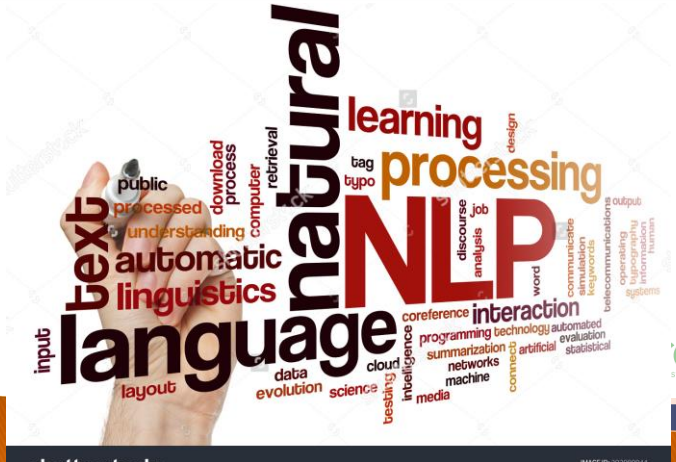
Last Lecture: Main points to recall



7-abr-18

Emerging Applications in TM

making good progress

mostly solved

still really hard

Spam detection
- Let's go to Agra! ✓
- Buy V1AGRA ... ✗

Part-of-speech (POS) tagging

ADJ ADJ NOUN VERB ADV
Colorless green ideas sleep furiously.

Named entity recognition (NER)

PERSON ORG LOC
Einstein met with UN officials in Princeton

Sentiment analysis
- Best roast chicken in San Francisco!
- The waiter ignored us for 20 minutes.

Coreference resolution
- Carter told Mubarak he shouldn't run again.

Word sense disambiguation
- I need new batteries for my *mouse*.

Parsing
- I can see Alcatraz from the window!

Machine translation (MT)
- 第13届上海国际电影节开幕...
- The 13th Shanghai International Film Festival...

Information extraction (IE)
- You're invited to our dinner party, Friday May 27 at 8:30
- Party May 27 add

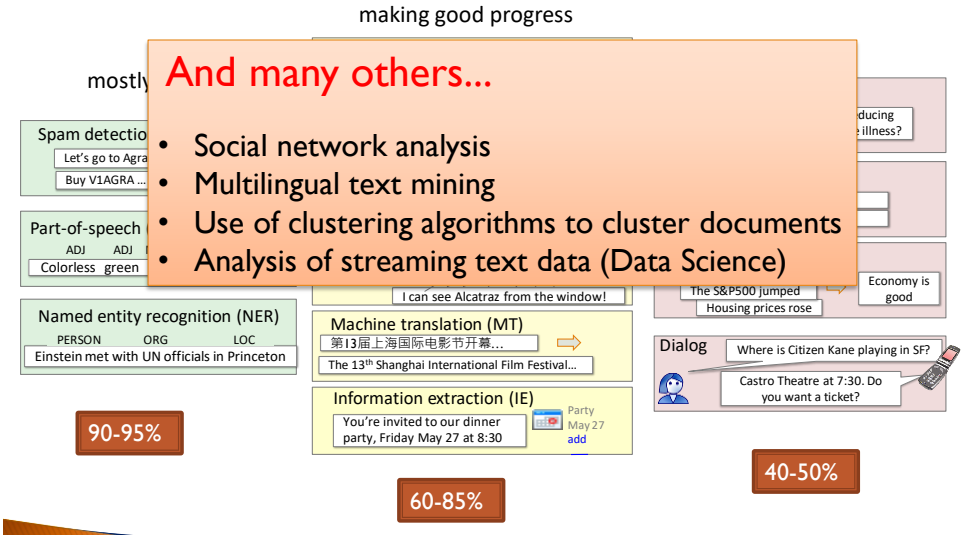
Question answering (QA)
- Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase
- XYZ acquired ABC yesterday
- ABC has been taken over by XYZ

Summarization
- The Dow Jones is up
- The S&P500 jumped
- Housing prices rose
- ⇨ Economy is good

Dialog
- Where is Citizen Kane playing in SF?
- Castro Theatre at 7:30. Do you want a ticket?

Emerging Applications in TM



And many others...

- Social network analysis
- Multilingual text mining
- Use of clustering algorithms to cluster documents
- Analysis of streaming text data (Data Science)

3

Why NL understanding is difficult?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either♥

segmentation issues

the New York-New Haven Railroad
the New York-New Haven Railroad

idioms

dark horse
get cold feet
lose face
throw in the towel

neologisms

unfriend
Retweet
bromance

world knowledge

Mary and Sue are sisters.
Mary and Sue are mothers.

tricky entity names

Where is *A Bug's Life* playing ...
Let It Be was recorded ...
... a mutation on the *for* gene ...

Ambiguity

include your children when baking cookies

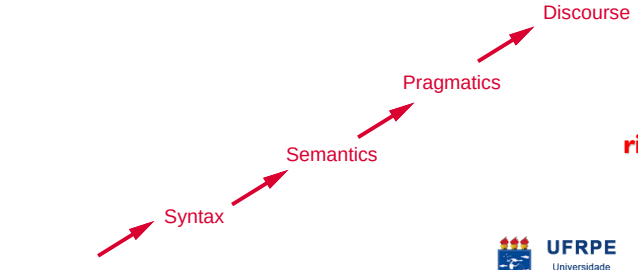
two soviet ships collide, one dies

But that's what makes it fun!

