

*A wide-coverage free/open-source
deep parser for Brazilian
Portuguese: a work in progress*

Leonel Figueiredo de Alencar

<http://www.leonel.profusehost.net/>

Research Group on Language and Computation
(**CompLin**)

Postgraduate Program in Linguistics
Federal University of Ceará (UFC)

Our goals

- Long-term goal: development of a free/open-source robust parser for unrestricted text in Brazilian Portuguese (henceforth BP), integrating symbolic and statistical NLP techniques
 - Short-term goal: development of shallow processing tools for BP, adapting/integrating available resources as much as possible
-
-

AeliusDonatus

- Aelius: tools for shallow processing of BP
 - Already implemented: tokenizers and taggers
 - Under development: chunker

<http://aelius.sourceforge.net/>

- Donatus: tools for deep processing of BP
 - Implemented: ALEXP, a tagger-parser interface (ALENCAR, 2011)
 - Under development: deep grammar for BP nominal expressions in a constraint-based formalism
-
-

Who was Aelius Donatus?

Roman grammarian (mid 4th century), author of *Ars Grammatica*, which popularized the notion of parts of speech (*partes orationis*), such as Nouns, Verbs, Adjectives, etc (JUNGEN; LOHNSTEIN, 2006). However, in computational linguistics, parsing a sentence implies labeling the sentence constituents (typically word groups forming a unit), assigning the sentence a hierarchical structure.

Deep parsing of Portuguese: state of the art

Freely accessible parsers	claims robustness?	FOSS	disclosed source	freely downloadable	X-bar theory
CURUPIRA	yes	no	no	yes	no
Grammar Play	no	yes	yes	yes	yes
VISL	yes	no	no	no	no
LX-Parser	yes	no	yes	yes	yes
LX-Gram	yes	no	yes	yes	yes

Freely accessible deep parsers of Portuguese

- Curupira

<http://www.nilc.icmc.usp.br/nilc/tools/curupira.html>

- Grammar Play

<http://sites.google.com/site/gabrielothero/home/publicacoes>

- VISL

<http://beta.visl.sdu.dk/visl/pt/parsing/automatic/trees.php>

- LX-Parser

<http://lxcenter.di.fc.ul.pt/services/en/LXServicesParser.html>

- LX-Gram

<http://nlxgroup.di.fc.ul.pt/lxgram/>



Robustness

- A robust parser always delivers some useful (though somewhat degraded) output, even under unpredicted circumstances (LJUNGLÖF; WIRÉN, 2010, p. 79-80)



Free/open source software (FOSS)

- FOSS = software whose license complies with the licenses approved either by the Free Software Foundation (FSF) or the Open Source Initiative (OSI):

<http://www.gnu.org/licenses/license-list.html>

<http://www.opensource.org/licenses>



LX-Center's tools are not FOSS

- LX-Parser and LX-Gram (as well as other LX-Center's products like the LX-Tagger) are freely distributed (by anonymous download) in source form

<http://lxcenter.di.fc.ul.pt/tools/en/>

- However, at least two license conditions are not compliant with FSF/OSI principles:
-
-

FOSS incompatible conditions in the LX-Parser license

- “6. The user is not allowed to distribute or market any derivative product or service based on all or part of the parser.“
- “7. The user is not permitted to make available to the public all or part of the contents of the parser, by the distribution of copies, by renting, leasing or any other form of distribution, including free or open-source ones, web services, 'mash-up' or others.“

http://lxcenter.di.fc.ul.pt/tools/en/conteudo/LX-Parser_License.pdf

Some basic formal language theory

- For a given string language there is a infinite number of grammars generating this language
- Example: Drummond's language (from poem *Quadrilha*)

$L = \{ \text{„John loved Theresa“}, \text{„John loved Theresa who loved Raymond“}, \text{„John loved Theresa who loved Raymond who loved Mary“}, \dots \}$

A grammar representing an inelegant solution for parsing Drummond's language

$S \rightarrow N V N Sbar$

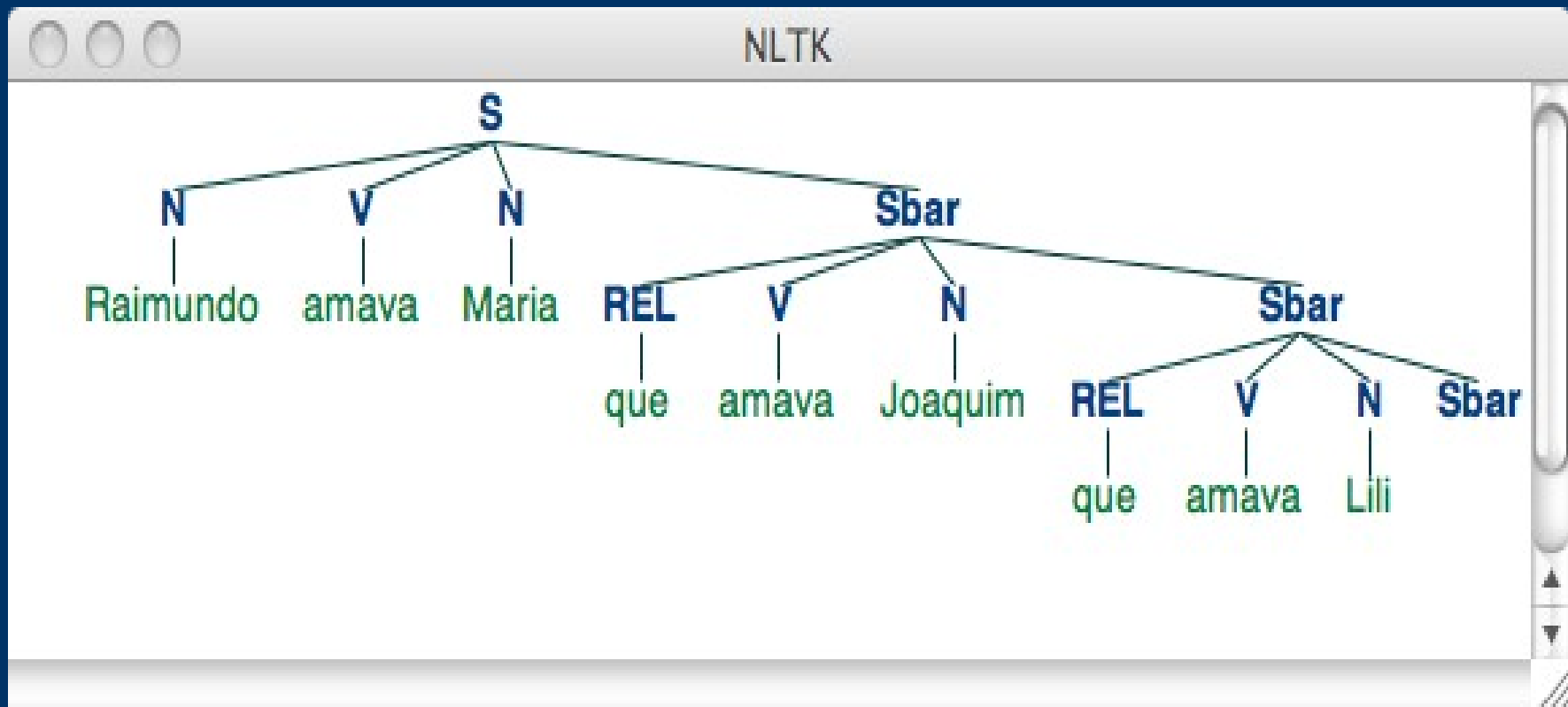
$Sbar \rightarrow REL V N Sbar \mid$

$REL \rightarrow 'que'$

$N \rightarrow 'João' \mid 'Teresa' \mid 'Joaquim' \mid 'Raimundo' \mid 'Lili' \mid$
 $'Maria' \mid 'se'$

