Approaches for Natural Language Processing (NLP)

Thierry Hamon

Institut Galilée - Université Paris 13, Villetaneuse, France
&
LIMSI-CNRS, Orsay, France
hamon@limsi.fr
http://perso.limsi.fr/hamon/

March 2014
ERASMUS Mobility - Mälardalen University (MDH) - Västerås - Sweden
Introduction

Plan

- Word and sentence segmentation
- Morphological analysis
- Parsing of texts
- Semantic analysis
Word and sentence segmentation
Text: a set of characters (string)

Segmentation of textual data: identification of a sublist of characters (substring) as a linguistic unit (word, sentence, phrase, term)

But it is (very) difficult to define precisely a linguistic unit

Linguistic unit identification: the first main step for the natural language processing

Several further tasks depend on its quality: content analysis, indexing, part-of-speech tagging, multilingual alignment, etc.
Why word and sentence segmentation?

- sentences: most of the grammars describe sentences
- words: basic information provided by dictionaries
  NB: Words can be simple (book) or complex/compound (French fries, spare time, one-way)

Identification of two types of units:

- units with regular structure (punctuation, number, date, bibliographical references, etc.
- Units requiring a morphological analysis
Comparative Analyses of Hairpin Substrate Recognition by Escherichia coli and Bacillus subtilis Ribonuclease P Ribozymes.

Ando T, Tanaka T, Kikuchi Y.

Previously, we reported that the substrate shape recognition of the Escherichia coli ribonuclease (RNase) P ribozyme depends on the concentration of magnesium ion in vitro. We additionally examined the Bacillus subtilis RNase P ribozyme and found that the B. subtilis enzyme also required high magnesium ion, above 10 mM, for cleavage of a hairpin substrate. The results of kinetic studies showed that the metal ion concentration affected both the catalysis and the affinity of the ribozymes toward a hairpin RNA substrate.

PMID: 12951523 [PubMed - in process]
Non trivial problem

- Problem of sentence definition: fussy notion of sentence
  - What are the marks of sentence boundaries? ( . ! ? . . . )
  - How to deal with title? Is a title a sentence?
  - A sentence can contain several several sentences if it includes
    - parenthetical clauses (examples between parenthesis or long-dash),
    - lists

- Problem of the word definition: minimal free form? entry of a dictionary?
  - How to deal with the ambiguity of the boundary characters?
  - How many words: doesn’t, French fries
    produktionsstyrningssystemsparvuuppdatering
Abstract This article presents a survey of techniques for ranking results in search engines, with emphasis on link-based ranking methods and the PageRank algorithm. The problem of selecting, in relation to a user search query, the most relevant documents from an unstructured source such as the WWW is discussed in detail. The need for extending classical information retrieval techniques such as boolean searching and vector space models with link-based ranking methods is demonstrated. The PageRank algorithm is introduced, and its numerical and spectral properties are discussed. The article concludes with an alternative means of computing PageRank, along with some example applications of this new method.

Keywords Citation ranking · PageRank · Search engines · Information retrieval · Text indexing · Ranking · Markov chains · Power method · Power series

1 Introduction

Nowadays, an ever-increasing amount of human knowledge is placed in computerized repositories such as the World Wide Web. This gives rise to the problem of how to locate specific
Various segmentation according to the objective

Specific processing according to the application

- content analysis: coarse-grained segmentation.
- Machine Translation: proper nouns, words, idioms
  \( \text{Bill Gates } \rightarrow \text{r"{a}kningen grindar } \)
- Information extraction: proper nouns with their semantic class, numbers, date, etc.
- Document Indexing: term recognition (single and multi-word phrases)
Segmentation strategies

- Punctuation: useful but unreliable
  Character full stop ( . ):
  - sentence boundary
  - used in the abbreviation
  - part of the number
  - mark of list item

- Solution: identification of (more) reliable parts of the text to eliminate ambiguities
  - proper names, abbreviations, compound and complex nouns
  - with
    - regular expressions
    - dictionaries (not exhaustive)
    - automatic recognition (named entity recogniser)
Identification of reliable segments

Objective: Removing the ambiguous punctuation marks
Identification textual segments with specific structures

1. Numbers (5.345)
2. Section/alphanumeric references (section 2.1, Doc1-3.2)
3. Date and time (10/31/2013, 10-31-2012/9:00)
4. Acronyms (AT&T, P.M.)
5. Ponctuations (...)
6. Abréviations (i.e., mg.)

→ Use of regular expressions or automata
Named entities

Proper nouns, abbreviations and acronyms
Important problem in NLP:

- Most of the unknown words in a text
- But convey very useful information
- Similar to complex words with a lot of variation
  (from wikipedia EN)
  Carl XVI Gustaf of Sweden
  Carl XVI Gustaf
  Carl Gustaf Folke Hubertus
  King Carl Gustaf
  His Majesty Carl XVI Gustaf, King of Sweden
  Carl Gustaf

- Acronyms can be similar to words: *NATO*, *Laser*, *Radar*
Named entities recognition

- Set of named entities is very large and opened, especially in technical texts
- Some tasks or application need a semantic tagging

Need of several analysis

- use of dictionaries or existing lists
- definition of automata
- statistical analysis and machine learning

For abbreviation:

- Regular expressions:
  \[A-Z]\.|[A-Z]\)+|[A-Z][:consonant:]+\)
- Analysis of variation in texts:
  - if a string/word is only in upper-case (NATO) or capitalised (Nato), it is considered as an abbreviation
  - if a string/word is also in lower-case, it can not be considered as an abbreviation
Exploitation of punctuation marks as sentence boundaries:

- full stop is very ambiguous: sentence boundary, part of abbreviation, and both!
- exclamation mark (non ambiguous)
- question mark (non ambiguous)
- semi-colon: used to delimitate list items and sentences

Identification of the status of punctuation marks by taking into account the context
Full stop and ellipsis (1)

They are not sentence boundaries if they are followed

- by comma or parenthesis
- lower-case word

if they are inside

- abbreviations
- numbers
Full stop and ellipsis (2)