4

Processamento de Linguagem Natural

Aula 02: Introduction to NLP



Prof. Rinaldo Lima
rinaldo.ufrpe@gmail.com

UFRPE Universidade Federal Rural de Pernambuco | DEINFO Departamento de Estatística e Informática

7-abr-18

Contents

- **NLP Corpora**
- **NLP subtasks**
  1. Tokenization
  2. Sentence Splitting
  3. Lemmatization/Stemming
  4. POS tagging
  5. NER
  6. Parsing
  7. Chunking
  8. Coreference Resolution
  9. Semantic Roles
- **Applying NLP substasks in many TM applications**

NLP CORPORA (DATASETS)

UFRPE Universidade Federal Rural de Pernambuco | DEINFO Departamento de Estatística e Informática

7-abr-18

What is a corpus?

A corpus is a finite machine-readable body of naturally occurring text, selected according to specified criteria, eg:

- **Language and type:** English/German/Arabic/…, dialects v. "standard"
- **Genre and Domain**: 18th century novels, newspaper text, software manuals…
- **Web as Corpus**: URL "domain" = country: .uk .ar
- **Media**: "Written" Text, Audio, Transcriptions, Video.
- **Size**: 1000 words, 50K words, 1M words, 100M words,

Corpora Examples : Brown and LOB

Brown: The most famous corpus! (well, first widely-used corpus)

- A balanced corpus: representative of a whole language
- Brown: published American English from 1960s (newspapers, books, etc)
- 1 million words, POS tagged.

It will be used in our assignments

LOB: Lancaster-Oslo Bergen corpus

- British English version
- published British English text from 1960s

10

Some recent corpora

Corpus features: Size, Domain, Language

- **British National Corpus**: 100M words, balanced British English
- Newswire Corpus: 600M words, newswire, American English
- UN or EU proceedings: 20M+ words, legal, 10 language pairs
- Penn Treebank: 2M words, newswire American English
- Web: 8 billion(?) words, many domains and languages
- Web-as-Corpus:
  - harvest your own corpus from WWW,
  - via "seed terms" → Google API → web-pages → Corpus!

11

Corpus Annotation

Annotation is a process in which linguistics experts add (linguistic) information to the corpus that is not explicitly there (increases utility of a corpus), e.g.:

- Text Headers: meta-data for each text: author, date, type,…
- Part-of-speech tag for each word (very common!).
- Syntactic structure: parse-tree for each sentence
- Word Sense label for each word
- Prosodic information: pauses, rise and fall in pitch, etc.

12

Annotation with XML: Text Centric vs Data Centric XML

Listing 2.1 Purchase Order in XML

```
<po id="43871" submitted="2001-10-05">
   <billTo>
      <company>The Skateboard Warehouse</company>
      <street>One Warehouse Park</street>
      <street>Building 17</street>
      <city>Boston</city>
      <state>MA</state>
      <postalCode>01775</postalCode>
   </billTo>
   <shipTo>
      <company>The Skateboard Warehouse</company>
      <street>One Warehouse Park</street>
      <street>Building 17</street>
      <city>Boston</city>
      <state>MA</state>
      <postalCode>01775</postalCode>
   </shipTo>
   <order>
```

DATA CENTRIC XML DOC

TEXT CENTRIC XML DOC

```
<H1>Skateboard Usage Requirements</H1>
<P>In order to use the <B>FastGlide</B> skateboard you have to
have:</P>
<LIST>
<ITEM>A strong pair of legs.</ITEM>
<ITEM>A reasonably long stretch of smooth road surface.</ITEM>
<ITEM>The impulse to impress others.</ITEM>
</LIST>
<P>If you have all of the above, you can proceed to <LINK
HREF="Chapter2.xml">Getting on the Board</LINK>.</P>
```

13

Annotation Example: POS tagging

Some texts are annotated with Part-of-speech (POS) tags.

- POS tags encode simple grammatical functions.

<s><w pos=RN> Here </w> <w pos=BEZ> **is** </w> <w pos=IN> **a** </w><w pos=NN> **sentence** </w>.</s>

Several tag sets are available:

- Brown tag set (**87 tags**) in Brown corpus
  - Penn tag set (45 tags) in Penn Treebank
- CLAWS / LOB tag set (**132 tags**) in LOB corpus
  - CLAWS c5 tag set (62 tags) in BNC (British National Corpus)

- Tagging is usually done automatically (then proofread and corrected by humans)

Corpora Annotation: Stand-off vs Inline

- **Inline:**
  - Data and metadata (annotation or markup) are combined together into a single file.

- **Stand-off:**
  - Metadata is stored in a separate document, using reference **anchors**
  - Alignment based on token or **character offsets**
  - Primary data is left untouched

Inline Annotation Example - POS tagging

```
John went to Paris yesterday. He loved the excursion.
```

```
John_NNP went_VBD to_TO
Paris_NNP yesterday_NN ._.
He_PRP loved_VBD the_DT
excursion_NN ._.
```

Horizontal format

```
John_NNP
went_VBD
to_TO
Paris_NNP
yesterday_NN
._.
He_PRP
loved_VBD
the_DT
excursion_NN
._.
```

Vertical format

16

Stand-off Annotation Example - POS tagging

```
12345678901234567890123456789012345678901234567890123
        1         2         3         4         5
John went to Paris yesterday. He loved the excursion.
```

Character Offsets

```
1  4  NNP
6  9  VBD
11 12 TO
14 18 NNP
20 28 NN
29 29 .
31 32 PRP
34 38 VBD
40 42 DT
44 52 NN
53 53 .
```

