

Some automatic analyses of morphological structures

Nicolas Ballier USPC Paris Diderot CLILLAC-ARP (EA3967)

16/8/2017

OUTLINE AND ROADMAP

- ▶ 1: Linguistica (Python 3.5): the tool, its history and some uses
- ▶ 2: some uses of the program (data modelling and the learnability paradigm)
- ▶ 3. Morfessor (Python 2.7)
- ▶ 4. CHIPMUNK (Java)

SOME NOTIONS IN THE FIELD

- ▶ supervised versus unsupervised
- ▶ unsupervised morphology “discovery and morpheme segmentation”
- ▶ “learning algorithm”
- ▶ learnability
- ▶ Naive discriminant learning (Baayen 2011)

SECTION 1 : Linguistica 5 (python3)

<https://github.com/linguistica-uchicago/lxa5>

The screenshot shows a web browser window with the URL <http://people.cs.uchicago.edu/~jagoldsm/linguistica-site/> in the address bar. The page content is as follows:

The Linguistica Project

Introduction

In the years since 2003, several groups of students have worked with me on developing versions of Linguistica code. The central work behind Linguistica is a set of algorithms for determining the morphology of a natural language with no prior knowledge of the language. As our work has developed, there have been other functionalities that were natural to include in the package.

Linguistica 4 is a project written in Qt 3, developed from 2004 to 2010. There's a nice photo below of the group in the spring of 2006, including Yu Hu, Colin Sprague, and Aris Xanthos. Between 2007 and 2010, a number of improvements were made by Jonathan Nieder, Sravana Reddy, and Sonja Waxmonsky.

Linguistica 5 is a project written in Python 3 (well, some in Python 2.7 too) which is quite different from Linguistica 4 in its approach to the problem of morphology discovery. Jackson Lee and Anton Osten contributed greatly to this project, and Jackson has created an excellent GitHub site devoted to it: click [HERE to go to the GitHub page](#).

The Linguistica group at the University of Chicago draws its membership from the [Department of Linguistics](#) and the [Department of Computer Science](#). Our core interest is unsupervised learning of natural language structure, but this interest has taken us to work in a number of other areas, including automatically obtaining corpora through the Internet, and the discovery of structure in bioinformatic databases.

This site contains a good number of details about the Linguistica project and its supporting projects at the University of Chicago. By using the navigation menu on the left, you can learn more about the Linguistica project, download the latest version of the program and source code, read related papers about the theory involved, meet the group members, and navigate to other Natural Language Processing resources on the Web.

Enjoy!

Navigation Menu (Left Side):

- [Homepage](#)
- [Linguistica](#)
- [Documentation](#)
- [Setting up a development system](#)
- [Alchemist](#)
- [About Us](#)
- [Publications](#)
- [Downloads](#)
- [Dynamic computational networks](#)
- [Some data](#)

DataViz and unsupervised learning: Lee & Goldsmith 2016

- ▶ unsupervised word category induction (Christodoulopoulos et al. 2010)
- ▶ tools for data visualization
- ▶ model syntactic neighborhood
- ▶ reconstructing paradigms
- ▶ Signatures with the most stems in the Brown corpus
- ▶ Automatic outputs from character data

LEARNABILITY OF THE STRUCTURES

- ▶ morphological stems
- ▶ syntactic neighbours
- ▶ homophones
- ▶ “signatures”

Automatic output from the Brown corpus (Linguistica 2016)

affixes_to_signatures.txt
biphones.txt
phones.txt
predecessors.txt
signatures_to_stems.txt
signatures_to_stems_truncated.txt
signatures_to_words.txt
signatures_to_words_truncated.txt
stems_to_signatures.txt
stems_to_words.txt
successors.txt
triphones.txt
wordlist.txt, wordlist_by_avg_bigram_plog.txt ,
wordlist_by_avg_unigram_plog.txt
words_as_tries.txt
words_to_signatures.txt
words_to_sigtransforms.txt