Sentence boundaries if

1. they follow a closed parenthesis
2. they are before a capitalised word (except if it’s a proper noun of abbreviation)
3. they are followed by a line break \n
Comment:

- very general rules
- named entities (gene names – siRNA) or abbreviation can start with lower-case
Conclusion

Word and sentence segmentation takes into account several parameters:

- **Text Structure** (HTML markups, page setting)
- Need of dictionaries, it not sufficient
- Named entity recognition
- Manual check of the results can be needed
- Word and sentence segmentation are difficult to separate
- Unsolved/unsolvable problems:
  - governor’s (one word in the Brown corpus, two words in the Suzanne corpus)
  - it’s, who’s, don’t ...
  - disjoint units: *ne ... pas* (negative mark in French)
Morphological analysis
Applications using the morphologie

- Spell checking
- Hyphenation
- Analysis of the unknown words
- Reaccentuation
- Information and Document retrieval
- Language generation
- Text lemmatisation
- Part-of-speech tagging
Starting point: the words

Not the best (problem with definition) ...

... but not the worst

A choice coming from the written language

NB: avoid complex analysis and combinatorial explosion

- not take into account the compound tense (past perfect *have gone*, passive *voici has been taken*)
- not take into account the complex units/nouns which require a syntactic analysis
Morphological analysis of a text (1)

Corpus/Raw Text

\[ \text{Segmentation (words, expressions)} \]

Corpus/Segmentised text

\[ \text{POS tagging} \]

Corpus/Tagged Text

\[ \text{Morphological analysis (inflection, derivation, stemming)} \]

Corpus/Lemmatised Text

\[ \text{POS tagging} \]

Corpus/Tagged and Lemmatised Text
Two points of view:

- Identification of
  - compound words (*French fries*)
  - inflected forms (*I work, he works*)
  - derivational forms (*cell, cellular, medical, medicine*)

- Word description (*I worked*):
  - stem (*work*) (for *medical* → *medic*)
  - lemma (lexicon entry) (*to work*) (for *medical* → *medical*)
  - part-of-speech (*verb*)
  - morphological features (*1st person, singular, simple past, indicative voice*)
Morphological analysis of the words

- Inflection analysis
- Derivation analysis
- Idea for problem solving:
  - Syntactic ambiguity
  - Words with several part-of-speech categories
The more well-known processing, since the sixties

Methods:

- Use of lexicon of stems and lemma
- Splitting of the word to get a combination of the possible morphemes (small meaningful string)
- Use of analysis rules
Processing of the inflectional forms (2)

Current approaches:

- Exploitation of dictionaries of inflected forms, generated by applying inflectional rules on dictionaries of lemmatised forms

Examples of available resources:

- CELEX (English, Dutch, German)
- MULTEX (several European languages)
### Processing of the inflectional forms (3)

Samples of resources:

**CELEX (English):**

<table>
<thead>
<tr>
<th>Id</th>
<th>Lemma</th>
<th>Frequency</th>
<th>inflectional</th>
<th>Class</th>
<th>Compound?</th>
</tr>
</thead>
<tbody>
<tr>
<td>3357</td>
<td>BBC</td>
<td>491</td>
<td>14</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>3359</td>
<td>be</td>
<td>687085</td>
<td>4</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>3360</td>
<td>beach</td>
<td>1449</td>
<td>1</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>3361</td>
<td>beach</td>
<td>16</td>
<td>4</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>3362</td>
<td>beach ball</td>
<td>0</td>
<td>1</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

**CELEX (German):**

<table>
<thead>
<tr>
<th>Id</th>
<th>Lemma</th>
<th>Frequency</th>
<th>inflectional</th>
<th>Class</th>
<th>Compound?</th>
</tr>
</thead>
<tbody>
<tr>
<td>14508</td>
<td>gehen</td>
<td>7302</td>
<td>4</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>23459</td>
<td>Lufthafen</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>23478</td>
<td>Luftschiffahrt</td>
<td>0</td>
<td>2</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>48193</td>
<td>Wasserball</td>
<td>12</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>
Examples of approaches for the inflectional analysis

- Concatenative morphology
- Word-based morphology
- Two-level morphology
Concatenative morphology

Inflectional morphological analysis

Based on finite state automata
Two level morphology

(K. Koskenniemi, Helsinki)

Translation of a descriptive form to a real form
Based on trasnductors
Example:

\[ \text{book} \, +\text{pl} \rightarrow \emptyset \, s \]
\[ \text{bil} : \, +\text{undef} \, +\text{pl} \rightarrow \text{bil} \, \emptyset \, \text{ar} \]
\[ \text{livre} : \, +\text{masc} \, +\text{pl} \rightarrow \text{livre} \, \emptyset \, s \]
Word-based morphology (1)

*Ritchie et al.*

Based on context-free rules:
- Word → Verb
- Word → Noun
- Verb → Verbal-Prefix Verb
- Noun → Noun Nominal-Suffix

Use of features:

\{(V, +), (N, -), (PLU, +)\}
Example of structure/rule:

\[
\begin{array}{c}
\text{[BAR 0, V+ N-, SUBCAT NP]} \\
\text{[BAR 0, V+, N+, SUBCAT NULL]} \\
\text{[BAR 0, V+, N-, SUBCAT NP]} \\
\text{regular} \\
\text{\textasciitilde iz} \\
\end{array}
\]
Morphological analysis

Processing and analysis of the derivation (1)

- Based on dictionary:
  - Use of derivational information (derivational set of words): CELEX database

- Rule based approach: Stemming and suffix removing
  
  explanation $\rightarrow$ explain

  Allomorphy rule: -ain $\rightarrow$ an-

  Suffix rule: -tion $\rightarrow$ -iton, -ion, -ation

  inactivation ($\rightarrow$ act)

  Prefix rule: in activation]]+

  Suffix rules: [[[[ act ] ive ] ate ] tion ]
Sample of CELEX:

1182\alternate\V\+-+

1184\alternately\ADV\-+ly 1186\alternation\N\-e+ion
1187\alternative\N\+- 1188\alternative\A\-ion+ive
1189\alternatively\ADV\-+ly 1190\alternator\N\-e+or
1183\alternate\A\+-
Processing of the morphological ambiguities