17

Stand-off Annotation: a more complex example

"Layers" of annotations in several separated XML files

text.xml:

```
...</header>
<body>Fürchtet euch nicht ! Die einstige Fußball-Weltmacht zittert vor einem Winzling
. Mit seinem To... zum 1:0 für die Ukraine stürzte der 1,62 Meter große Gennadi Subow die
deutsche Nationalelf...</body>
```

tok.xml:

```
<markList type="token" xml:base="text.xml">
  <mark id="t1" xlink:href="#xpointer(string-range(//body,'',0,8))"/> <!-- Fürchtet -->
  <mark id="t2" xlink:href="#xpointer(string-range(//body,'',9,4))"/> <!-- euch -->
  <mark id="t3" xlink:href="#xpointer(string-range(//body,'',14,5))"/> <!-- nicht -->
  <mark id="t4" xlink:href="#xpointer(string-range(//body,'',20,1))"/> <!-- ! -->
  <mark id="t5" xlink:href="#xpointer(string-range(//body,'',22,3))"/> <!-- Die -->
  ...
```

infStat.xml:

```
<featList type="information_status" xml:base="tok.xml">
  ...
  <!-- euch: new -->
  <feat xlink:href="#xpointer(id('t2'))" value="type_infStat.xml#new"/>
  <!-- Die einstige Fußball-Weltmacht: accessible -->
  <feat xlink:href="#xpointer(id('t5')/range-to(id('t7')))" value="type_infStat.xml#acc"/>
  ...
```

type_infStat.xml:

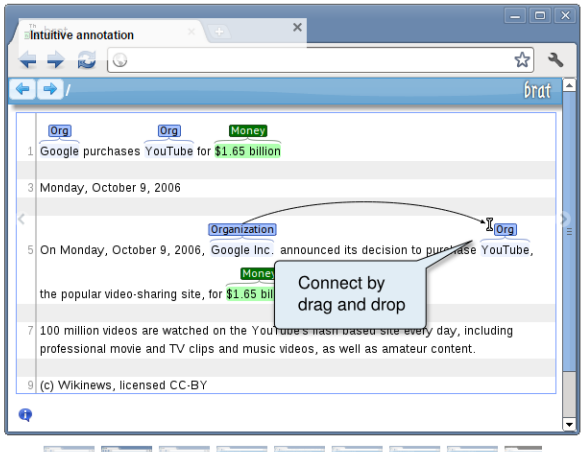
```
<typeList type="information_status">
  <type id="giv" name="giv" descr="The referent is given in the discourse.">
    <type id="new" name="new" descr="The referent is new in the discourse."/>
    <type id="acc" name="acc" descr="The referent is accessible."/>
    ...
  </type>
  ...
```

18

Corpora Annotation Tools

brat rapid annotation tool http://brat.nlplab.org/

online environment for collaborative text annotation



Learn more:

- What is it?
- What can you do with it?
- What does it do?
- What do I need to run it?

Try brat online

(username: "crunchy", password: "frog")

Take a tutorial: news, (reset), bio (reset),

Runs in your browser: no installation required

Intuitive annotation visualization and editing.

Create your own local brat installation:

Download v1.3

(MD5, SHA512, Repository (GitHub), Older versions)

Manage your own annotation effort

Easy to set up: installation instructions

Instructions for upgrading to v1.3 (Crunchy Frog)

Open source (MIT License)

Current version: v1.3 Crunchy Frog (2012-11-08).

19

Corpora Annotation Tools

Annotation & Text-Processing Tools

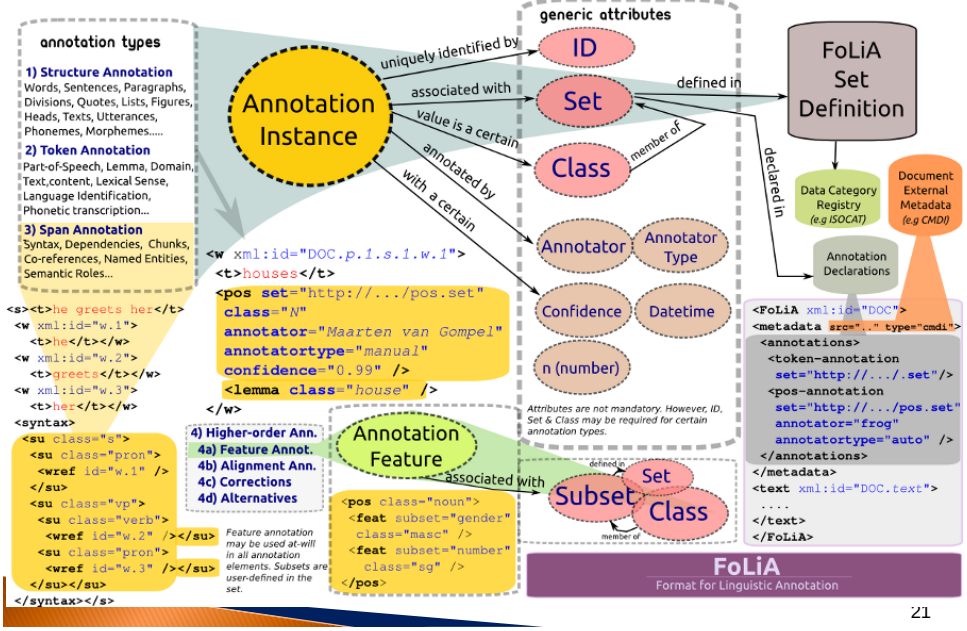
Text Coding/(Manual) Annotation Programs/Text-analysis Tools & Search Engines

Please note that some of these programs produce XML files in *standoff* format, which separates the text into different linked levels. The advantage of this type possible to link various types of annotation to the same set of data, but the disadvantage is that it's usually not possible to 'interact' directly with that data unles interface it's been created with. In other words, creating standoff annotations usually ties one into specific programs and the functionality the provide for merging