DATA CLUSTERING : LEXICAL-FUNCTIONAL NETWORKS

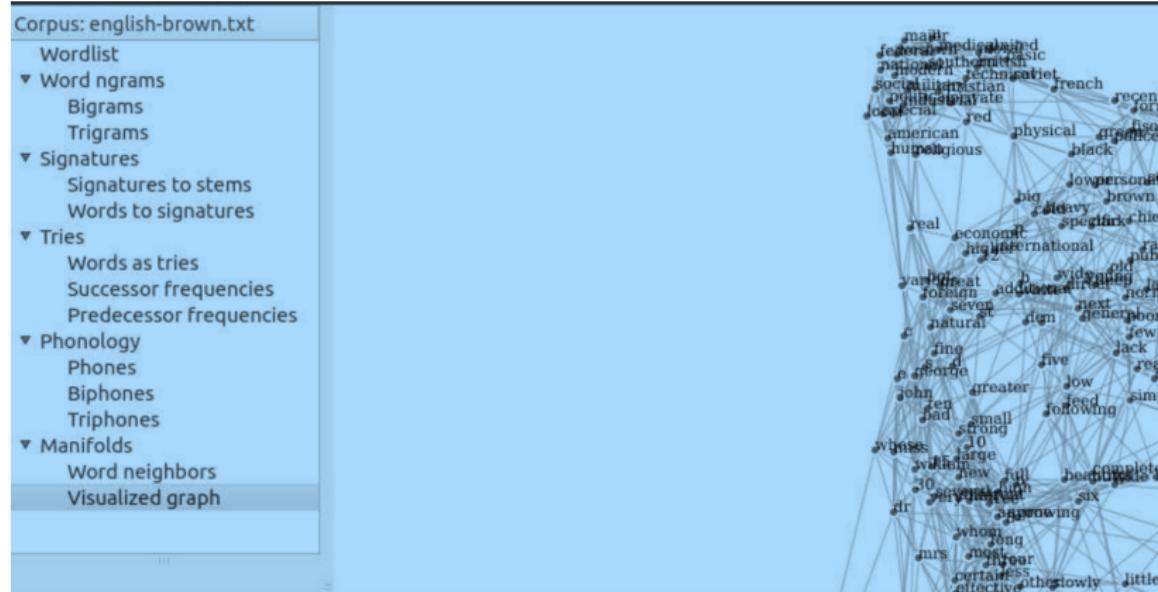


Figure 1:

LINGUISTICA : DETECTING MORPHOLOGICAL STRUCTURES (signatures)

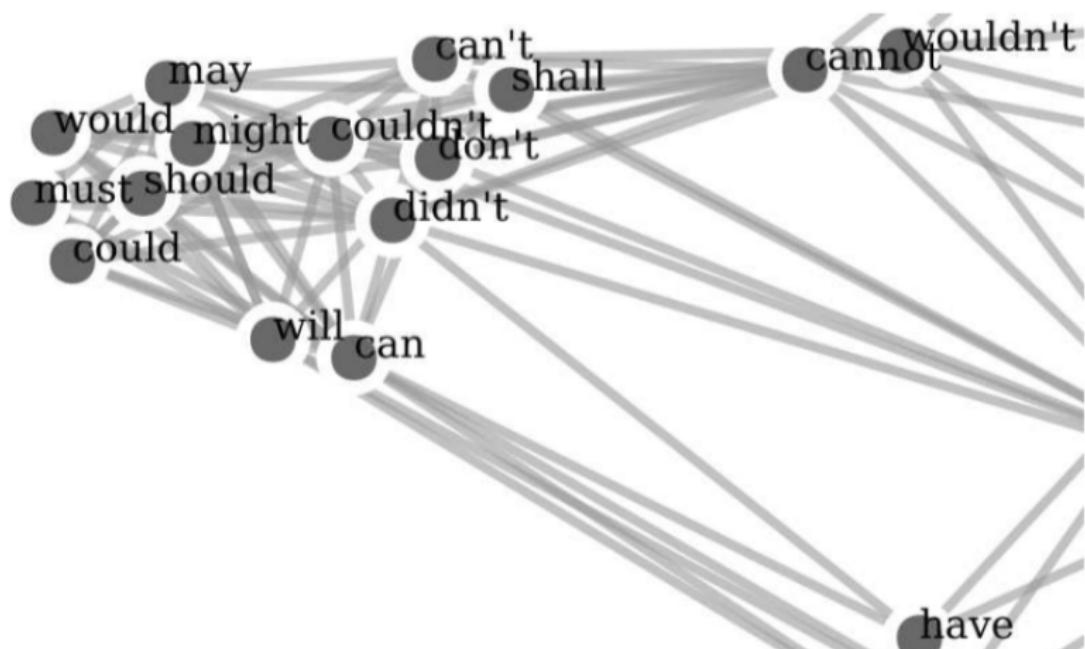
| | Signature | Stem count |
|----|---------------|------------|
| 1 | NULL/s | 2327 |
| 2 | 's/NULL | 813 |
| 3 | NULL/ly | 587 |
| 4 | NULL/d/s | 346 |
| 5 | NULL/d | 314 |
| 6 | ed/ing | 197 |
| 7 | '/NULL | 190 |
| 8 | 's/NULL/s | 181 |
| 9 | d/s | 175 |
| 10 | ies/y | 173 |
| 11 | NULL/ed/ing/s | 151 |
| 12 | NULL/ed | 134 |