Homographs

*morphological (and morphosyntactical) ambiguities can be undecidable with a “simple” morphological analysis*

book / to book
fö r (verb, noun, adverb, preposition)

Method: Exploitation of the word context (local syntactic information)

Solutions (according to the whole approach and architecture):

- **No decision** the ambiguity is just indicated
  - Ambiguous word is indicated as belong to several class
- **Decision** Use of statistical disambiguation rules (based on n-grams, a (shallow) syntactic analysis)
Stemming

- Identification of the smallest meaningful string
- Very used in information retrieval

Two main rule-based approaches:
- Separately, suffix removing then normalisation (Lovins, 1968)
- Simultaneously, suffix removing and normalisation (Porter, 1980)

Also, stemming based on appearance in corpora
**Lovins stemming (1)**

*Separately, suffix removing then normalisation*

**Step 1: Identification of suffix (terminal string) by decreasing length:**

<table>
<thead>
<tr>
<th>11</th>
<th>10</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>-alistically</td>
<td>-antalness</td>
<td>-alically</td>
</tr>
<tr>
<td>-arizability</td>
<td>-arisations</td>
<td>-antaneous</td>
</tr>
<tr>
<td>-izationally</td>
<td>-arizations</td>
<td>-antiality</td>
</tr>
<tr>
<td></td>
<td>-entialness</td>
<td>-arisation</td>
</tr>
</tbody>
</table>
Lovins stemming (2)

Step 2: Normalisation of the ending, according to a specific order:

1. double-character deletion: bb-, dd-, gg-, ll-, mm-, nn-, pp-, rr-, dd-, tt-, ...
2. iev- → ief-
3. uct- → uc-
4. umpt- → um-
5. rpt- → rb-
6. ...

Thierry Hamon (LIMSI & Paris Nord)
Lovins stemming (3)

Examples of stems computed with the Lovins algorithm:

<table>
<thead>
<tr>
<th>Word</th>
<th>String after suffix removing</th>
<th>Stem</th>
</tr>
</thead>
<tbody>
<tr>
<td>magnesia</td>
<td>magnes</td>
<td>magnes</td>
</tr>
<tr>
<td>magnetometer</td>
<td>magnetometer</td>
<td>magnetometer</td>
</tr>
<tr>
<td>magnetometry</td>
<td>magnetometr</td>
<td>magnetometer</td>
</tr>
</tbody>
</table>
### Porter stemming (1)

Simultaneously, suffix removing and normalisation

Definition of an set of rules, applied in a specific order.

<table>
<thead>
<tr>
<th>Step</th>
<th>Rules</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>-SSES → -SS</td>
<td>careSSES → careSS</td>
</tr>
<tr>
<td></td>
<td>-IES → -I</td>
<td>ponIES → ponI</td>
</tr>
<tr>
<td></td>
<td>-SS → -SS</td>
<td>careSS → careSS</td>
</tr>
<tr>
<td>1c</td>
<td>-Y → -I</td>
<td>happY → happI</td>
</tr>
<tr>
<td></td>
<td>-EMENT → -</td>
<td>replacEMENT → replac</td>
</tr>
<tr>
<td></td>
<td>-MENT → -</td>
<td>adjustMENT → adjust</td>
</tr>
<tr>
<td>3</td>
<td>-ATIONAL → -ATE</td>
<td>relATIONAL → relATE</td>
</tr>
<tr>
<td></td>
<td>-TIONAL → -TION</td>
<td>condiTIONAL → condiTION</td>
</tr>
</tbody>
</table>
Porter stemming (2)

Examples of stems computed with the Porter algorithm:

<table>
<thead>
<tr>
<th>Initial string</th>
<th>Cutting</th>
<th>Stem</th>
</tr>
</thead>
<tbody>
<tr>
<td>acid</td>
<td>acid</td>
<td>acid</td>
</tr>
<tr>
<td>acid</td>
<td>acid</td>
<td>acid</td>
</tr>
<tr>
<td>acidic</td>
<td>acid+ic</td>
<td>acid</td>
</tr>
<tr>
<td>acidify</td>
<td>acidifi</td>
<td>acidifi</td>
</tr>
<tr>
<td>acidity</td>
<td>acid+ity</td>
<td>acid</td>
</tr>
<tr>
<td>acidulate</td>
<td>acidul+ate</td>
<td>acidul</td>
</tr>
<tr>
<td>acidulated</td>
<td>acidul+ated</td>
<td>acidul</td>
</tr>
<tr>
<td>acidulous</td>
<td>acidul+ous</td>
<td>acidul</td>
</tr>
</tbody>
</table>
Lemmatisation and part-of-speech tagging

Lemmatisation: Identification of the canonical form of a word given its inflection form

universities → university

Part-of-speech tagging: Association of a tag (referring morpho-syntactical description) to a word

Tag: morpho-syntactical information, i.e. a grammatical class (Noun), morphological features (gender, number, tense - neutrum, plural, present)

Several NLP tasks (syntactic parser, semantic analysis) and applications need such information
Part-of-speech (POS) tags

Noun: book, car / bok, bil
Verb: (to) eat, (to) program
Adverb: daily / dagligen
Adjective: great / vacker
proper Noun: UNIX, Kernighan
...