DART (Dialogue Annotation & Research Tool	An annotation & analysis tool designed for the semi-automatic annotation of spoken (transcribed) dialogues on the levels of syntax, prag (surface) polarity, semantics (topics), & semantico-pragmatics (modes, 'IFIDs'). DART produces annotations in what I refer to as 'Simple XML', a highly readable format that still allows the corpus user to 'interact' w perform corrections, add additinal annotations, etc. As a research, tool, DART also offers facilities for the creation of dialogue corpora & and their associated analysis resources, a built-in analysis, as well as speech-act statistics. For a detailed description, see my recent article in Corpus Linguistics and Linguistic Theory about version 1. Version 2, as well as DART's 'big brother', the Text Analysis & Research Tool (TART), are currently under developm

http://martinweisser.org/corpora_site/annotation_tools.html

FOLIA – Format for Linguistic Annotation (2012)
Radboud University - http://proycon.github.io/folia/



NLP Subtasks

UFRPE Universidade Federal Rural de Pernambuco | DEINFO Departamento de Estatística e Informática

7-abr-18

Levels of NLP Analysis

- Morphology: Concerns the way words are built up from smaller meaning bearing units. (come(s), walk(ed))
- Syntax: concerns how words are put together to form correct sentences and what structural role each word has.
- Semantics: concerns what words mean and how these meanings combine in sentences to form sentence meanings.
- Pragmatics: concerns how sentences are used in different situations and how its use affects the interpretation of the sentence.
- Discourse: concerns how the immediately preceding sentences affect the interpretation of the next sentence.

Tokenization (Word Segmentation)

Tokenization is a processing step where the input text is automatically divided into units called tokens where each is either a word or a number or a punctuation mark

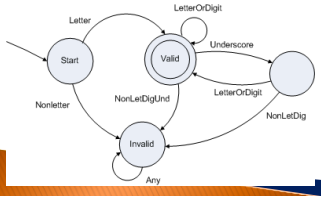
- In some written languages (e.g. Chinese) words are not separated by spaces.
- Even in English, characters other than white-space can be used to separate words [e.g. , ; . - : ( )]

Issues in Tokenization

- `Finland's capital` → `Finland Finlands Finland's ?`
- `what're, I'm, isn't` → `What are, I am, is not`
- `Hewlett-Packard` → `Hewlett Packard ?`
- `state-of-the-art` → `state of the art ?`
- `Lowercase` → `lower-case lowercase lower case ?`
- `San Francisco` → one token or two?
- m.p.h., PhD. → ??

Tokenization Algorithm for English

A finite automata




```
Initialize:
  Set Stream to the input text string
  Set currentPosition to 0 and internalQuoteFlag to false
  Set delimiterSet to ' , . ; : ! ? ( ) <>+"\n\t space
  Set whiteSpace to \t\n space
Procedure getNextToken:
  L1: cursor := currentPosition; ch := charAt(cursor);
    If ch = endOfStream then return null; endif
  L2: while ch is not endOfStream nor instanceOf(delimiterSet) do
        increment cursor by 1; ch := charAt(cursor);
      endwhile
      If ch = endOfStream then
        If cursor = currentPosition then return null; endif
      endif
      If ch is whiteSpace then
        If currentPosition = cursor then
          increment currentPosition by 1 and goto L1;
        else
          Token := substring(Stream,currentPosition,cursor-1);
          currentPosition := cursor+1; return Token;
        endif
      endif
      If ch = ' then
        If charAt(cursor-1) = instanceOf(delimiterSet) then
          internalQuoteFlag := true; increment currentPosition by 1; goto L1;
        endif
        If charAt(cursor+1) != instanceof(delimiterSet) then
          increment cursor by 1; ch := charAt(cursor); goto L2;
        elseif internalQuoteFlag = true then
          Token := substring(Stream,currentPosition,cursor-1);
          internalQuoteFlag := false;
        else
          Token := substring(Stream,currentPosition,cursor);
        endif
        currentPosition := cursor+1; return Token;
      endif
      If cursor = currentPosition then
        Token := ch; currentPosition := cursor+1;
      else
        Token := substring(Stream,currentPosition,cursor-1);
        currentPosition := cursor;
      endif
      return Token;
endprocedure
```

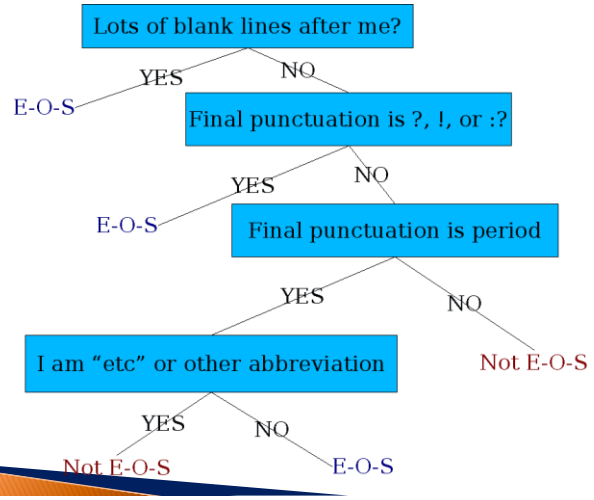
Fig. 2.2 Tokenization algorithm

Sentence Segmentation (Splitting)

- !, ? are relatively unambiguous
- Period "." is quite ambiguous
  - Sentence boundary
  - Abbreviations like Inc. or Dr.
  - Numbers like .02% or 4.3
- Build a binary classifier
  - Looks at a "."
  - Decides EndOfSentence/NotEndOfSentence
  - Classifiers:
    - hand-written rules,
    - regular expressions, or
    - machine-learning

Sentence Segmentation (Splitting)