$V \rightarrow 'suicidou' \mid 'amava'$

Parse tree generated by the inelegant grammar



A grammar representing a more elegant solution for parsing Drummond's language

S -> NP VP

VP -> V NP

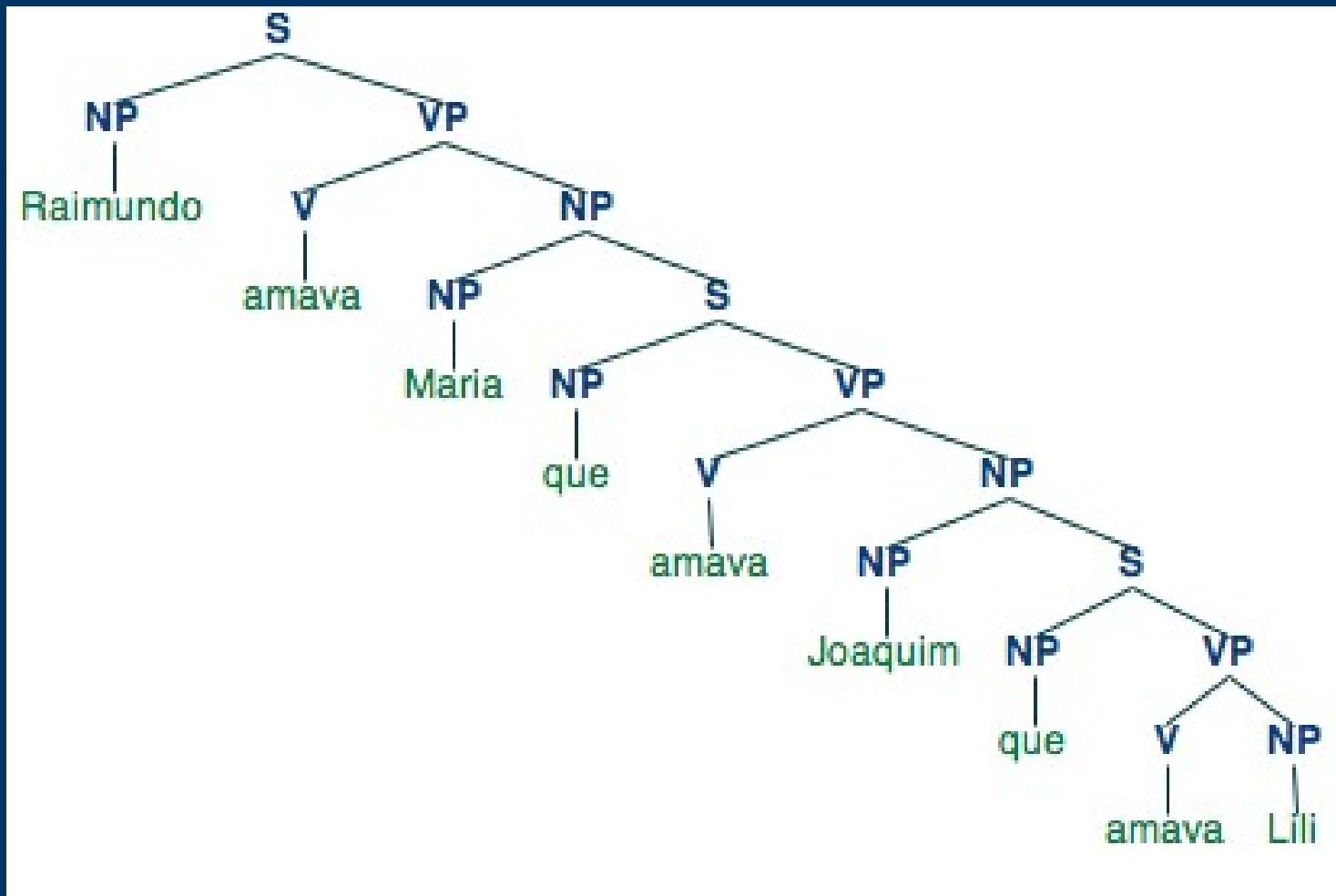
NP -> NP S

NP -> 'João' | 'Teresa' | 'Joaquim' | 'Raimundo' | 'Lili' |
'Maria' | 'que' | 'se'

V -> 'suicidou' | 'amava'



Parse tree #1 generated by the more elegant solution



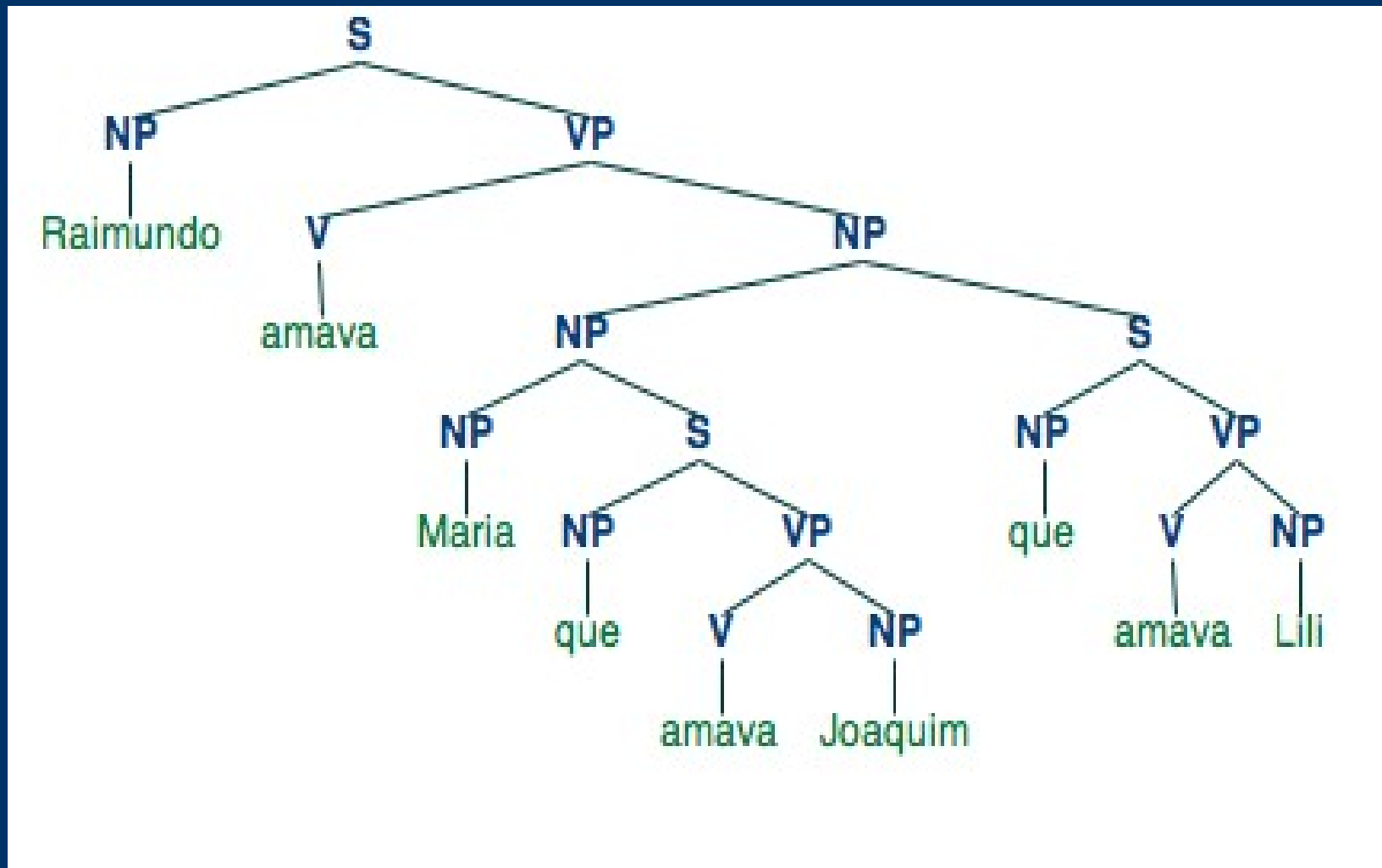
Traditional grammar revisited

- The latter tree can straightforwardly be interpreted in traditional grammar terms
 - The subject of the sentence is the NP daughter node of S
 - The predicate of the sentence is the VP daughter node of S
 - The object of the sentence is the NP daughter node of VP
 - The higher level relative clause is a modifier of its sister NP node
 - These syntactic-semantic relations are not so clearly reflected in the former tree (e.g. “Mary” and the relative clause do not form a constituent)
-
-

Linguistic elegance and its usefulness for natural language technology

- Example of a task in a natural language-based database querying (see Bird; Klein; Loper, 2009, p. 361-365):
 - Quem Raimundo amava?
“Whom did Raymond love?”
 - With linguistically well-motivated trees like the previous one, solving this task is quite straightforward: search for sentence trees with Raymond as subject and main verb *love* and then search for the verb's object
-
-

Pervasive ambiguity: separating the wheat from the chaff



X-bar theory: the X-bar schema

X and Y are variables ranging over lexical categories (N, V, A, etc.):

(i) $XP \rightarrow YP, X'$ (specifier rule)

(ii) $X' \rightarrow X', YP$ (adjunct rule)

(iii) $X' \rightarrow X, YP^*$ (complement rule)

(RADFORD, 1988, p. 277)

Some instantiations of X-bar theory rules

The left-hand sides of the rules (i) – (iii) are unordered sets.

Thus, rule (iii) derives both head-first and head-last orderings:

- weil Hans [_{VP} Teresa liebte] (German)
 - because John [_{VP} loved Teresa] (English)
 - Ich laufe [_{PP} den Fluss entlang] (German)
 - I walk [_{PP} along the river] (English)
-
-

X-bar theory

- X-bar theory is an effort by generative linguists to capture cross-categorial generalizations (e.g. complements are sisters of their governing heads)
 - It aims at constraining the search space for the task of formulating the grammar of a natural language
 - It is argued that, without such constraints, which are claimed to be hard-wired into the human brain, the language acquisition problem is unsolvable (a child would never attain the right hypothesis due to impoverished input)
-
-

X-bar theory

- This effort was initiated by Chomsky about 40 years ago
 - In the meantime, there are several (conflicting) versions of X-bar theory
 - Chomskyan generative linguists (i.e. working under the Minimalist Program) claim all phrases are endocentric; apparent exceptions are derived by movement
 - Non-transformational generative models like LFG admit of exocentric constructions (typically in non-configurational languages like Malayalam)
-
-

Is X-bar theory useful in NLP outside generative linguistics?

- X-bar theory results from applying software design principles (e.g. economy, elegance, and generality) to the task of formally describing a natural language or the general architecture of human language
- The linguist/programmer writing a symbolic parser for a natural language can be compared in some respects to a child acquiring its first language

Is X-bar theory useful in NLP outside generative linguistics?

- X-bar theory is definitely not the only constrained syntactic theory; it is perhaps the one with the oldest tradition and all formal linguists are familiar with
- For a large number of languages, there exists some X-bar theoretic description, so that a programmer writing a parser for a given language does not have to reinvent the wheel



Is X-bar theory useful in NLP outside generative linguistics?