Thierry Hamon (LIMSI & Paris Nord)
POSTag set

No unified tag set
Depend on the tasks and tool definition

- Minimal set: about 16 postags
- up to: 190 postags

Brill: 50 postags
TreeTagger: 36 postags for English
Multext tag set: potentially about 200 postags

Examples of tag sets:

- Penn TreeBank tag set: JJ, NN, VBZ
- Multtext: Vmip1s--, Nc-p--, Sp+Da--p--d, A--mp--
Main approaches for POS tagging

- Stochastic and machine learning methods
- Rule based methods
- Hybrid methods
Rule-based approaches

Exploitation of resources

Comments:

- Manual definition of rule dictionary (time consuming process)
- Easy to localise and understand errors
- The rule base can be modified
  - But, carefully, rules can be contradictory
- Good precision
Identification of the tag of the word (for a given position in a text) according to:

- preceding tags (n-grams, usually bigrams)
- the probability of the tag for the given word

Use of Markov chains or HMM based methods (CRF for instance), maximal entropy classifier
Stochastic methods (2)

Conditional probabilities of tagging according to previously seen words

\[
W = \cdots w_{i-2} \; w_{i-1} \mid w_i \cdots \leftarrow \text{words}
\]
\[
T = \cdots t_{i-2} \; t_{i-1} \mid t_i \cdots \leftarrow \text{POS tags}
\]

\[
p(T \mid W) = \frac{p(T)p(W \mid T)}{p(W)}
\]

Simplified hypothesis

\[
p(T \mid W)p(W) = p(t_1)p(t_2 \mid t_1) \prod_{i=3}^{n} p(t_i \mid t_{i-1}, t_{i-2}) \prod_{i=1}^{n} p(w_i \mid t_i)
\]

probability of the transition \( p(t_i \mid t_{i-1}, t_{i-2}) \)

probability of the issue \( p(w_j \mid t_i) \)
Stochastic methods (2)

Comments:

- Requirement: learning examples (a already tagged corpus)
- Model depend on language but also the type of text (and the topic)
- Difficult to identify the origin of the errors
- Need of the computing of probabilities and bigrams (or n-grams) for all the terms of the corpus (Wall Street Journal corpus, Brown corpus)
- Good precision
  - up 97% for the general English
  - up 98% for the biomedical English (Genia Tagger)
Example of POS tagger

TreeTagger

(Institute for Computational Linguistics, University of Stuttgart)
Probably one of the most used tagger

- Probabilistic tagger using decision trees
- Defined for Several languages (different learning models): English, French, German, Italian, Russian
- Learning on a already tagged corpus (for English: WSJ)
- Numerous “rules” (between $10^3$ to $10^4$)
- Some (ambiguous/unkwn) words can be tagger previously (to the corpus tagging)
TreeTagger
(Example of output for English)

Nonalcoholic JJ nonalcoholic
steatohepatitis SYM steatohepatitis
( (
NASH NP Nash
)
) )
is VBZ be
a DT a
morbid JJ morbid
condition NN condition
highly RB highly
related VBN relate
to TO to
obesity NN obesity
. SENT .
Hybrid methods

Use of rules defined by machine learning approach (probabilistic rules)

Pros:

- Rules are not built manually
- Identification of useful information that can not be found by humans

Cons:

- Interpretation of some rules can be difficult to identify from the linguistic point of view
- Complex interaction between rules
Hybrid methods

Eric Brill tagger

- Learning of POS tagging rules on a tagged corpus (Brown and WSJ corpora)
- Learned rules can be applied on new corpora
- Transformation-based error-driven learning
  - Computed probabilities of the POS tag for a word
  - Analysis of its own errors
  - Correction of the rules according to the error analysis
Brill tagger
Overview of the method

1. Initial tagging
2. Computing of the correct transformation space (rules)
3. Evaluation function
4. Computing of the rank list of rules

Text without POS tags

Initial POS tagger

Tagged text current

Tagged text (gold standard)

Evaluator of the rules

Rules

Error-driven learning of the transformations
Brill tagger
Example of transformation rules

Rewriting rules

*Change a tag from NN (Noun) to VB (verb) if the previous tag is TO*

to/TO eat/NN
→ to/TO eat/VB
Brill CCGT
(Example of output for English)

Nonalcoholic NNP nonalcoholic
steatohepatitis NN steatohepatitis
( (
NASH NNP nash
) )
is VBZ be
a DT a
morbid JJ morbid
condition NN condition
highly RB highly
related VBN relate
to TO to
obesity NN obesity
. . .
Brill tagger

- Automatic acquisition of rules
- Results comparables to stochastic methods
- Smaller learning corpus
- Human readable rules which can be modified (manually)
- Smaller number of rules ($\sim 10^2$)
Examples of POS taggers

- **Brill tagger**  
  [mail.cst.dk/tools/index.php](http://mail.cst.dk/tools/index.php)  

- **Cognitive computation group, U Illinois**  
  [l2r.cs.uiuc.edu/~cogcomp/pos_demo.php](http://l2r.cs.uiuc.edu/~cogcomp/pos_demo.php)

- **TreeTagger**  
  [www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html](http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html)

- **Genia Tagger**  
  [www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/](http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/)

- **TnT tagger**  
  [http://www.coli.uni-saarland.de/~thorsten/tnt/](http://www.coli.uni-saarland.de/~thorsten/tnt/)

- **Multext**  
POS Taggers for Swedish

- HMM tagger:
  http://ufal.mff.cuni.cz/~hajic/tools/swedish/

- TnT Model: http://stp.lingfil.uu.se/~evafo/software/
  http://www.lingfil.uu.se/staff/beata_megyesi/?languageId=1

- Brill tuning
  - http://www.ling.gu.se/~lager/Mutbl/mutbl_lite.html
Syntax

Syntactic parsing of texts

Objectives: Associate to a string separated in linguistic units:
- A representation of structural groups between units (chunks - phrases)
- Functional relations between groups of units

Syntactic parsing have to:
- Segment the sentences in phrases (noun phrases, verbal phrases) - constituent analysis
- Identify and characterise (syntactic) relations - dependency analysis
Example of syntactic analysis

he books a ticket
(u₁, u₂, u₃, u₄)

G₁
sentence

G₂
nominal phrase

G₃
verbal phrase

G₄
nominal phrase

u₁
he

u₂
books

u₃
a

u₄
ticket
Use of syntactic analysis in NLP

- Improvement of the spell checking concerning syntactic errors (gender or number agreements for instance)
- A requirement to semantic-based processing and application (translation, terminology building, information retrieval, ...)
  Disambiguation of POS tags, sentence subject identification, etc.
- Text and speech generation

Syntactic analysis:

- Crucial for some applications but not need for others
- Time consuming (especially for dictionary and rule based parsing)
- Sometimes a shallow parsing is sufficient (terminology, indexing, information extraction and retrieval)
  (NP: *stenosis of the aorte*)
Syntactic parsing

- Definition of constituent groups
- Definition functional relations between groups
- Definition of dependency between constituents

⇒ Hierarchical analysis of the sentence
Definition of formal grammar:

- $V$: vocabulary
- $V_T$: terminal symbols
- $V_N$: nonterminal symbols
- $V_T \cap V_N = \emptyset$; $V = V_T \cup V_N$
- $S$: “axioms” of the grammar (sentences)
- $P$: rules
I see a man with a telescope.