NULL/ed/ing/s (number of stems: 151)

| | | | |
|----------|------------|----------|-----|
| abound | administer | affirm | aff |
| appeal | arrest | assault | att |
| awaken | award | beckon | be' |
| bloom | bolt | broaden | bu |
| claw | click | climb | clu |
| coil | compound | concern | cor |
| confront | contact | contrast | cra |
| crown | decay | deck | dia |
| display | drill | drown | du |
| eschew | escort | exceed | exc |
| extend | filter | flounder | fro |
| haunt | hoot | hover | ho |

LINGUISTICA : DETECTING SYNTACTIC STRUCTURES

NETWORK OF MODAL NEIGHBOURS IN THE BROWN CORPUS (Lee & Goldsmith 2016)



LINGUISTICA : DETECTING STRUCTURES

SYNTACTIC NEIGHBOURS IN THE BROWN CORPUS (Lee & Goldsmith 2016)

| Word | Syntactic neighbors |
|-------|--|
| the | a his their an its this my our that |
| would | could must will can should may might |
| after | before like during since without through |

Table 1: Syntactic neighbors

Figure 4:

FIRST STEPS WITH LINGUISTICA

<http://lingistica-uchicago.github.io/lxa5/download.html>

for the command line interface

Library/Python/3.5/anaconda/lib/python3.5/site-packages/lingistica

go to /Library/Python/3.5/anaconda/lib/python3.5/site-packages/lingistica

python -m linguistica cli *** #####follow the steps described here:
<http://lingistica-uchicago.github.io/lxa5/cli.html>

LINGUISTICA (ref): <http://www.aclweb.org/anthology/N16-3005>
John A. Goldsmith. 2001. Unsupervised learning of the morphology
of a natural language. Computational Linguistics, 27(2):153–198.

John A. Goldsmith. 2006. An algorithm for the unsupervised

Morfessor

training sets : corpora / lexical inventories

Mathias Creutz & Krista Lagus (Helsinki) Creutz, M., & Lagus, K. (2005). Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Helsinki University of Technology.

Sami Virpioja Peter Smit Stig-Arne Grönroos Mikko Kurimo Aalto University.

The output can be directed to a file using the following syntax:

```
morfessor1.0.pl -data inputfilename > outputfilename
```

If additionally some optional parameters are defined, Morfessor can be invoked, e.g., like this:

```
morfessor1.0.pl -savememory  
-gammalendistr 10 -trace 19 -data inputfilename > outputfilename
```

"Morfessor is a family of probabilistic machine learning methods that find morphological segmentations for words of a natural language, based solely on raw text data." (Morfessor report)

MORFESSOR: FIRST STEPS

Install setuptools for python curl

```
https://bootstrap.pypa.io/ez_setup.py -o - | python ez_setup.py
```

```
####copy morfessor.master.zip from github:
```

```
####https://github.com/aalto-speech/morfessor ####read  
report + doc from
```

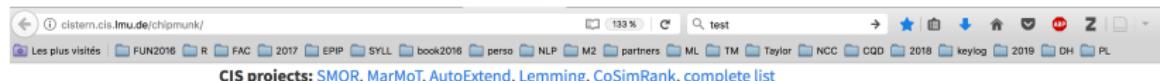
```
http://morfessor.readthedocs.io/en/latest/cmdtools.html ####run  
the script morfessor-train –encoding=ISO_8859-15 –traindata-list  
–logfile=log.log -s model.bin -d ones traindata.txt
```