- As far as Brazilian Portuguese is concerned, Othero (2009) is a good starting point for implementing a symbolic parser based on X-bar theory



Relevance of X-bar theory for natural language technologies

- Grammar development and maintenance
 - Rule-based machine translation (RBMT)
 - Corpus linguistics/text technology
 - Statistical NLP
-
-

Grammar Development and Maintenance

- Different people working within the same X-bar theoretic approach can more easily collaborate in a grammar engineering effort for one specific natural language



Rule-Based Machine Translation

- Grammars for different languages implemented in the same X-bar framework by different teams can more easily be integrated into a rule-based machine translation system



Treebanks and statistical NLP

- Treebanks are becoming increasingly more important resources for statistical NLP
- Statistical parsers trained on treebanks with more hierarchized structures perform better (in terms of F-Score) than with less hierarchized structures (MAIER, 2007)



X-bar theory as a tool for principled syntactic corpus annotation

- X-bar theory fulfills the following criterium:

„[...] one of the prerequisites for achieving a reliably annotated corpus is to base the annotation scenario on a well-defined linguistic theory“ (HAJIČOVÁ et al., 2010, p. 171)



Development environment for the first phases of the project

- The Natural Language Toolkit (NLTK):

<http://www.nltk.org>

- The most user-friendly and apparently the richest free/open-source NLP library
 - Implemented in Python with interfaces to libraries implemented in other languages
 - Python: ideal language for prototyping
-
-

NLTK's typical NLP Pipelines: some basic terminology

Pipeline 1

(1) sentence tokenization => (2) word tokenization
=> (3) tagging => (4) chunking => (5) NER =>
(6) relation extraction

Pipeline 2

(1) sentence tokenization => (2) word tokenization
=> (7) deep (complete) parsing => (8) natural
language understanding

(1) – (4): shallow processing

(4): shallow (partial) parsing

AeliusDonatus' Pipeline

(1) sentence tokenization => (2) word
tokenization => (3) tagging => (4) contraction
splitting => (5) chunking => (6) deep syntactic
parsing



Aelius: customized NLTK resources for shallow processing of Brazilian Portuguese

- In its current release version, Aelius is a suite of Python, NLTK-based modules and language data for tokenizing and POS-tagging Brazilian Portuguese texts (ALENCAR, 2010):
<http://aelius.sourceforge.net/>
 - It offers Python/NLTK interfaces to some external resources (e.g. MXPOST taggers, Stanford taggers and parsers, etc.)
 - The present development version also includes a chunking module
-
-

Aelius taggers: near state-of-the-art accuracy in some texts

Taggers	Architecture	Accuracy on sample LH	Accuracy on sample CT	Accuracy on sample DC
AeliusRUBT	Hybrid (NLTK regular expression/n-gram tagger)	95.29%	94.70%	92.10%
AeliusHunPos	HunPos (HMM based)	96.35%	95.83%	92.78%
AeliusBRUBT	Brill (NLTK transformation-based learning tagger)	95.30%	95.10%	91.83%

Samples used for evaluating taggers

- LH: the first 8 chapters (about 18k tokens) from the novel *Luzia-Homem* by Domingos Olímpio (1906)
 - CT: excerpts from different 19th century literary works (about 4k tokens)
 - CJ: 2 scientific texts in the fields of medical science and forestry (about 2k tokens)
-
-

Tagset used by Aelius taggers

- Aelius taggers were trained on the Tycho Brahe Parsed Corpus of Historical Portuguese
<http://www.tycho.iel.unicamp.br/~tycho/corpus/en/>
 - This corpus uses a very informative tagset with 376 tags labeling not only parts of speech, but also inflectional features
 - Due to data sparsity, it is expected that tagger performance decreases with the increase of the tagset, if the training material remains constant
-
-

Aelius interfaces to external, non-NLTK taggers

- HunPos open-source POS-Tagger (HALÁCSY; KORNAI; ORAVECZ, 2007) (NLTK's hunpos module)
 - Language model for BP: AeliusHunPos
- Stanford POS-Tagger (NLTK's stanford module)
 - No language model for BP available

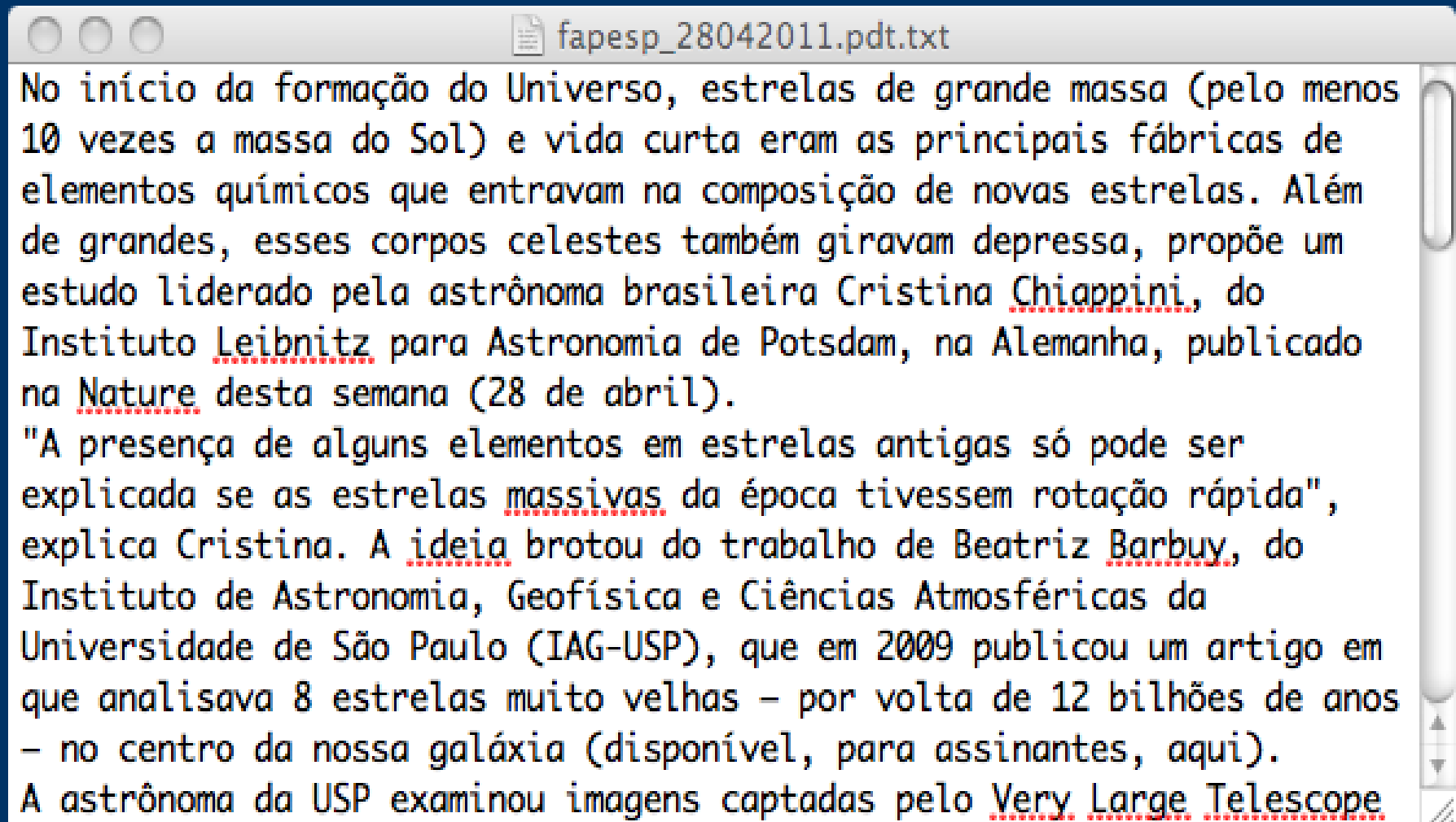


Aelius interfaces to external, non-NLTK taggers

- MXPOST (RATNAPARKHI, 1996)
 - Language model for European Portuguese: LX-Tagger (BRANCO; SILVA, 2004) with reported 96.24% accuracy:

<http://lxcenter.di.fc.ul.pt/tools/en/conteudo/LXTagger.html>

Example input to Aelius POS-Tagging module



fapesp_28042011.pdt.txt

No início da formação do Universo, estrelas de grande massa (pelo menos 10 vezes a massa do Sol) e vida curta eram as principais fábricas de elementos químicos que entravam na composição de novas estrelas. Além de grandes, esses corpos celestes também giravam depressa, propõe um estudo liderado pela astrônoma brasileira Cristina Chiappini, do Instituto Leibnitz para Astronomia de Potsdam, na Alemanha, publicado na Nature desta semana (28 de abril).

"A presença de alguns elementos em estrelas antigas só pode ser explicada se as estrelas massivas da época tivessem rotação rápida", explica Cristina. A ideia brotou do trabalho de Beatriz Barbuy, do Instituto de Astronomia, Geofísica e Ciências Atmosféricas da Universidade de São Paulo (IAG-USP), que em 2009 publicou um artigo em que analisava 8 estrelas muito velhas – por volta de 12 bilhões de anos – no centro da nossa galáxia (disponível, para assinantes, aqui).

A astrônoma da USP examinou imagens captadas pelo Very Large Telescope

Original text

<http://revistapesquisa.fapesp.br/?art=71494&bd=2&pg=1&lg=>

NOTÍCIAS

Volta aos primórdios do Universo

Pesquisa indica que estrelas responsáveis por produzir elementos há 12 bilhões de anos tinham rotação rápida

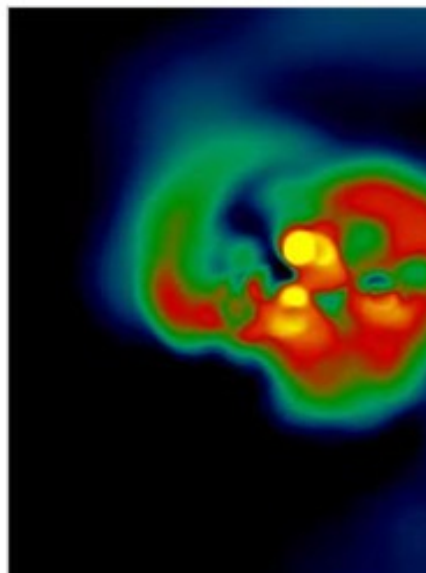
Maria Guimarães

Edição Online - 27/04/2011

 Tweet < 0

No início da formação do Universo, estrelas de grande massa (pelo menos 10 vezes a massa do Sol) e vida curta eram as principais fábricas de elementos químicos que entravam na composição de novas estrelas. Além de grandes, esses corpos celestes também giravam depressa, propõe um estudo liderado pela astrônoma brasileira Cristina Chiappini, do Instituto Leibnitz para Astronomia de Potsdam, na Alemanha, [publicado na Nature desta semana](#) (28 de abril).