**Syntactic parsing**

- **P** → **SN** **SV**
- **SN** → **PRO**
- **SV** → **V** **SN**
- **SN** → **DET** **N** **SP**
- **SP** → **PREP** **SN**
I see a man with a telescope.
Sentence constituents

The *sentence constituents* are chunks, based on elementary part-of-speech categories

- *main categories* (full words)
  - nouns, verbs, adjectives, adverbs
- *secondary categories* (grammatical words - or “empty words”) 
  - prepositions, interjections, etc.

Chunk: set of words gather around a **head** (main word of the chunk)
Other words of the chunk are called **dependents** or **modifiers**.
sentence constituents

phrase

*phrase*: basic element of a sentence

<table>
<thead>
<tr>
<th>Type de syntagme</th>
<th>Fonction</th>
<th>Exemple</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal phrase (SN)</td>
<td>sujet</td>
<td><em>Peter, a tree</em></td>
</tr>
<tr>
<td></td>
<td>object</td>
<td><em>the blue car</em></td>
</tr>
<tr>
<td>pronominal phrase</td>
<td>sujet</td>
<td><em>you, mine</em></td>
</tr>
<tr>
<td></td>
<td>object</td>
<td><em>I</em></td>
</tr>
<tr>
<td>verbal phrase (SV)</td>
<td>predicat</td>
<td><em>write, want to write</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>go to the station</em></td>
</tr>
</tbody>
</table>
sentence constituents

phrase

- verbal phrase (SV) : write, want to write, write a book
  - narrow definition : verb
  - wide definition (generative grammar) :
    verb and all its complements/arguments, verb = predicate
sentence constituents

phrase

A phrase can be:
- minimal: defined with compulsory POS categories
  \[ SN = \text{DER} + \text{Noun} : \text{the book, a cat, the university} \]
- (expanded): including optional modifiers:
  - \( SN : \text{the small white book, the most famous university of Sweden} \)
  - \( SV : \text{is used in the manufacture of plastics} \)

Representation of phrases and sentence as trees
Word order in the sentence

- Position of the constituent in a sentence
- Various canonical order according to the language
  - SVO (canonical order of the sentence in English, French, Swedish): *Th student reads a book.*
  - SOV (Japanese)
  - Free order: Russian

Several orders can co-exist (SOV in subordinate in German)
Word order in a sentence

Comments:

- Modification of the word order $\implies$ modification of the semantic of the sentence
  
  *Frankly, did he speak? Did he speak frankly?*

- Free order: prepositional and adverbial phrases (SPrep, SAdv)

- Constraint order: articles, pronouns, interrogative words (wh-)

- Passive voice: subject inversion (the book is read by Peter)
Specificity of syntactic parsing

Specificity of syntactic parsing regarding the syntax from a linguistic point of view

Theories of the syntax: description of the syntax at the linguistic level

But, from the natural language processing point of view:

- Few information is available (from morphological analysis, part-of-speech tagging and position in the sentence)
- Impossible to use tests (distributional criteria - substitution, move, transformation), includes an semantic interpretation and validity of the sentences
Specificity of syntactic parsing

Example of disambiguation tests:

*Time flies like an arrow*
- *Time goes fast like an arrow*
- *Who flies like an arrow?*
- *you time flies, like you time an arrow*
- *(you) time flies similar to an arrow*

*I’m going to sleep*
- *I’m falling asleep*
- *I’m leaving and I go to sleep*

It requires a lot of various knowledge (agreement rules, semantic information, context, etc.)
Specificity of syntactic parsing

- Analyse morpho-syntaxique
  Teacher strikes idle kids
  Ambiguity of words (*strikes*, *idle*)
  Time flies like an arrow, fruit flies like a banana

- Découpage des syntagmes
  - Local ambiguities:
    (analysis of the (stenosis of the aorte))
    ((analysis of the stenosis) of the aorte)
  - The scope of the adverbs, adjectives, quantifiers, négation
  - Prepositional phrases

- Function identification
  Missing constituent: *John asked Bill to eat the leftovers*
  Inverted order: *Blessed are the poor in spirit*
Syntactic parsing

- Penn Treebank
  http://www.cis.upenn.edu/~treebank/
- Memory-based Shallow parser (ILK)
  ilk.uvt.nl/cgi-bin/tstchunck/demo.pl
- LinkParser (Carnegie Mellon University)
- Stanford parser (Stanford University)
  http://nlp.stanford.edu/software/lex-parser.shtml
Syntactic principles

N. Chomsky [1956, 1957] :

- Formal description of the natural language
- Analogy with formal languages (recursivity, syntactic conformity, ...)
- Objective of the syntax: Generation of all the grammatically correct sentences and only those sentences
Syntax

Syntactic formalisms

- Formal Grammars
- Transformational grammars
- Current formalisms (HPSG, GPSG, LFG, TAG)
- Limits
Formal Grammars

*Based on rewriting rules*

The grammar is the finite and generic representation of a language

- **axiom**: the sentence \((S)\)
- **Nonterminal symbols**: the POS tags and phrase tags
- **terminal symbol**: lexicon (words)
- **rules**: for rewriting, derivation, and production
  \((u \rightarrow v)\)

Analysis of a sentence: Building its derivation tree
Example of derivation tree

```
SN
  DET NOUN NOUN
  the coronary angiography

SV
  V
  shows

SN
  DET ADJ NOUN
  a significative stenosis
```
Formal grammars
context free grammar

- High number of possibilities
- Generation of sentences which are not correct grammatically
  *eat the mouse the cat*
- Generation of sentences which are not correct grammatically
  *the car eats the mouse*
Transformational Grammars (1)

- Transformation: application of elementary operations (deletion, addition, movement, substitution)
- Objective: Modification the structure of the phrase and transformation in another structure
- Two linguistic schools:
  - Noam Chomsky school
  - Zellig Harris school
Transformational Grammars (2)

Noam Chomsky school

Gather sentences which are superficially different

*The main objective is to precise if a coronary disease is present*
*Precise whether a coronary disease is present is the main objective*

Separate sentences which are superficially similar

*The plane landed at Paris*
*The plane took off at Paris*

→ Definition of syntactic functions and relations thank to surface string but also parsing tree.