Determining if a word is end-of-sentence: a Decision Tree



28

Input: a text with periods
Output: same text with End-of-Sentence (EOS) periods identified

Sentence Splitting Algorithm for English

Ruled-based

Rules:
All ? ! are EOS
If " or ' appears before period, it is EOS
If the following character is not white space, it is not EOS
If) }] before period, it is EOS
If the token to which the period is attached is capitalized and is < 5 characters and the next token begins uppercase, it is not EOS
If the token to which the period is attached has other periods, it is not EOS
If the token to which the period is attached begins with a lowercase letter and the next token following whitespace is uppercase, it is EOS
If the token to which the period is attached has < 2 characters, it is not EOS
If the next token following whitespace begins with $ ({ [" ' it is EOS
Otherwise, the period is not EOS

29

Morphological Analysis

- **Morphology** is the field of linguistics that studies the internal structure of words
- A *morpheme* is the smallest linguistic unit that has semantic meaning
  - e.g. "carry", "pre", "ed", "ly", "s"
- Morphological analysis is the task of segmenting a word into its morphemes:
  - carried ⇒ carry + ed (past tense)
  - independently ⇒ in + (depend + ent) + ly
  - Googlers ⇒ (Google + er) + s (plural)
  - unlockable ⇒ un + (lock + able) ?
    ⇒ (un + lock) + able ?

- **Morphemes:**
  - The small meaningful units that make up words
  - **Stems**: The core meaning-bearing units
  - **Affixes**: pieces that adhere to stems

30

Tokenization: language issues

- **French**
  - *L'ensemble* → one token or two?
    - *L ? L' ? Le ?*
- **German noun compounds are not segmented**
  - *Lebensversicherungsgesellschaftsangestellter*
    - 'life insurance company employee'
  - German information retrieval needs compound splitter

31

Lemmatization

- **Reduce inflections or variant forms to base form. Dictionary form of a word.**
  - *am, are, is* → *be*
  - *car, cars, car's, cars'* → *car*
- ***the boy's cars are different colors* → *the boy car be different color***
- **Most used in Machine translation**
  - Spanish quiero ('I want'), quieres ('you want') same lemma as querer 'want'

Stemming

- **Reduce terms to their stems or roots**
- ***Stemming* is the crude chopping of affixes**
  - language dependent
  - e.g., *automate(s), automatic, automation* all reduced to *automat*.

for example compressed and compression are both accepted as equivalent to compress.

for exampl **compress** and **compress** ar both accept as equival to compress

Porter's algorithm: the most used English stemmer

Step 1a

```
sses → ss   caresses → caress
ies  → i    ponies   → poni
ss   → ss   caress   → caress
s    → ø    cats     → cat
```

Step 1b

```
(*v*)ing → ø  walking  → walk
              sing     → sing
(*v*)ed  → ø  plastered → plaster
…
```

Step 2 (for long stems)

```
ational→ ate relational→ relate
izer→ ize    digitizer → digitize
ator→ ate    operator  → operate
…
```

Step 3 (for longer stems)

```
al    → ø  revival    → reviv
able  → ø  adjustable → adjust
ate   → ø  activate   → activ
…
```

Why only strip –ing if there is a vowel?

```
(*v*)ing → ø  walking  → walk
              sing     → sing
```


Stemming Algorithm for English

Ruled-based

Input: a text token and a dictionary
Doubling consonants: b d g k m n p r l t

Rules:
If token length < 4 **return** token
If token is number **return** token
If token is acronym **return** token
If token in dictionary **return** the stored stem
If token ends in s'
  strip the ' and **return** stripped token
If token ends in 's
  strip the 's and **return** stripped token
If token ends in "is", "us", or "ss" **return** token
If token ends in s
  strip s, check in dictionary, and **return** stripped token if there
If token ends with es
  strip es, check in dictionary, and **return** stripped token if there
If token ends in ies
  replace ies by y and **return** changed token
If token ends in s
  strip s and **return** stripped token
If token doesn't end with ed or ing **return** token
If token ends with ed
  strip ed, check in dictionary and **return** stripped token if there
If token ends in ied
  replace ied by y and **return** changed token
If token ends in eed
  remove d and **return** stripped token if in dictionary
If token ends with ing
  strip ing (if length > 5) and **return** stripped token if in dictionary
If token ends with ing and length ≤ 5 **return** token
// Now we have SS, the stripped stem, without ed or ing and it's
// not in the dictionary (otherwise algorithm would terminate)
If SS ends in doubling consonant
  strip final consonant and **return** the changed SS if in dictionary
If doubling consonant was l **return** original SS
If no doubled consonants in SS
  add e and **return** changed SS if in dictionary
If SS ends in c or z, or there is a g or l before the final doubling consonant
  add e and **return** changed SS
If SS ends in any consonant that is preceded by a single vowel
  add e and **return** changed SS
return SS

Part-Of-Speech (POS) Tagging

- Annotate each word in a sentence with a part-of-speech tag

I ate the spaghetti with meatballs.
Pro V Det N Prep N

John saw the saw and decided to take it to the table.
PN V Det N Con V Part V Pro Prep Det N

- Useful for subsequent syntactic parsing and word sense disambiguation

Part Of Speech (POS) Tagging

POS labels were defined by the Penn Treebank project

Alphabetical list of part-of-speech tags used in the Penn Treebank Project:

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun

Phrase Chunking

- Find all non-recursive noun phrases (NPs), verb phrases (VPs), and Prepositional phrases in a sentence.
  - [NP I] [VP ate] [NP the spaghetti] [PP with] [NP meatballs].
  - [NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September]

38

Named Entity Recognition (NER)

A Named Entity denotes

The most common provided by several tools

- named (PERSON, LOCATION, ORGANIZATION, MISC),
- numerical (MONEY, NUMBER, ORDINAL, PERCENT), and
- temporal (DATE, TIME, DURATION, SET) entities.