© ATHENA STACY/UNIVERSITY OF TEXAS

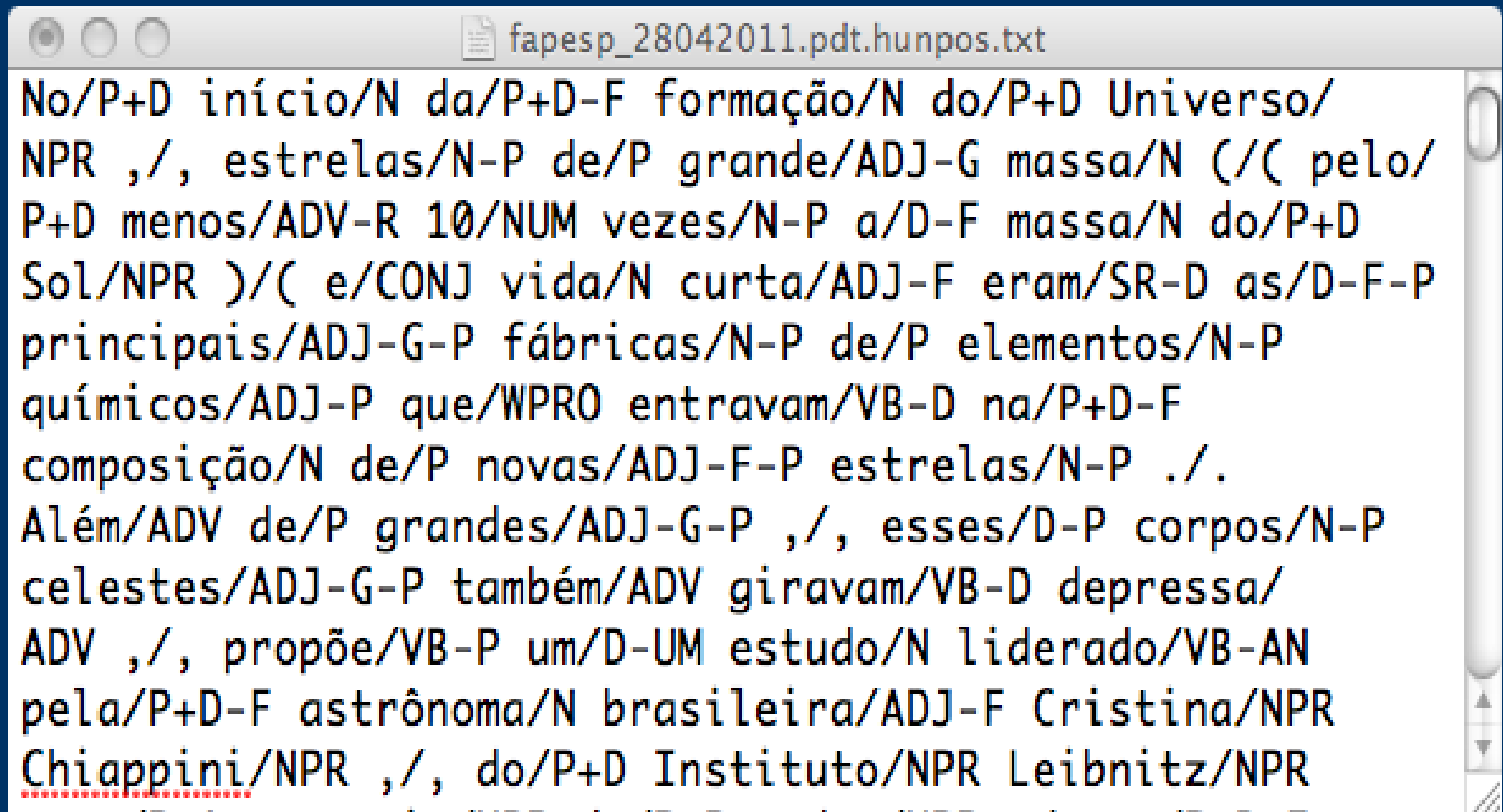


Example xml output from AeliusHunPos

```
<div type="chap" n="1"><p n="1"><s n="1"><w xml:id="w1" type="P+D">No</w> <w xml:id="w2" type="N">início</w> <w xml:id="w3" type="P+D-F">da</w> <w xml:id="w4" type="N">formação</w> <w xml:id="w5" type="P+D">do</w> <w xml:id="w6" type="NPR">Universo</w> <w xml:id="w7" type=",">,</w> <w xml:id="w8" type="N-P">estrelas</w> <w xml:id="w9" type="P">de</w> <w xml:id="w10" type="ADJ-G">grande</w> <w xml:id="w11" type="N">massa</w> <w xml:id="w12" type="(">(</w> <w xml:id="w13" type="P+D">pelos</w> <w xml:id="w14" type="ADV-R">menos</w> <w xml:id="w15" type="NUM">10</w> <w xml:id="w16" type="N-P">vezes</w> <w xml:id="w17" type="D-F">a</w> <w xml:id="w18" type="N">massa</w> <w xml:id="w19" type="P+D">do</w> <w xml:id="w20" type="NPR">Sol</w> <w xml:id="w21" type="(">)</w> <w xml:id="w22" type="CONJ">e</w> <w xml:id="w23" typ
```

U: --- fapesp_28042011.pdt.hunpos.xml Top (1,0) (nXML Valid)

Example simple text output from AeliusHunPos



```
fapesp_28042011.pdt.hunpos.txt
No/P+D início/N da/P+D-F formação/N do/P+D Universo/
NPR ,/, estrelas/N-P de/P grande/ADJ-G massa/N (/C pelo/
P+D menos/ADV-R 10/NUM vezes/N-P a/D-F massa/N do/P+D
Sol/NPR )/( e/CONJ vida/N curta/ADJ-F eram/SR-D as/D-F-P
principais/ADJ-G-P fábricas/N-P de/P elementos/N-P
químicos/ADJ-P que/WPRO entravam/VB-D na/P+D-F
composição/N de/P novas/ADJ-F-P estrelas/N-P ./..
Além/ADV de/P grandes/ADJ-G-P ,/, esses/D-P corpos/N-P
celestes/ADJ-G-P também/ADV giravam/VB-D depressa/
ADV ,/, propõe/VB-P um/D-UM estudo/N liderado/VB-AN
pela/P+D-F astrônoma/N brasileira/ADJ-F Cristina/NPR
Chiappini/NPR ,/, do/P+D Instituto/NPR Leibnitz/NPR
```

Tokenizing, tagging, and chunking a sentence with Aelius

**Cientistas mostram que reservas florestais
comunitárias podem retardar a fragmentação da
floresta amazônica**

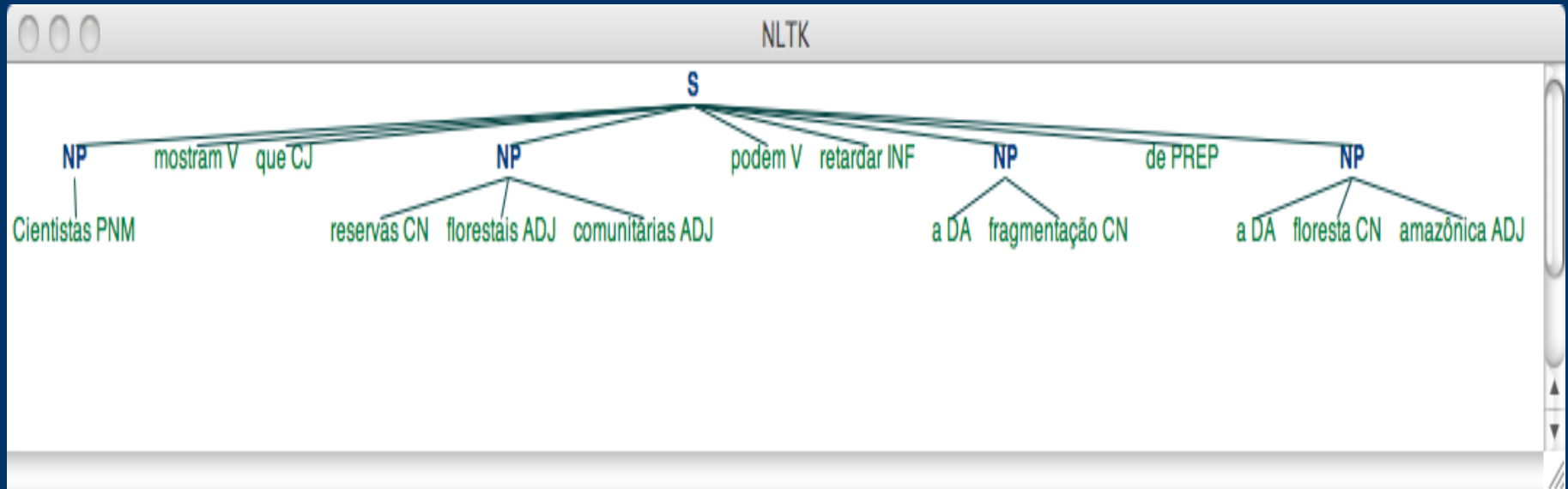
Source: <http://lba.cptec.inpe.br/lba/site/?p=reservasf&t=0>



Tokenizing, tagging, and chunking a sentence with Aelius

```
>>> from Aelius import Chunking,AnotaCorpus
>>> from Aelius import carrega as load
>>> lxtagger=load("lxtagger")
>>> s="cientistas mostram que reservas florestais comunitárias podem retardar a fragmentação da
floresta amazônica".decode("utf-8")
>>> t=AnotaCorpus.TokPort.tokenize(s)
>>> t
[u'cientistas', u'mostram', u'que', u'reservas', u'florestais', u'comunit\xe1rias', u'podem', u'retardar',
u'a', u'fragmenta\xe7\xe3o', u'da', u'floresta', u'amaz\xf4nica']
>>> t=AnotaCorpus.toquenizaContracoes([t])
>>> tagged_tokens=AnotaCorpus.anota_sentencas(t,lxtagger,"mxpost")
>>> tagged_tokens
[[('Cientistas', 'PNM'), ('mostram', 'V'), ('que', 'CJ'), ('reservas', 'CN'), ('florestais', 'ADJ'), ('comunit
\xe1rias', 'ADJ'), ('podem', 'V'), ('retardar', 'INF'), ('a', 'DA'), ('fragmenta\xe7\xe3o', '
CN'), ('de', 'PREP'), ('a', 'DA'), ('floresta', 'CN'), ('amaz\xf4nica', 'ADJ')]]
>>> shallow_tree=Chunking.CHUNKER.parse(tagged_tokens[0])
>>> shallow_tree
Tree('S', [Tree('NP', [('Cientistas', 'PNM')]), ('mostram', 'V'), ('que', 'CJ'), Tree('NP', [('reservas', 'CN
'), ('florestais', 'ADJ'), ('comunit\xe1rias', 'ADJ')]), ('podem', 'V'), ('retardar', 'INF'), Tree('NP', [
('a', 'DA'), ('fragmenta\xe7\xe3o', 'CN')]), ('de', 'PREP'), Tree('NP', [('a', 'DA'), ('floresta',
'CN'), ('amaz\xf4nica', 'ADJ')])])])])])])
>>>
```

Chunk tree



Proper names correctly chunked by LX-NER

<http://lxcenter.di.fc.ul.pt/services/en/LXServicesNer.html>

Além de grandes, esses corpos celestes também giravam depressa, propõe um estudo liderado pela astrônoma brasileira **Cristina Chiappini**, do **Instituto Leibnitz para Astronomia de Potsdam**, na **Alemanha**, publicado na **Nature** desta semana (28 de abril).