Few implementation (easier to use syntagmatic grammars)
Transformational Grammars (3)

Zellig Harris school

Transformation between surface strings
Objective: make a link between subgroup of sentences (kernel sentences) with others

Example of A V C sentences:
(A : antibody ; V : produce, form, synthetize ; C : cell)

lymphocytes have a role in the production of antibody
Antibodies are produced by the cell

Implementation of parsers from English (Sager 1981)
Harris work is more related to distributional analysis (semantic analysis thanks to word distribution)
Dependency grammars

- Identification of hierarchical (syntactic) relations between the words
- Reconsideration of the notions of constituent and tree of constituents
- Relation between words defined dependency relations (and a graph dependency)
- Order of the terminal words \(\neq\) word order in the sentence
- Addition of constraints on the dependencies

Example of dependency tree:

\[
[V \text{ are seen } [N [N \text{ Peter}] [\text{Coord et}] [N \text{ Mary}]]]
\]

Better adapted grammars for the natural language where words are in a more free order
Lexical Functional Grammars (LFG)

(Bresnam et Kaplan 1982)

Objective: Several representation of the sentence

Association of

- hierarchical representation, describing the phrase and word order: constituent structures (c-structures)
- representation with a feature structure, describing the grammatical relations: fonctionnal structures (f-structure)
Example of feature structure

\[
\begin{array}{c}
\text{subject} \\
\text{agreement 1} \\
\text{P} \\
1 \\
\text{gender} \\
\text{number} \\
\text{sing} \\
\text{SN} \\
1 \\
\text{cat} \\
\end{array}
\]

(1)
Feature structures (1)

- Set of pairs (feature, value)

\[
\begin{align*}
\text{gender} & : \text{m} \\
\text{number} & : \text{sing}
\end{align*}
\]

- Complex value (embedded structure)

\[
\begin{align*}
\text{cat} & : \text{SN} \\
\text{agreement} & : \text{gender} : \text{m} \\
\text{number} & : \text{sing}
\end{align*}
\]
Feature structures (2)

- Reentrant structure

\[
\begin{align*}
\text{subject} & \quad \begin{array}{c}
\text{cat} \\
\text{agreement}
\end{array} & \quad \begin{array}{c}
P \\
1
\end{array} & \quad \begin{array}{c}
\text{gender} \\
\text{number}
\end{array} & \quad \begin{array}{c}
m \\
\text{sing}
\end{array} & \quad \begin{array}{c}
\text{cat} \\
\text{agreement}
\end{array} & \quad \begin{array}{c}
\text{SN} \\
1
\end{array}
\end{align*}
\]
Subsomption

generalisation / specialisation

\[
\begin{bmatrix}
\text{cat} & \text{SV}
\end{bmatrix}
\] (5)

\[
\begin{bmatrix}
\text{cat} & \text{SV} \\
\text{agreement} & \begin{bmatrix}
\text{pers} & 3
\end{bmatrix}
\end{bmatrix}
\] (6)

(4) subsume (5)
(4) \(\sqsubseteq\) (5)
Unification

The less specific feature structure is subsumed by two feature structures

\[
\begin{align*}
(7) & \quad : \quad \begin{bmatrix}
\text{cat} & \text{SV} \\
\text{agreement} & \begin{bmatrix}
\text{pers} & 3
\end{bmatrix}
\end{bmatrix} \\
(8) & \quad : \quad \begin{bmatrix}
\text{cat} & \text{SV} \\
\text{agreement} & \begin{bmatrix}
\text{number} & \text{sing}
\end{bmatrix}
\end{bmatrix}
\end{align*}
\]

\[
\text{unification : } \begin{bmatrix}
\text{cat} & \text{SN} \\
\text{agreement} & \begin{bmatrix}
\text{pers} & 3 \\
\text{number} & \text{sing}
\end{bmatrix}
\end{bmatrix}
\]

\[
(7) \sqcup (8) = (9), \ (9) \sqsubseteq (7) \ \text{et} \ (9) \sqsubseteq (8)
\]
Excerpt of the lexicon

\[
\text{artery} : \begin{bmatrix}
\text{cat} & \text{SN} \\
\text{tête} & \text{agreement} & \begin{bmatrix}
\text{number} & \text{sing} \\
\text{pers} & 3
\end{bmatrix}
\end{bmatrix}
\]

\[
\text{shows} : \begin{bmatrix}
\text{cat} & \text{V} \\
\text{head} & \text{form} & \text{shows} & \begin{bmatrix}
\text{agreement} & \begin{bmatrix}
\text{number} & \text{sing} \\
\text{pers} & 3
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\]
Example of rules

\[ X_0 \rightarrow X_1 \; X_2 \]

\[
\begin{align*}
< X_0 \text{ cat } > &= P \\
< X_1 \text{ cat } > &= SN \\
< X_2 \text{ cat } > &= SV \\
< X_0 \text{ head } > &= < X_2 \text{ head } > \\
< X_0 \text{ head subject } > &= < X_1 \text{ head } >
\end{align*}
\]
Head-driven Phrase Structure Grammars (HPSG)

(Pollard et Sag 1987)

- Complete linguistic theory: phonology, lexicon, syntax, semantics, pragmatics
- Focus on the sub-categorisation
- Introduction of the linguistic head (X-Bar theory)
- Formalism: Typed feature structures (TFS)
  Hierarchy of type (allowing the definition of sub types)
- HPSG elements:
  - lexical entries based on sub-categorisation
  - lexical rules for the derivation of new entries (for instance, the passive voice)
  - rules for the building of constituents
  - definition of correct building
Categorial Grammars (CG)