- Named entities are recognized using supervised machine learning algorithms
- Numerical entities are recognized using a rule-based system.
- Numerical entities that require normalization, e.g., dates.
- It can also be defined by Regex (regular expressions)

39

Gazeteers – NER by lists

- A gazetteer consists in a set of lists containing names of entities such as **cities**, **organizations**, **days of the week**, etc.
- These lists are used to find occurrences of these names in text, e.g. for the NER task .
- The word 'gazetteer' is often used interchangeably for both the set of entity lists and for the processing resource that makes use of those lists

40

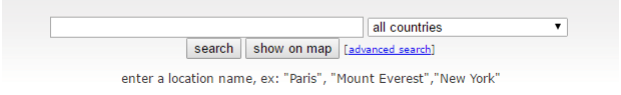
Gazeteers - Examples

Below is a section of the list for units of currency:

```
Ecu
European Currency Units
FFr
Fr
German mark
German marks
New Taiwan dollar
New Taiwan dollars
NT dollar
NT dollars
Real
...
```

Usando a Web para construir gazetters.

The GeoNames geographical database covers all countries and contains over eleven million placenames that are available for download free of charge.



41

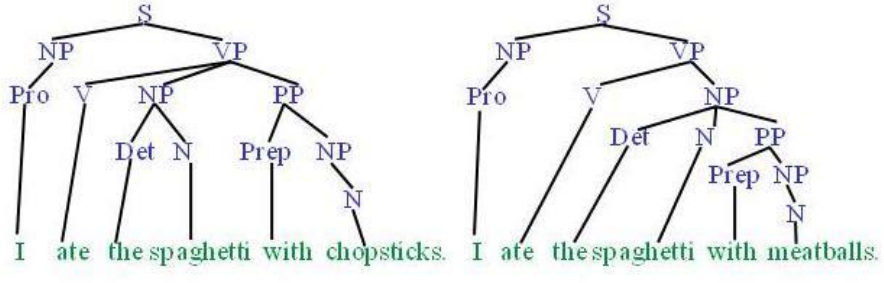
PARSING
1. Constituent

UFRPE Universidade Federal Rural de Pernambuco | DEINFO Departamento de Estatística e Informática

7-abr-18

Syntactic Parsing (Constituent)

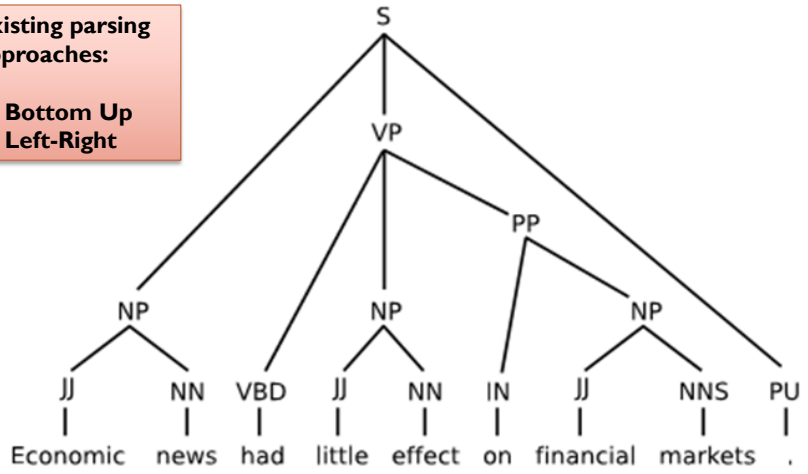
- It produces the most probable syntactic parse tree for a sentence
- It is usually based on a context-free grammar
- It organizes the words in a sentence in a hierarchical structured according to their internal phrases



Constituent Parsing Tree - Example

Existing parsing approaches:

- **Bottom Up**
- **Left-Right**



PARSING
2. Dependency

UFRPE Universidade Federal Rural de Pernambuco | DEINFO Departamento de Estatística e Informática

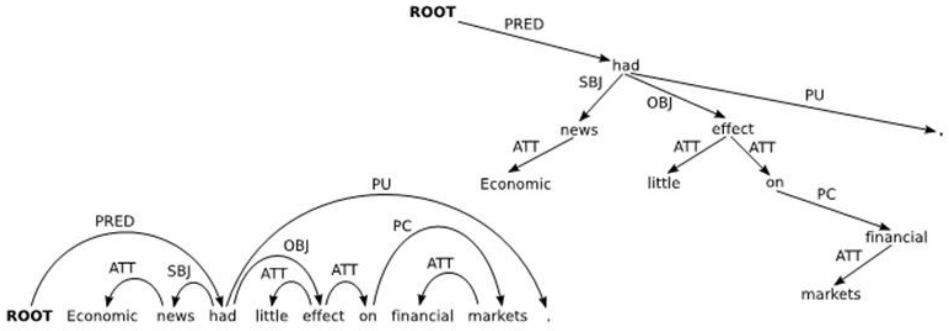
Dependency Parsing

Dependency Parsing is an approach based on the linguistic theory of **dependency grammar.**

The basic assumption:

> syntactic structure consists essentially of words connected by **asymmetric binary relations** of dependency relationships between them

Dependency Parsing Example



Generated Relations:

amod (news , Economic)
nsubj (had, news)
amod (effect , little)
dobj (had , effect)
amod (markets , financial)
prep_on (effect , markets)

The verb "had" has the subject "news"

Coreference Resolution (Anaphora)