A less trivial example

O INPE – órgão vinculado ao Ministério da Ciência e Tecnologia – é responsável pelo monitoramento e estudo do território brasileiro por satélite, pela coleta de dados científicos sobre camadas atmosféricas e a formação de pesquisadores na área.

Source: http://www.ufc.br/portal/index.php?option=com_content&task=view&id=11284&Itemid=1

Failure of LX-NER in correctly chunking more complex proper names

<http://lxcenter.di.fc.ul.pt/services/en/LXServicesNer.html>

O **INPE** – órgão vinculado ao **Ministério** da Ciência e **Tecnologia** – é responsável pelo monitoramento e estudo do território brasileiro por satélite, pela coleta de dados científicos sobre camadas atmosféricas e a formação de pesquisadores na área.

Comparison with Aelius chunking module using LX-Tagger



Tagging a sentence with AeliusBRUBT

```
>>> s="Cientistas mostram que reservas florestais comunitárias podem retardar a fragmentação da floresta amazônica".decode("utf-8")
>>> t=AnotaCorpus.TokPort.tokenize(s)
>>> b=load("AeliusBRUBT.pkl")
>>> tagged_sent=AnotaCorpus.anota_sentencas([t],b,"nltk")
>>> for w,t in tagged_sent[0]:
    print "%s/%s " % (w,t),
```

```
Cientistas/N-P mostram/VB-P que/WD reservas/N-P florestais/N-P comunitárias/N-P podem/VB-P retardar/VB a/P fragmentação/N da/P+D-F floresta/N amazônica/ADJ-F
```

```
>>>
```

Tagging a sentence with AeliusHunPos

```
>>> h=load("AeliusHunPos")
>>> tagged_sent=AnotaCorpus.anota_sentencas([t],h,"hunpos")
>>> tagged_sent
[[('Cientistas', 'N-P'), ('mostram', 'VB-P'), ('que', 'C'), ('reservas', 'VB-P'), ('florestais', 'N-P'), ('comunit\xc3\xa1rias', 'ADJ-F-P'), ('podem', 'VB-P'), ('retardar', 'VB'), ('a', 'D-F'), ('fragmenta\xc3\xa7\xcc3\xa3o', 'N'), ('da', 'P+D-F'), ('floresta', 'N'), ('amaz\xcc3\xb4nica', 'ADJ-F)]]
>>> for w,t in tagged_sent[0]:
    print "%s/%s " % (w,t),

Cientistas/N-P mostram/VB-P que/C reservas/VB-P florestais/N-P comunitárias/ADJ-F-P podem/VB-P retardar/VB a/D-F fragmentação/N da/P+D-F floresta/N amazônica/ADJ-F
>>>
```

Evaluating tagger accuracy: AeliusBRUBT versus AeliusHunPos

- AeliusBRUBT: 4 errors in 13 tokens
 - Cientistas/N-P mostram/VB-P que/WD@C reservas/N-P florestais/N-P@ADJ-G-P comunitárias/N-P@ADJ-F-P podem/VB-P retardar/VB a/P@D-F fragmentação/N da/P+D-F floresta/N amazônica/ADJ-F
- AeliusHunPos: 2 errors in 13 tokens
 - Cientistas/N-P mostram/VB-P que/C reservas/VB-P@N-P florestais/N-P@ADJ-G-P comunitárias/ADJ-F-P podem/VB-P retardar/VB a/D-F fragmentação/N da/P+D-F floresta/N amazônica/ADJ-F

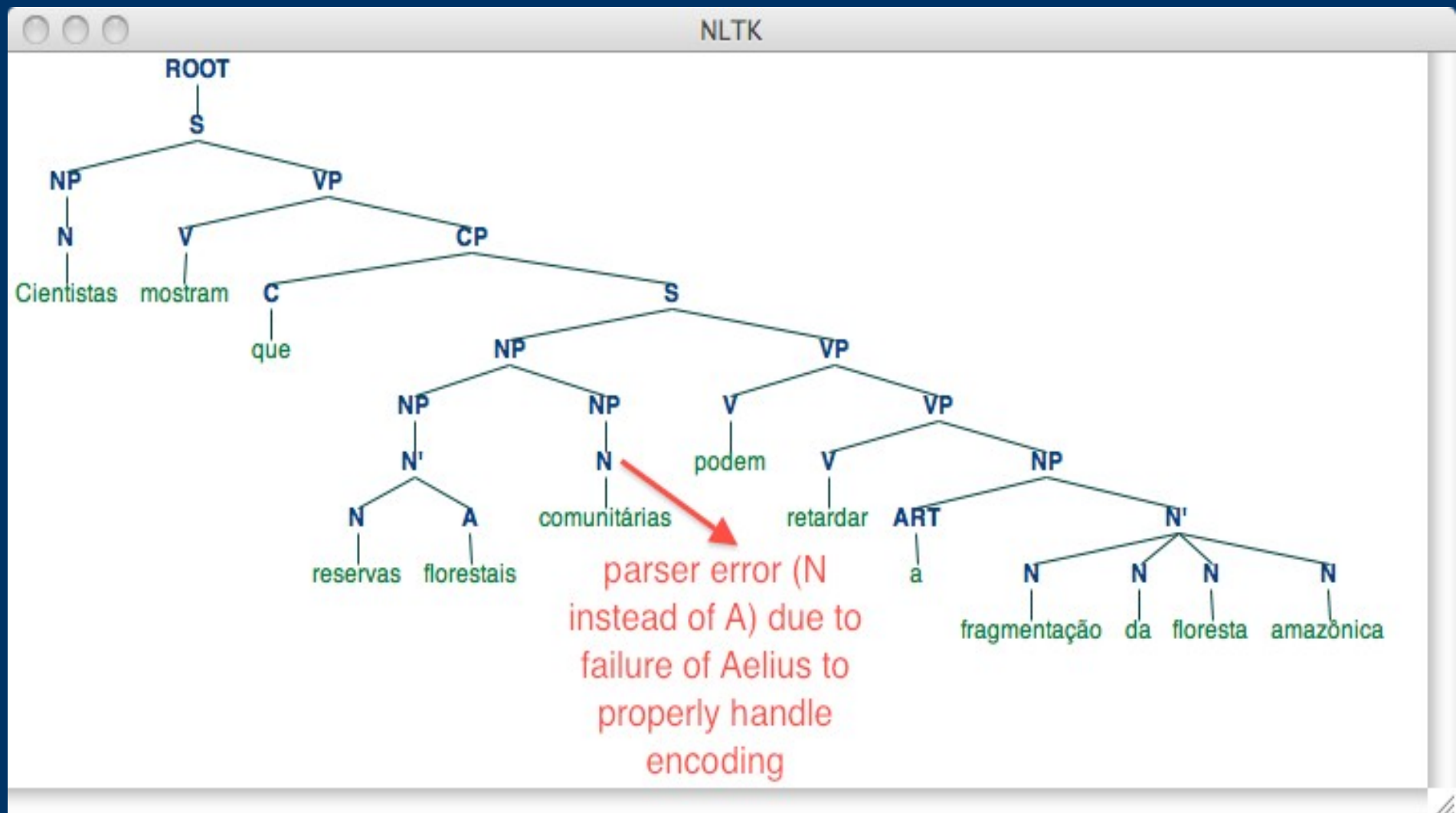
Evaluating tagger accuracy: LX-Tagger

- LX-Tagger: 1 (light) error in 14 tokens
 - Cientistas/PNM@CN mostram/V que/CJ reservas/CN florestais/ADJ comunitárias/ADJ podem/V retardar/INF a/DA fragmentação/CN de/PREP a/DA floresta/CN amazônica/ADJ