(Steedman 1998)

- Theory more based linguistic principles
- Not centred on constituents
  - Incompleted processing of the coordinantion
- Based on Lambek calcul (1958) – non-commutative logic
- Formalism: $\lambda$ – *calcul*
- Motivated by the compositionality principle
- Capacity of the CFG by adding supplementary operators
- Feature structures can be added to non-terminal elements
Syntax

Tree Adjoining Grammars (TAG)

(Joshi 1975)

- Formalism: based on the rewriting of trees with operation of substitution and adjunction
- Principle: For each elementary tree
  - at least one lexical head
  - one node for each argument categorised by the head
  - semantic association
  - une unique semantic unit
Limits

Formal systems based on a small set of operators
But

- require language description through the definition of dictionaries of lexicon and rules
- difficulties to analysis free texts (problems with the non canonical sentences)
- time consuming analysis

Less theoretical-based approaches:

- systems with flexible syntactic rules
- shallow parsing: focus on the constituent identification (without intending to analyse the structure of the constituents)
  - Xerox Shallow parser
  - In terminology building: focus on the identification and the parsing of the noun phrases
Strategies for processing local ambiguities (1)

- Back tracking: choice of a solution and the other solutions are kept in memory.
  If the rest of the analysis leads to reconsidering the choice, the parser comes back on the initial choice.
- Parallelism: considering all the possible choices
- Lockahead: the choice is done according to elements/information which follow the ambiguous element
Strategies for processing local ambiguities (2)

- **charts** approach: realisation of partial syntactic analysis and memorisation

- Specific algorithms: Avoid strong parsing problems or intrusive solutions
  
  For instance, two steps:
  
  - identification and analysis of minimal phrases
  - the processing of complex difficulties

Operational constraints (time processing, solution space) leads to reconsider theoretical aspects
Authors: John Lafferty, Daniel Sleator et Davy Temperley

- Identification dependency between words in English text
- Based on dependency grammar (with specific modification due to the implementation) with unification principle
- No decision: Give several syntactic parsing of the sentences
- Time processing could be very long according to the sentence complexity (several hours)
- Dictionary-based parser:
  - Part-of-speech tagging of the words before parsing (coarse-grained POS tags)
  - Parsing rules are defined in a specific dictionary
- Used in AbiWord (free word processor) for grammar checking
Semantic analysis

Introduction

Meaning representation for computing and interpretation of a sentence
Relation between the sign and the real word
A train at the Copenhagen station:

Semantics is (probably) everywhere in the NLP
More or less deep analysis
Application:

- information extraction
- Human machine dialogue
- Linguistic processing: Anaphora (reference identification)
- Resource definition and building
- Word sens disambiguation
Various semantics

- Computational semantics: meaning representation and inference
- Lexical semantics: more dedicated to the word denotation
  - Meaning-text theory
  - Generative lexicon
Computational semantics

Understand the meaning of a sentence

- Computing the meaning of a sentence
- Identifying its truth conditions
- Example: The train is at the Copenhagen station

General method: principle of compositional meaning
The meaning of the sentence is constructed from
- the meaning of the words and
- the syntactic construction (to which semantic rules are associated)
Computing the meaning

- First Order Logic (FOL)
  
  *Mary loves every bird*
  
  \[ \forall x (\text{bird}(x) \rightarrow \text{love}(m, x)) \]

- Each FOL expression is still a string
- Not very flexible even if most of the common sentence can be represented
- Fixed number and order of arguments

  - VP: loves Mary
    
    \[ \text{love}(?, m) \] (not in FOL) and \[ \text{love}(x, m) \] (free variable is not controlled)
    
    also
    
    \[ \forall x \text{love} x, m \] (everybody loves Mary) and \[ \exists \text{love}(m, x) \] (Mary loves everyone)
Computing the meaning

- Lambda Calcul (Grammar of Montague)
  - Abstract definition of the functions and its arguments (more flexible)
  - Add a new operator to bind free variables
    \( \lambda x \text{love}(x, \text{mary}) \) (VP: to love Mary)
    Abstraction of \( x \) to identify missing information
  - Instantiating free variable through the \( \beta \) reduction
    (replacement of the bound variable by the argument)
    \[ \lambda x \text{love}(x, m)(j) \Rightarrow \text{love}(j, m) \] (John loves Mary)
  - Arguments of the functions can be predicate and formula
    (allow the recursivity)

Implementation: Prolog or Lambda-Prolog
Computing the meaning

- Modal Logic
  - Progressive modification the initial content by linguistics element
  - dynamic semantics, discourse analysis, dialogue
    Example of use: weather forecast service (by phone)

- ...
Lexical semantics

Objective: Study and description of the meaning of the word in a natural language

- Use of various information: morphology, syntax, pronunciation
- Exploitation of lexicographic resources: dictionary, lexicon
  - to face the size of the lexicon (in French, 60,000 dictionary entries, with 4 or 5 meaning on average → 240,000 to 300,000 meanings)
  - to take into account idioms
  - And don’t forget proper nouns!

More adapted to the NLP on large volume of data than computational semantics
Applications

- Machine translation

- a tap leaks / a thief runs away
  same word for the verb: **fuire**
  Translation in French: *un robinet fuit / un voleur fuit*

- Information retrieval and extraction:
  increase of the recall: *(muscle) ache*/ *(muscle) pain*
  decrease error: *race (competition or taxonomic group)*

- Question-Answering system:
  Analysis of the question and identification of the topic
  *When was Einstein born?*

- ...
Approaches for the lexical semantics

- Ad-hoc models based on existing lexicon
- A low increase of specific resources (WordNet, FrameNet, VerbNet, GermanNet, …)
  difficulty to build a real semantic lexicon
- Is it possible (and useful) gather all the meaning of all the words of a language?
  We face to the problem of neologism (word creation)
- Several meaning:
  * The French president *(need of a reference)*
  * Even Peter is come *(implicite meaning)*
  * he catch a disease *(conceptual meaning)*
  ...