- **Determine which phrases in a document refer to the same underlying entity.**
  - John put the carrot on the plate and ate it.
  - Bush started the war in Iraq. But the president needed the consent of Congress.
- **It is also called anaphora resolution**
- **CR types:**
  - Pronominal
  - nominal
- **Some cases require difficult reasoning**

48

Semantic Role Labeling

- Semantic Role Labeling (SRL) aims at automatically assigning semantic roles for each predicate in a sentence.
- SRL is also called **thematic role labeling**, or **shallow semantic parsing**.
- It determine which constituents in a sentence are semantic arguments for a given predicate, and then determining the appropriate role for each of those arguments.
- SRL has the potential to improve performance in any language understanding task
- The state-of-the-art approaches to SRL are based on supervised machine learning and Semantic Roles acquired from a corpus

49

Semantic Role Labeling with FrameNet

FrameNet is a computational resource that provides **Frame Semantics** for NLP

- The state-of-the-art approaches to SRL are based on supervised machine learning
- The central idea of **Frame Semantics** is that a word meanings must be described in relation to semantic frames (syntactic relations)
- FrameNet is a collection of facts that specify "characteristic features, attributes, and functions of a word, and its characteristic interactions with things necessarily or typically associate

Framenet Original Project (English)

https://framenet.icsi.berkeley.edu/fndrupal/

FrameNet Example

`Apply_heat`: Frame Elements

Cooker
Food
Temperature_setting
Duration
Heating_instrument
Medium

Lila **FRIED** the eggs in a copper pan.

Semantic Role Labeling (SRL): Example

- **For each clause, determine the semantic role played by each noun phrase that is an argument to the verb.**

agent patient source destination instrument

- John drove Mary from Austin to Dallas in his Toyota Prius.
- The hammer broke the window.

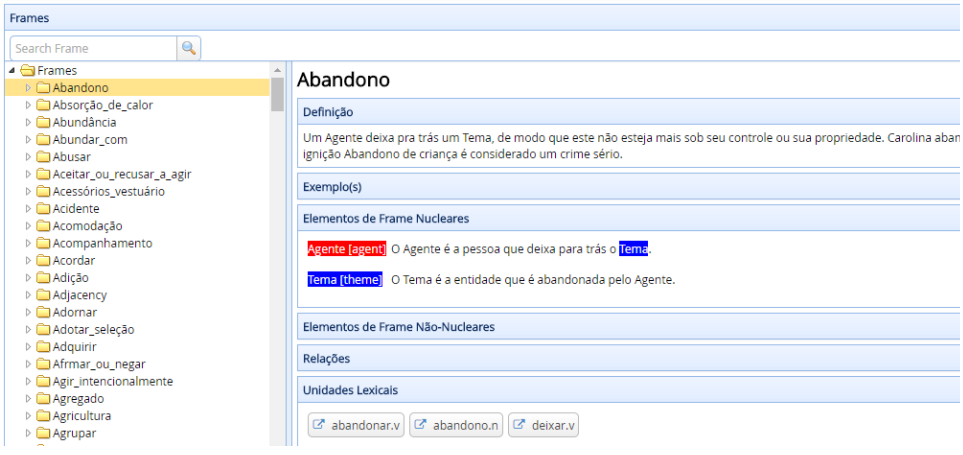
53

FrameNet Brazilian Portuguese

http://www.ufjf.br/framenetbr/

54

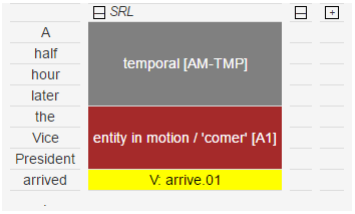
FrameNet Brazilian Portuguese: Example

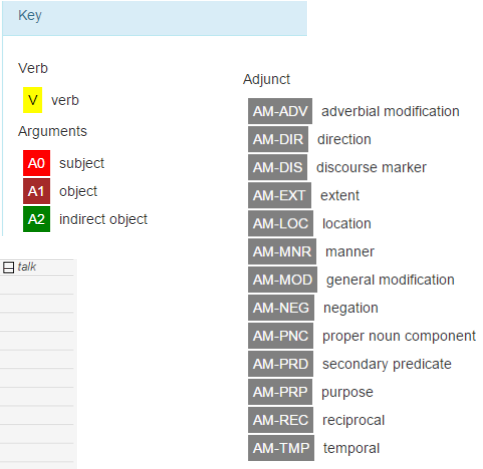


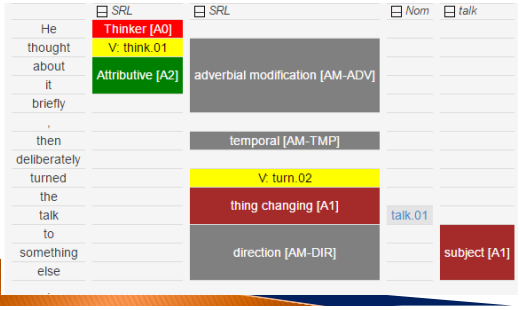
http://webtool.framenetbr.ufjf.br/index.php/fnbr/report/frame/main

55

SRL Example







NLP Tools in English

- **Stanford CoreNLP** demos
  - http://nlp.stanford.edu/software
- **GATE** (Java)
  - http://gate.ac.uk
- **OpenNLP**
  - http://opennlp.apache.org
- **ILLINOIS Curator** demos
  - http://cogcomp.cs.illinois.edu
- **LingPipe**
  - **http://alias-i.com/lingpipe-3.9.3/**
- **SEMAFOR** // demos
  - **http://www.cs.cmu.edu/~ark/SEMAFOR/**
- **FreeLing**
  - http://nlp.lsi.upc.edu/freeling/node/30