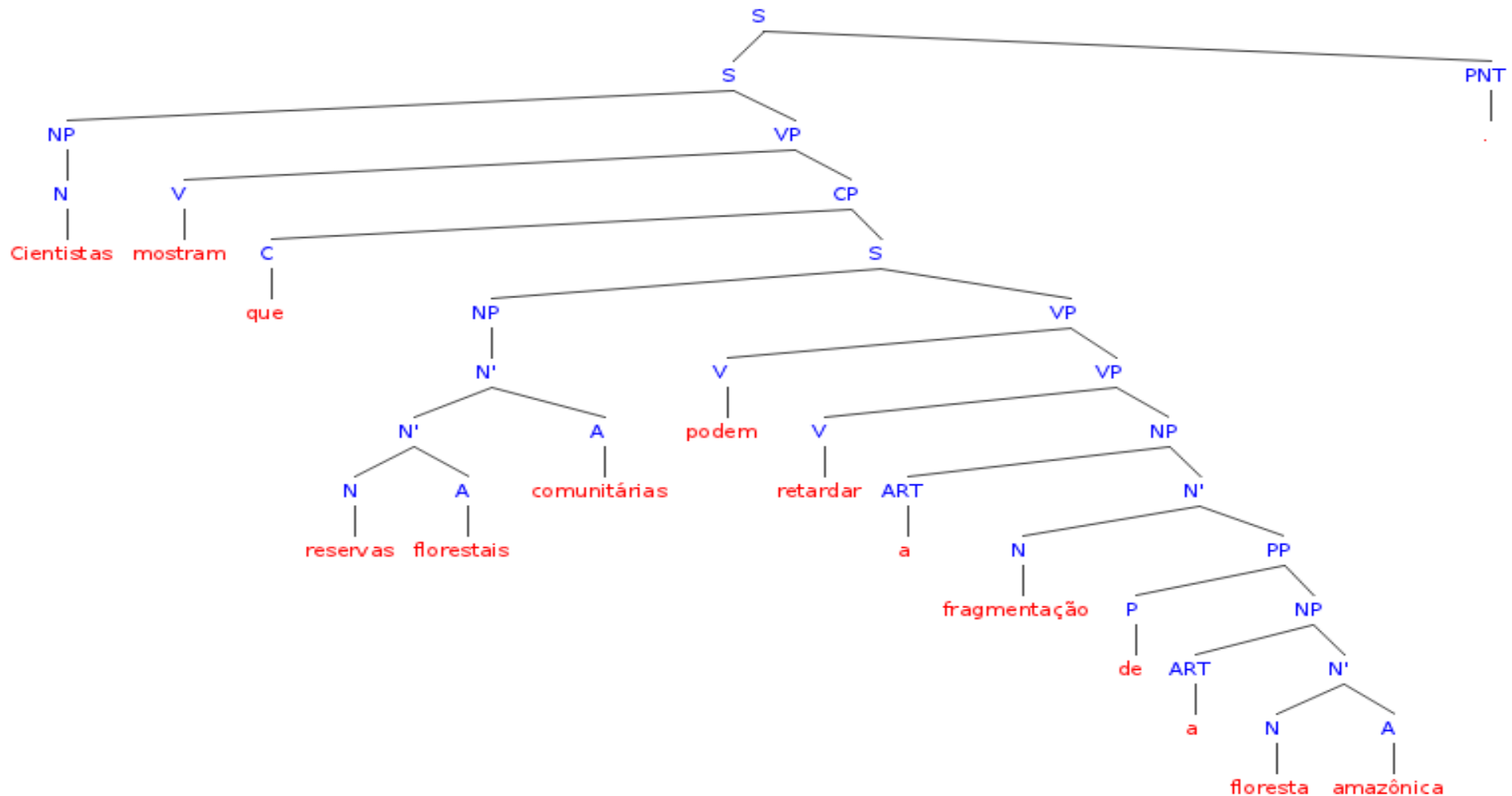
Parsing example

```
>>> tagged_tokens=[(['Cientistas', 'PNM'), ('mostram', 'V'), ('que', 'CJ'), ('reservas', 'CN'), ('florestais', 'ADJ'), ('comunit\xc3\xa1rias', 'ADJ'), ('podem', 'V'), ('retardar', 'IN F'), ('a', 'DA'), ('fragmenta\xc3\xa7\xc3\xa3o', 'CN'), ('de', 'PREP'), ('a', 'DA'), ('floresta', 'CN'), ('amaz\xc3\xb4nica', 'ADJ')]]
>>> from Aelius import StanfordParser
>>> lx="/Users/leonel/stanford-parser/cintil-1.ser.gz"
>>> lxparser=StanfordParser.StanfordParser(lx,encoding="utf-8")
>>> tokens=[w for w,t in tagged_tokens[0]]
>>> tokens
['Cientistas', 'mostram', 'que', 'reservas', 'florestais', 'comunit\xc3\xa1rias', 'podem', 'retardar', 'a', 'fragmenta\xc3\xa7\xc3\xa3o', 'de', 'a', 'floresta', 'amaz\xc3\xb4nica']
>>> lxparser.parse(tokens).draw()
```

Parse tree generated by the LX-Parser via the Aelius interface



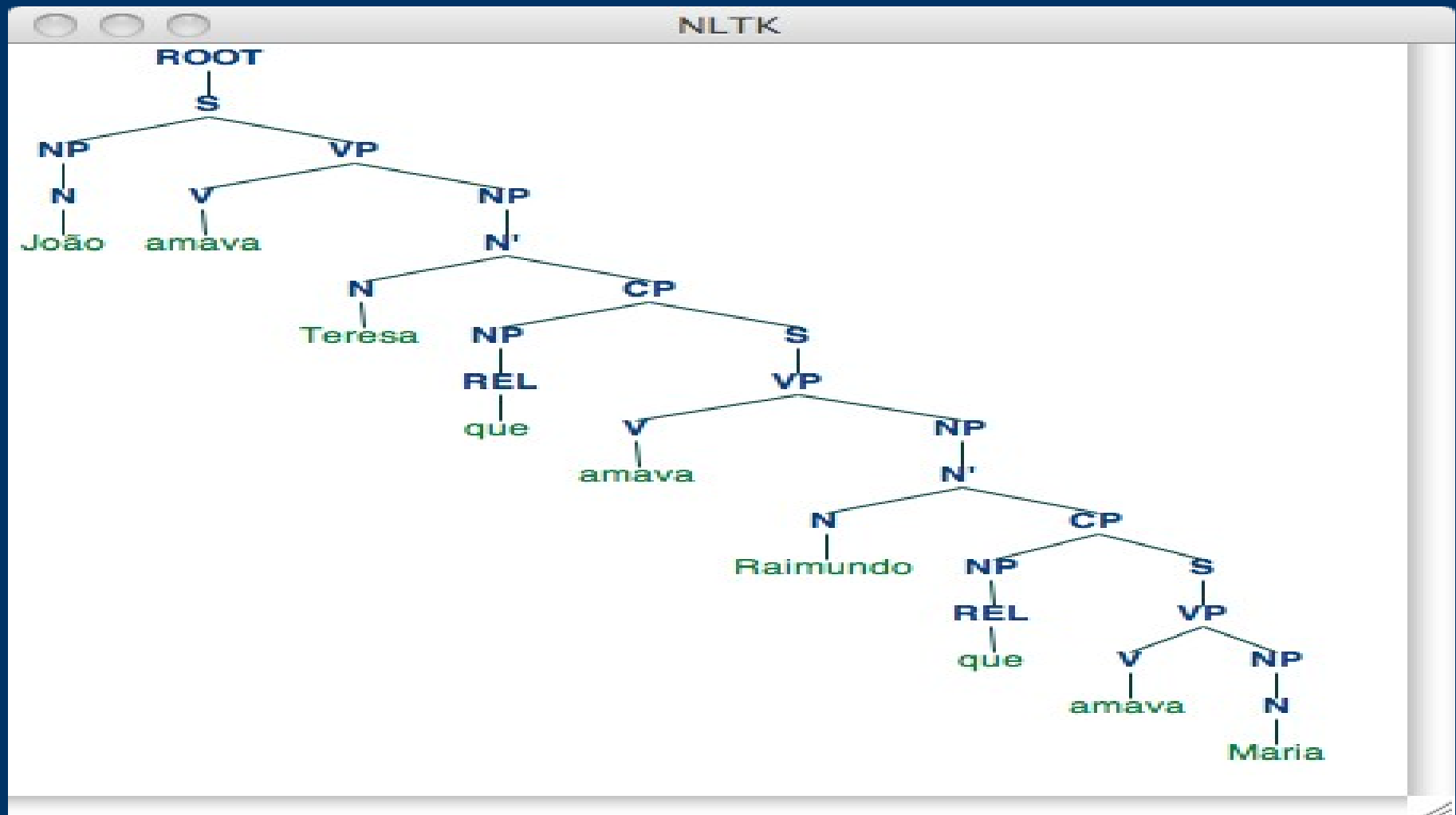
Parse tree generated on-line by the LX-Parser



