMMAIRE:				
	FICHIER=Graphs/Treecloud_N_Pers_Top_v1_FR.fst2 MODE=REPLACE			

# Take advantage of the work already done by Unitex

# Unitex/GramLab analysis steps

program called		created files	At the end of the Unitex analysis process, text.snt contains a cleaned text	
1	Normalize	text.snt	(normalization of separator characters),	
	Tokenize	tokens.txt, text.cod	<i>text.cod</i> contains the list of indexes of the tokens into the <i>tokens.txt</i> file list.	
	Dico	dlf, dlc, err	<i>dlf, dlc, err,</i> respectively contain simple words, compound words, unknown words <b>concord.ind</b> contains the matched	
	Locate	concord.ind		
ŧ	Concord		sequences with their position into the text (XXX, and multiword units)	
rdances Annotated text				