JAVA

PYTHON

- **NLTK** - http://www.nltk.org/

- **Spacy** https://spacy.io/docs/usage/word-vectors-similarities

- **CLIPS** http://www.clips.ua.ac.be/pages/MBSP

TextBlob - **https://textblob.readthedocs.io/en/dev/index.html**

CoreNLP

version 3.7.0

- Overview
- Usage
- Annotators
  - Summary
  - Annotator dependencies
  - Tokenization
  - Sentence Splitting
  - Lemmatization
  - Parts of Speech
  - Named Entity Recognition
  - RegexNER (Named Entity

Stanford CoreNLP – a suite of core NLP tools

Table of Contents

- About
- Download
- Human languages supported
- Programming languages and operating systems
- License
- Citing Stanford CoreNLP in papers

About

Stanford CoreNLP provides a set of natural language analysis tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and word dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between entity mentions, get quotes people said, etc.

NLP subtasks

- Tokenization
- Sentence Splitting
- Lemmatization
- Parts of Speech
- Named Entity Recognition
- RegexNER (Named Entity Recognition)
- Constituency Parsing
- Dependency Parsing
- Coreference Resolution

58

CoreNLP Pipelines

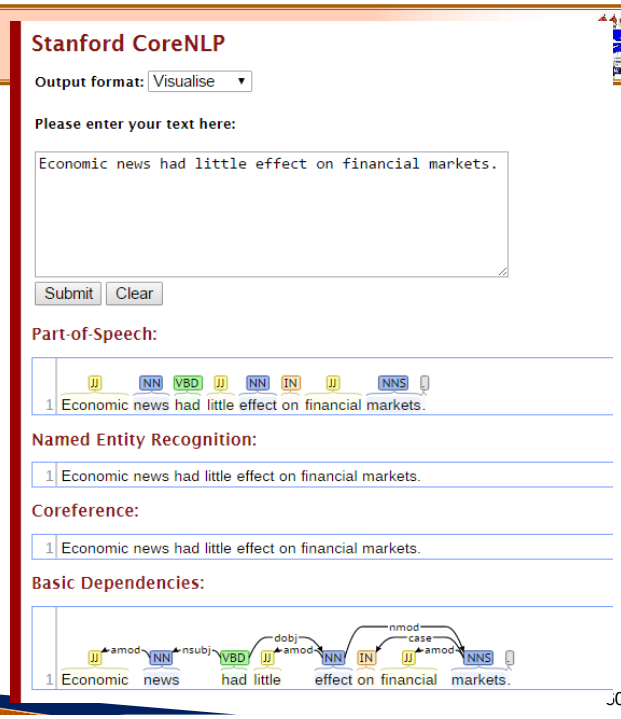
Annotator dependencies

PROPERTY NAME	ANNOTATOR CLASS NAME	REQUIREMENTS
tokenize	TokenizerAnnotator	None
cleanxml	CleanXmlAnnotator	`tokenize`
ssplit	WordsToSentenceAnnotator	`tokenize`
pos	POSTaggerAnnotator	`tokenize, ssplit`
lemma	MorphaAnnotator	`tokenize, ssplit, pos`
ner	NERClassifierCombiner	`tokenize, ssplit, pos, lemma`
regexner	RegexNERAnnotator	?
sentiment	SentimentAnnotator	?
parse	ParserAnnotator	`tokenize, ssplit`
depparse	DependencyParseAnnotator	`tokenize, ssplit, pos`
dcoref	DeterministicCorefAnnotator	`tokenize, ssplit, pos, lemma, ner, parse`

59

CoreNLP Tools

Demo



60

Applying NLP substasks in many TM applications

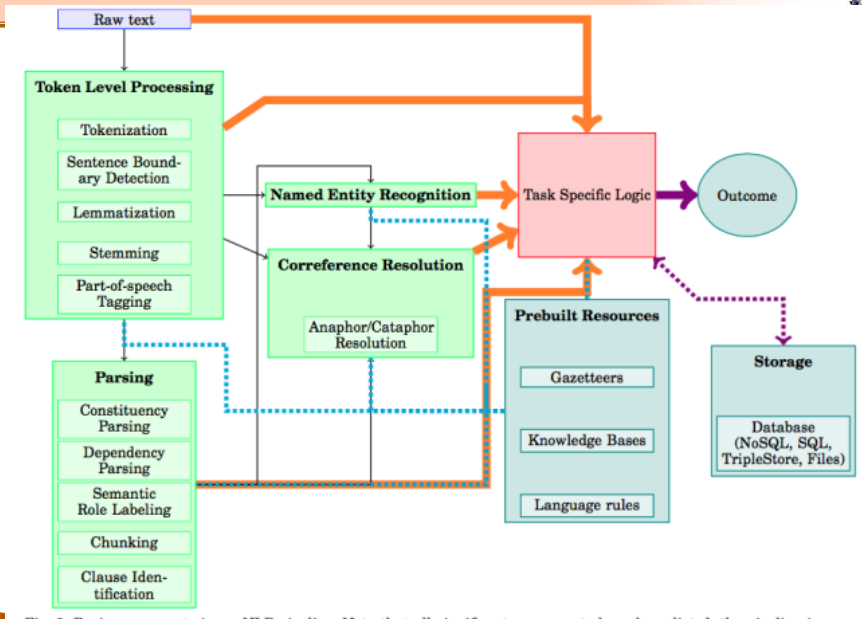


Fig. 2: Basic components in an NLP pipeline. Note that all significant components have been listed; the pipeline is constructed by customizing, adding, and/or, removing these components

61

Having Fun in the Weekend!

1ª. Lista de Exercícios