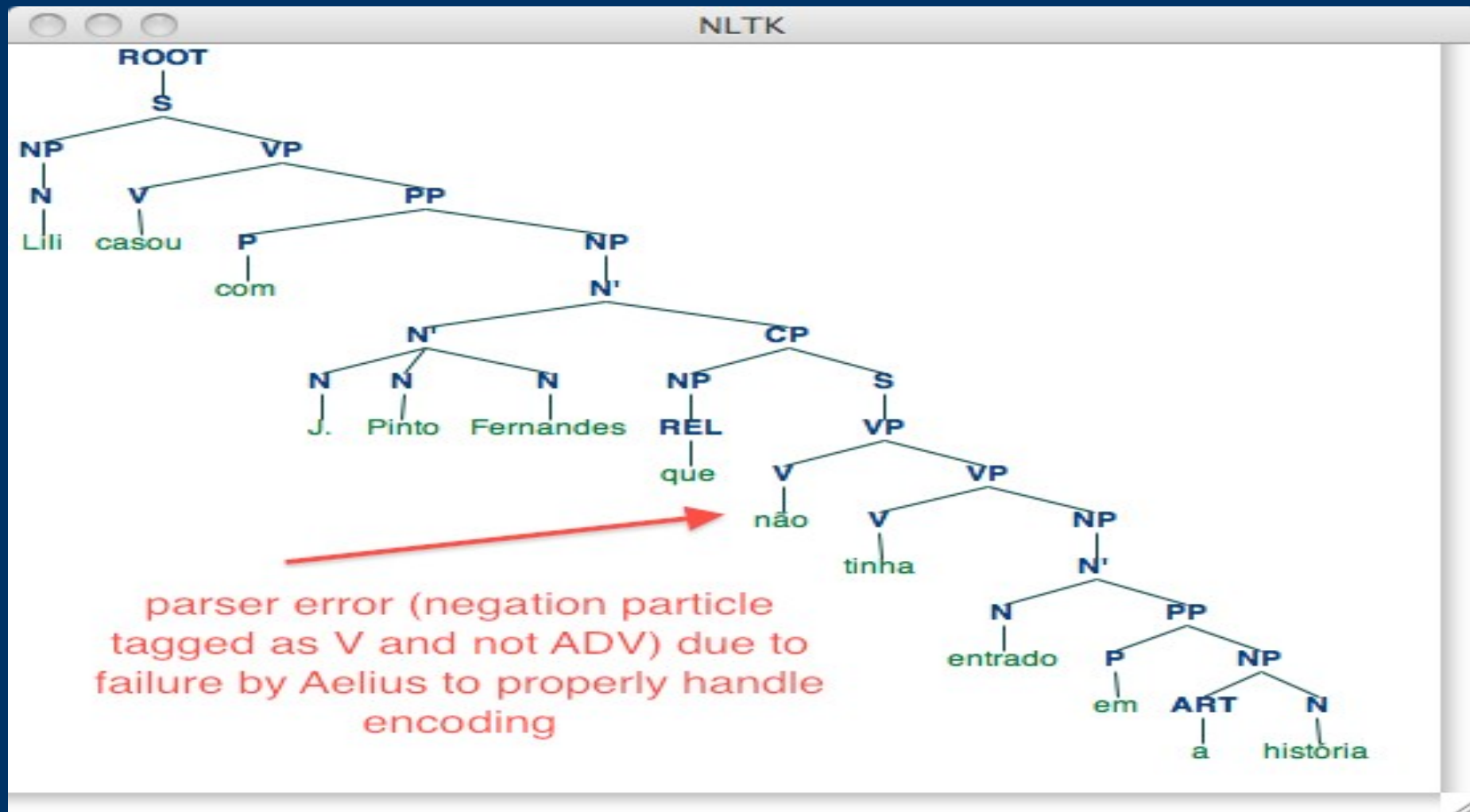
Piping tagger output into the parser

- In the previous example from the Python shell, we extracted the tokens to be parsed by the LX-Parser (via Aelius interface to the StanfordParser) from the output of the LX-Tagger (generated via Aelius)
 - The tagger was used in this case to expand contractions (*da* was split into the preposition *de* and the article *a*), since it efficiently handles ambiguous contractions like *deste*
 - This is a typical example of resource reuse in our project
-
-

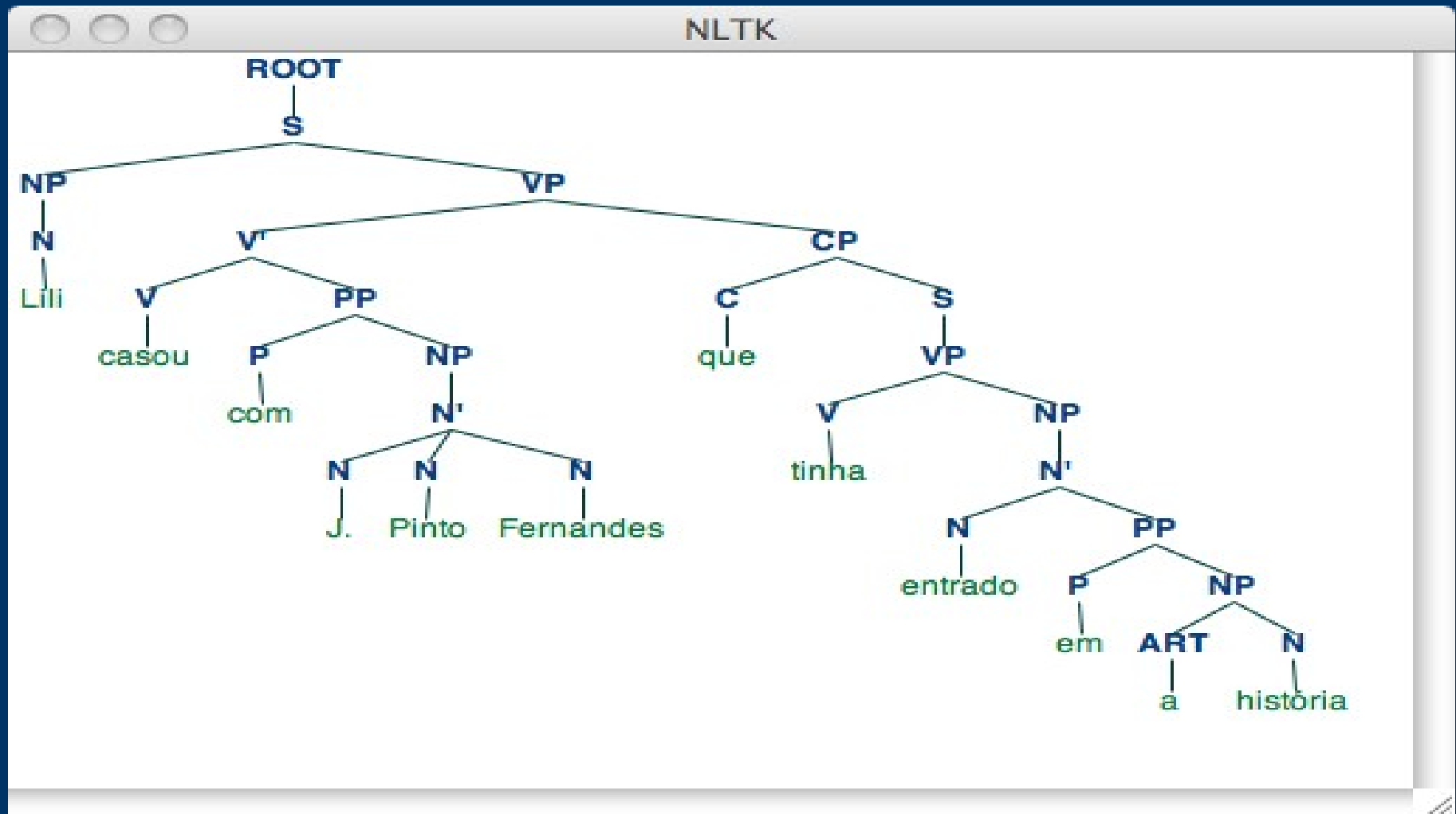
Drummond analyzed by the LX-Parser via Donatus



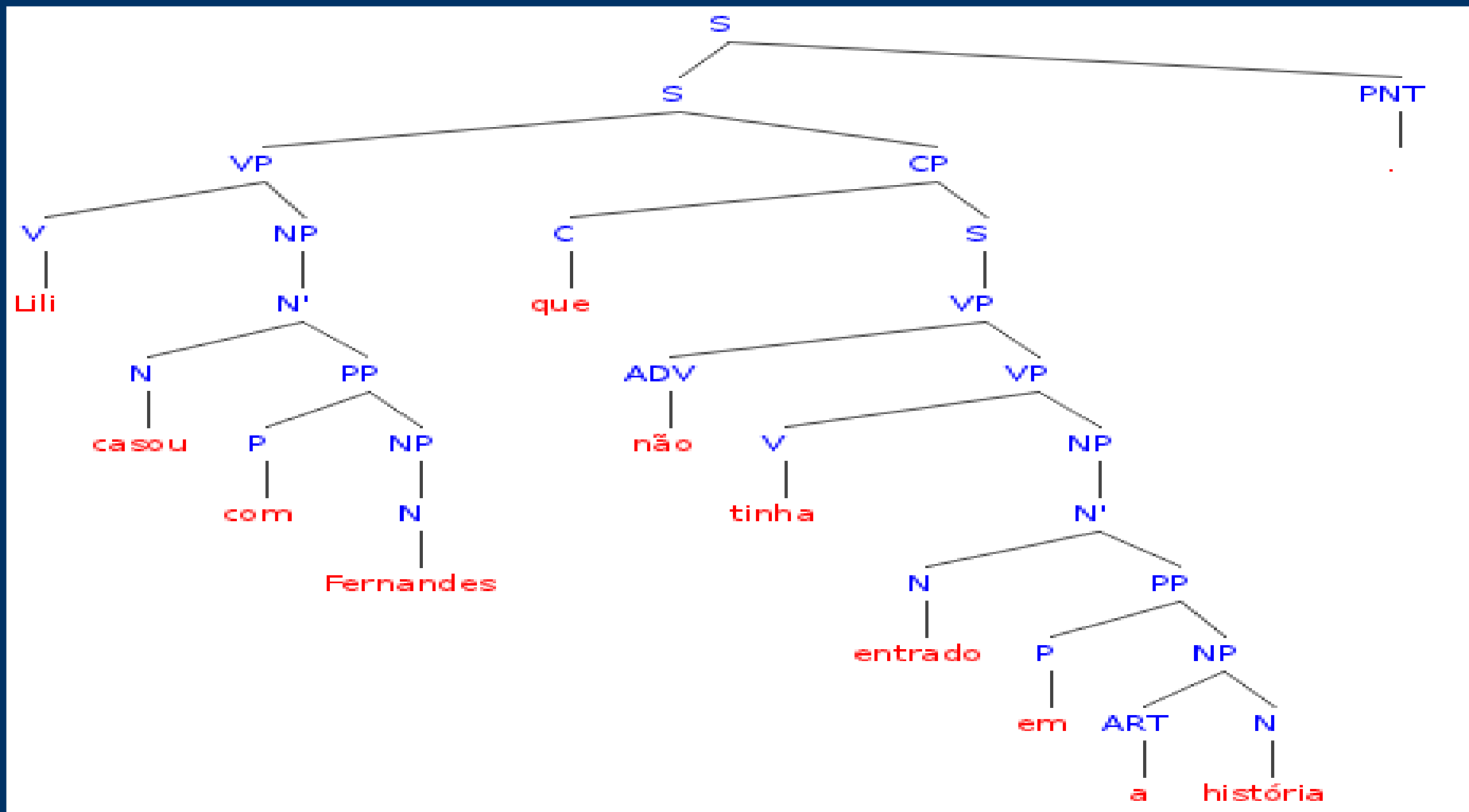
Performance decrease of the LX-Parser: example #1



Performance decrease of the LX-Parser: example #2



Parsing failure in the on-line version of the LX-Parser



The Lexicon Bottleneck in Deep Parsing

- We hypothesize that the main problem to overcome in developing a robust parser for unrestricted text is modeling the lexicon, majorly due to
 - The sheer amount of entries needed ($> 2M$)
 - Categorial ambiguity of words
 - Productive word formation processes
 - Non-standard spelling(ALENCAR, 2011)