EXAMPLE of OUTPUT

```
INFO:morfessor.io:Reading corpus from list 'traindata.txt'...
INFO:morfessor.io:Done. INFO:morfessor.baseline:Compounds in
training data: 34148 types / 34148 tokens
INFO:morfessor.baseline:Starting batch training
INFO:morfessor.baseline:Epochs: 0 Cost: 1047578.61975
..... INFO:morfessor.baseline:Epochs: 1 Cost:
790234.810818 .....INFO:
morfessor.baseline:Epochs: 2 Cost: 747577.177172
.....INFO:morfessor.baseline:
Epochs: 3 Cost: 743192.072777
.....INFO:morfessor.baseline:
Epochs: 4 Cost: 742134.191572
.....
INFO:morfess ..... INFO:morfessor.baseline:Epochs: 6
Cost: 741768.998685 INFO:morfessor.baseline:Done. Epochs: 6
Final cost: 741768.998685 Training time: 206.947s
INFO:morfessor.io:Saving model to 'model.bin'...
INFO:morfessor.io:Done. openroam-prg-og-1-158-87:Morfessor
```

CHIPMUNK (Java-based)

SUPERVISED LEARNING trained on English data



The screenshot shows a web browser window with the URL cistern.cis.lmu.de/chipmunk/. The page title is "ChipMunk - A morphological segmenter and stemmer". Below the title, there is a banner with the text "CIS projects: SMOR, MarMoT, AutoExtend, Lemming, CoSimRank, complete list". The browser's address bar and various tabs are visible at the top.

ChipMunk - A morphological segmenter and stemmer



(Source: wikipedia.org)

ChipMunk is a tool for labeled segmentation, morphological analysis and stemming. The implementation found here is not the one used in the [paper](#), but a complete rewrite. On this page you can find links to the source code, binaries, pretrained models, datasets and more.

Usage

The [chipmunk_example.sh](#) script shows how ChipMunk can be trained and run. It is completely self-contained and will download all the needed JARs and datasets.

Figure 5:

CHIPMUNK : FIRST STEPS

sh chipmunk_example.sh ##### runs the input.txt file

Creutz, M., & Lagus, K. (2005). Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Helsinki University of Technology.

<http://nlp.cs.swarthmore.edu/~richardw/papers/creutz2005-unsupervised.pdf>

CHIPMUNK: sample outputs

wonderful wonder:SEGMENT ful:SEGMENT
wonderland wonder:SEGMENT land:SEGMENT
wonderland wonder:SEGMENT land:SEGMENT
wonderland wonder:SEGMENT land:SEGMENT
wonderment wonder:SEGMENT ment:SEGMENT
wonderment wonder:SEGMENT ment:SEGMENT
wonderment wonder:SEGMENT ment:SEGMENT
wondrous wondrous:SEGMENT
wondrous wondrous:SEGMENT
wondrous wondrous:SEGMENT
wonga-wonga wonga:SEGMENT -:SPECIAL wonga:SEGMENT
wonga-wonga wonga:SEGMENT -:SPECIAL wonga:SEGMENT
wonkywonky:SEGMENT
wonkywonky:SEGMENT
wonted wont:SEGMENT ed:SEGMENT
wonted wont:SEGMENT ed:SEGMENT
woodbine woodbine:SEGMENT
woodbine woodbine:SEGMENT
woodblock wood:SEGMENT block:SEGMENT
woodblock wood:SEGMENT block:SEGMENT
woodblock wood:SEGMENT block:SEGMENT
woodchuck wood:SEGMENT chuck:SEGMENT

Figure 6:

SAMPLE OUTPUT (Slovak)

big bytoví hudobníci big:SEGMENT :SPECIAL bytoví:SEGMENT
:SPECIAL hudobníci:SEGMENT čižmáky č:SEGMENT
ižmáky:SEGMENT :SPECIAL Dekapitulácia
Dekapitulácia:SEGMENT demokratúra demokratúra:SEGMENT
dePrimátor dePrimát:SEGMENT or:SEGMENT diskutéka
diskutéka:SEGMENT

REFERENCES

Main papers describing the programs plus

Baayen, R. H. (2011). Corpus linguistics and naive discriminative learning. *Revista Brasileira de Linguística Aplicada*, 11(2), 295-328.
[http://www.sfs.uni-tuebingen.de/~hbaayen/publications/
BaayenBJAL2011.pdf](http://www.sfs.uni-tuebingen.de/~hbaayen/publications/BaayenBJAL2011.pdf)

Goldsmith, J. A. (2010). Segmentation and morphology. *The Handbook of Computational Linguistics and Natural Language Processing*, Blackwell Wiley, 364-393. Goldsmith, J. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198. Goldsmith. J. 2006. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12(4):353–371) Goldsmith et al. 2017 *Linguistica*
<https://linguistica-uchicago.github.io/lxa5/demo.html>

CONCLUSION

THANK YOU !

nballier@free.fr