Thierry Hamon (LIMSI & Paris Nord)
How to record the meaning of a word?

- Description of the semantic relations with other words
- Association of subcategorisation schemes
- Association of thematic and semantic roles
- Representation of the meaning with *semes* (semantic primitives)
Semantic relations

Paradigmatic relations (substitution of words):

- Synonymy: words with the “same” meaning
- Hyperonymy: relation between a word and a more general one
- Meronymy: relation between a word designing a part and one designing the whole
- Antonymy: words with contrary meanings

Syntagmatic relations:

- Collocations: words appearing in the neighbourhood of target word
- Argument structure: expected arguments of word → subcategorisation scheme
  Selection restriction: semantic constraints on the arguments
Desambiguation

Remove the homonymy or the polysemy of the words

- Use of the context:
- Problem with the pun and rhetoric effect (advertisement, slogan)
Approaches

- Lexico-syntactic patterns
- Distributional analysis
- Machine-Learning (symbolic - ILP - and numerical (CRF, SVM, supervised and unsupervised)
Meaning-text theory

- Linguistic theory born in Russia in the sixties (Zholkovsky & Mel’čuk)
- Now developed by Igor Mel’čuk
- Objective: providing useful information to go from an idea (the meaning) to its realisation in a given language (the text)
- Lexicon: *Explanatory Combinatorial Dictionary*
Meaning-text theory

Lexicon: *Explanatory Combinatorial Dictionary*

- Central point of the theory (contrary to classical linguistic approach)
- Formal dictionary: rigorous decomposition of the meaning
  exhaustive description the combination of the syntax and the lexicon
- Include:
  - morphological, syntactic and semantic representation of the words
  - lexical functions:
  - Example: *magn* function
    - *magn*(rain) = heavy
    - *magn*(wind) = strong

- Published dictionaries for Russian and French
Comments

- A lot of lexicographic work (slow)
- A complex description (formalism, writing rules, ...)
- Variation according to the dictionary publication
- No full electronic version, difficulty of use in both manual and automatic processing
Generative lexicon

(Pustejovsky 1995)

- Focus on the polysemy (several meaning for the word)
- Avoid the enumeration of the meanings (difficulty to identify all the meaning of a word), and then the combinatory explosion
- Based on general meanings and used of the context to compute meanings
Generative lexicon

Principle

- Explanatory model for the polysemy (generative model)
- Model for articulating the syntax and the semantics
- Model for relating lexical meanings
- Model to define generalisation of the lexical units (behaviour)
- Model for taking into the context in the meaning computing
Lexical entry

4 levels for the word representation

- argumental structure: definition of the predicative structure of the word
- event structure: definition of the type of event (state, process, transition)
- qualia structure: properties (attribute) of the word
- structure for the lexical inheritance: organisation of the lexical concept in a network
Comments

- Processing of the regular/standard polysemy
- Few description at the lexical level
- Taking into account the generative process

But

- imprecise methodology for the definition of lexical entries
- scalability problem: difficulty to build a real generative lexicon
Semantics requires a lot of resources (lexicon, dictionary) providing meanings (semantic tags, semantic relations, semantic frame) 
The most well known:

- Wordnet
- Framenet

Also:

- Recycling machine-readable dictionary
- Building resources from the corpus
WordNet

- Lexical network of the general English words (nouns, verbs, adjectives, adverbs) – (150,000 words)
- Based on psycholinguistic work (started in 1985, first version in 1993)
- Free resource
- Meaning and lexicon organised
  - around the synsets (set of synonyms)
  - main relation: hyperonymy/hierarchical relations
- With semantic relations between the synsets (115,000)
WordNet

Exemple: man, adult male

- hyperonymes:
  - person, individual, someone, mortal, human, soul
  - living thing, life form, organism, being
  - entity
  - causal agent, cause, causal agency
  - entity

- hyponymes: blackman, bachelor, dandy, boyfriend...

- méronymes: adult male body
FrameNet

- Frame semantics (Fillmore, 1968): Understanding of new things by performing mental operation on already known things → Prototypical situation (frame)
  - Words referring the same frame called *Lexical Unit*
  - Definition of the frame with *Frame Element* (semantic roles)
  - Definition of hierarchical relations and inter-hierarchical relations at the frame level

**Example: Frame Apply Heat**

- Lexical Unit: *to cook, to bake, to grill*
- Frame Element: the *Cook, the Food, the Heating instrument*
- Example of sentence: 
  
  \[ \text{Cook } \text{The boys} \ \underline{\text{grill}} \ \text{Food } \text{their fishes} \ \text{Heating instr on a open fire} \]

- Relations:
  - More general frame: Activity, Intentionally_affect
  - Causative frame: Absorb_heat
  - Usage frame: Cooking_creation
FrameNet

Currently:
- more than 1,000 frames
- more than 10,000 Lexical Units (lemma associated to frames)
- more than 170,000 annotated sentences

Avoiding the syntactic and semantic combinatory
Also defined as description of a type of event or relation (with the participant)
Fillmore *et al.* 2003, https://framenet.icsi.berkeley.edu
Swedish Framenet http://spraakbanken.gu.se/eng/swefn
  - 51 frames
  - 2,300 lexical units (LUs)
Conclusion

- Segmentation and Morphology
  - Combination of resources and formalised rules
  - The most deeply studied and well defined steps of the NLP

- Syntaxe
  - Mainly based on mathematical, logic and theoretical approaches
  - Remaining time-consuming step

- Semantics
  - A growing step, thanks to the increase of processing power
  - Several proposed theories which required resources (to build)

- NLP requires a lot of linguistic and semantic resources

  *But existing resources can be hard to reuse in another context*
Alternative to the lack of resource describing specific information: corpus linguistics

- Linguistic information comes from the texts and the context of the words
- A growing use of machine learning approaches

Shallow parsing/processing: solution to the time-consuming approaches
The processing focus on the targeted (linguistic) elements in order to avoid time-consuming global processing

Specific processing in the specialised texts (technical, scientific documents)
Mainly based on the use of shallow processing at the syntactic and semantic levels