Overcoming the Lexicon Bottleneck

- We propose to handle these difficulties using taggers as lexicon analyzers (ALENCAR, 2011), instead of modeling lexical morphosyntactic knowledge through finite-state transducers or extracting these informations from corpora



A simple NLTK toy grammar for Drummond's language

S -> NP VP

VP -> V | V PP | V NP | V VP | VP PP | ADV VP

NP -> D N

NP -> NP S

NP -> D PNM | REL | CL

PP -> P NP

N -> CN

P -> PREP

D -> DA |

V -> PPA



Generating lexicon entries on the fly from tagger output

- The previous grammar has no lexical entries
- These are generated on the fly by the Donatus module ALEXP from the output of a tagger
- Example using LX-Tagger via Aelius:

```
Lili/PNM casou/V com/PREP  
J/PNM ./PNT Pinto/PNM  
Fernandes/PNM que/REL não/ADV  
tinha/V entrado/PPA em/PREP a/DA  
história/CN ./PNT
```

Handling complex proper names

- The chunker is applied to the tagger output to identify complex proper names
- The individual elements of proper name chunks are joined with an underscore “_”

J/PNM ./PNT Pinto/PNM Fernandes/PNM

J_Pinto_Fernandes/PNM

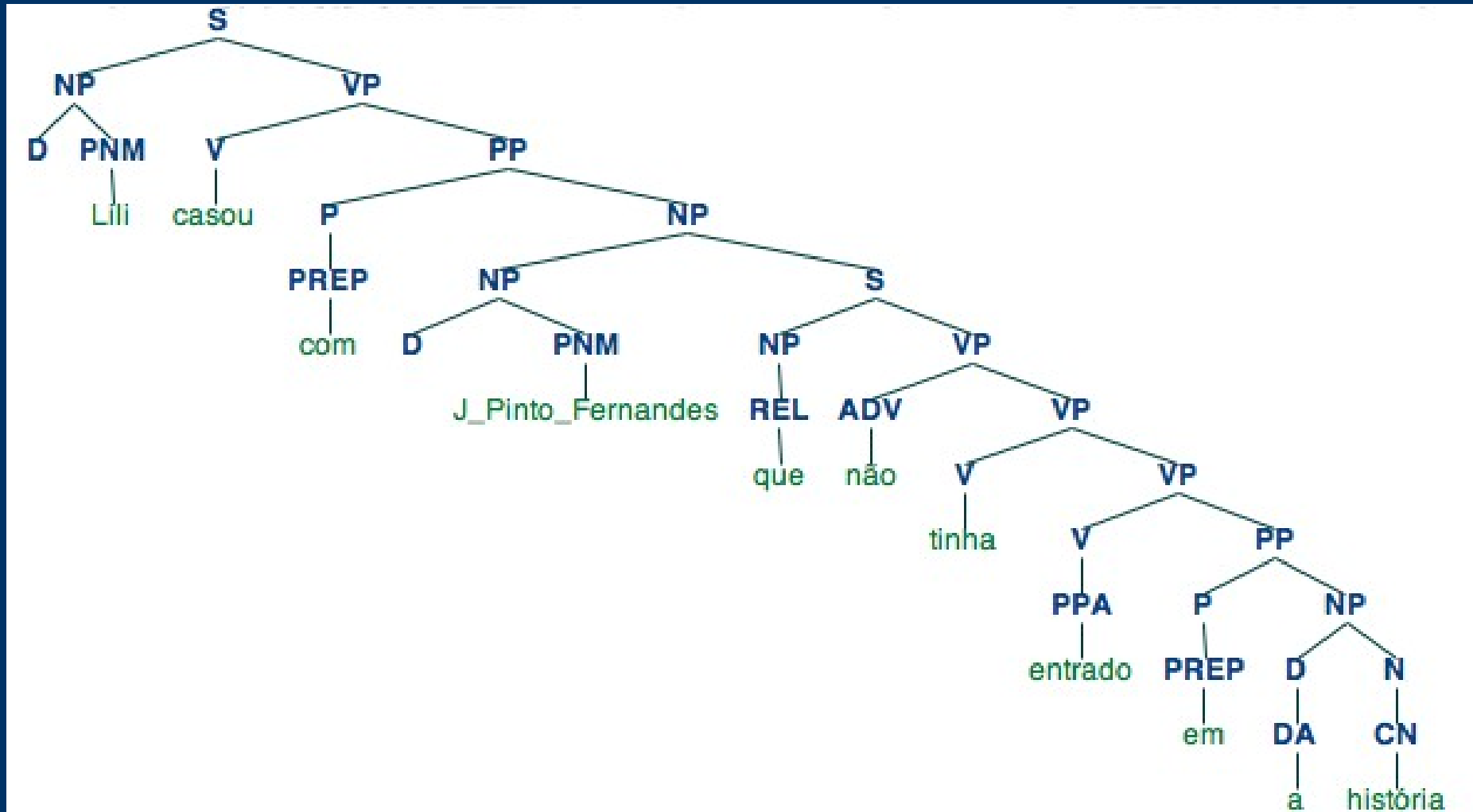
ALEXP: The Tagger-Parser Interface

```
>>> alexp.analisaBlocoDeSentencas(g,anotado="drummond.quadrilha.mxpost.02.txt")
Nr. 1: 1 análise(s): João amava Teresa
Nr. 2: 1 análise(s): João não amava Teresa
Nr. 3: 1 análise(s): João amava Teresa que amava Raimundo
Nr. 4: 2 análise(s): Raimundo amava Maria que amava Joaquim que amava Lili
Nr. 5: 1 análise(s): Lili não amava ninguém
Nr. 6: 2 análise(s): João foi para os Estados_Unidos
Nr. 7: 2 análise(s): Raimundo morreu de desastre
Nr. 8: 1 análise(s): Joaquim suicidou se
Nr. 9: 10 análise(s): Lili casou com J_Pinto_Fernandes que não tinha entrado em a história

100.00% analisadas de um total de 9 sentenças
2.33 análise(s) por sentença

>>> alexp.mostraArvores(9)
```

A correct parse tree generated by our toy grammar



Concluding remarks

- AeliusDonatus is still mostly a single person's research effort
- We are thankful to our students who have collaborated in preparing gold standards for evaluating tagger and chunker performance



Concluding remarks

- In many aspects, our work is complementary to the following PhD research projects under our supervision:
 - Tiago Martins da Cunha's work on a RBMT system for English nominal expressions (CAPES doctoral sandwich scholarship – Universität des Saarlandes, Germany, 2012, supervised by Johann Haller)
 - Andréa Feitosa dos Santos' future implementation of a LFG fragment for BP (DAAD doctoral sandwich scholarship – Universität Konstanz, Germany, March 2012 – February 2014, supervised by Miriam Butt and Georg A. Kaiser)
-
-

Concluding remarks

- As a FOSS effort, collaboration by other people is welcome!
- Annotated corpus data covering social, regional, and diaphasic variation of BP are urgently needed.



Thank you!



References

ALENCAR, L. F. de. Aelius: uma ferramenta para anotação automática de corpora usando o NLTK. **ELC 2010 – IX Encontro de Linguística de Corpus**, PUCRS, Porto Alegre, 8 e 9 de outubro de 2010.

On-line: <<http://corpuslg.org/gelc/elc2010.php>>

ALENCAR, L. F. de. Utilização de informações lexicais extraídas automaticamente de corpora na análise sintática computacional do português. **Revista de Estudos da Linguagem**, Belo Horizonte, vol. 19, n. 1, jan./jun. 2011. To appear.



References

BRANCO, A.; SILVA, J. 2004. Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese. In: LINO, M. T. et al. (Eds.). INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION, n. 4, 2004, Lisboa. *Proceedings...* Paris: ELRA, 2004. p. 507-510.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural language processing with Python: analyzing text with the Natural Language Toolkit*. Sebastopol: O'Reilly, 2009. 502 p.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Toolkit*. [s.l]: [s.n.], 2011. On-line: <<http://www.nltk.org>>.

References

JUNGEN, O.; LOHNSTEIN, H. *Einführung in die Grammatiktheorie*. München: Wilhelm Fink, 2006.

HAJIČOVÁ, E. et al. Treebank annotation. In: INDURKHAYA, N.; DAMERAU, F. J. *Handbook of Natural Language Processing*. 2. ed. Boca Raton, FL: Chapman & Hall/CRC, 2010. p. 167-188.

HALÁCSY, P.; KORNAI, A. ; ORAVECZ, C. HunPos: an open source trigram tagger. ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, n. 45, 2007, Praga. *Proceedings...* Stroudsburg: Association for Computational Linguistics, 2007. p. 209-212.

References

LJUNGLÖF, P. ; WIRÉN, M. Syntactic parsing. In: INDURKHYA, N.; DAMERAU, F. J. (Eds.). *Handbook of Natural Language Processing*. 2. ed. Boca Raton, FL: Chapman & Hall/CRC, 2010. p. 59-91.

MAIER, W. NeGra und TüBa-D/Z: ein Vergleich. In: REHM, G.; WITT, A.; LEMNITZER, L. (Eds.). *Data structures for linguistic resources and applications: Proceedings of the Biennial GLDV Conference 2007*. Tübingen: Gunter Narr, 2007. p. 29-38.

References

OTHERO, G. A. *Teoria X-barra: descrição do português e aplicação computacional*. São Paulo: Contexto, 2006. 160 p.

OTHERO, G. A. *A gramática da frase em português: algumas reflexões para a formalização da estrutura frasal em português*. Porto Alegre: Edipucrs, 2009. 160 p. On-line:<
<http://www.pucrs.br/edipucrs/gramaticadafrase.pdf>>.

References

RADFORD, A. *Transformational grammar: a first course*. Cambridge: Cambridge University Press, 1988.

RATNAPARKHI, A. A Maximum Entropy Model for Part-Of-Speech Tagging. EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING, 1996, Philadelphia, Pennsylvania. *Proceedings...* Pennsylvania: University of Pennsylvania, 1996. p. 133-142. On-line: <http://acl.ldc.upenn.edu/W/W96/W96-0213.pdf>.
