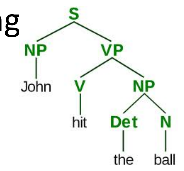
Toward NLP

Master Data Mining

Julien Velcin

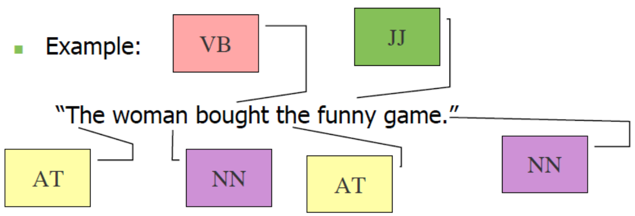


Shallow parsing

- Light parsing
- Identify the constituents (noun groups, verbs), but not the internal structure
- Two main steps:
  – Part of Speech tagging (POS)
  – Chunking

Tagging

- (PoS) tags: syntactic categories such as nouns, verbs and adjectives



Tagging

- (PoS) tags can be used for the identification of noun phrases etc.
  – Thematic categorization: focus is on noun terms
  – Sentiment categorization: adjectives

Part-of-speech (PoS) tagging

- By introducing part-of-speech tags we introduce word-types enabling to differentiate words functions
  - For text-analysis part-of-speech information is used mainly for "information extraction" where we are interested in e.g. named entities which are "noun phrases"
  - Another possible use is reduction of the vocabulary (features)
    - …it is known that nouns carry most of the information in text documents
- Part-of-Speech taggers are usually learned by probabilistic models (e.g., HMM, SVM) on **manually tagged** data; there exists also rule-based algorithms

PoS table (tagset)

part of speech	function or "job"	example words	example sentences
Verb	action or state	(to) be, have, do, like, work, sing, can, must	EnglishClub.com **is** a web site. I **like** EnglishClub.com.
Noun	thing or person	pen, dog, work, music, town, London, teacher, John	This is my **dog**. He lives in my **house**. We live in **London**.
Adjective	describes a noun	a/an, the, 69, some, good, big, red, well, interesting	My dog is **big**. I like **big** dogs.
Adverb	describes a verb, adjective or adverb	quickly, silently, well, badly, very, really	My dog eats **quickly**. When he is **very** hungry, he eats **really** quickly.
Pronoun	replaces a noun	I, you, he, she, some	Tara is Indian. **She** is beautiful.
Preposition	links a noun to another word	to, at, after, on, but	We went **to** school **on** Monday.
Conjunction	joins clauses or sentences or words	and, but, when	I like dogs **and** I like cats. I like cats **and** dogs. I like dogs **but** I don't like cats.
Interjection	short exclamation, sometimes inserted into a sentence	oh!, ouch!, hi!, well	**Ouch**! That hurts! **Hi**! How are you? **Well**, I don't know.

PoS examples

verb
Stop!

noun	verb
John	works.

noun	verb	verb
John	is	working.

pronoun	verb	noun
She	loves	animals.

noun	verb	adjective	noun
Animals	like	kind	people.

noun	verb	noun	adverb
Tara	speaks	English	well.

noun	verb	adjective	noun
Tara	speaks	good	English.

pronoun	verb	preposition	adjective	noun	adverb
She	ran	to	the	station	quickly.

pron.	verb	adj.	noun	conjunction	pron.	verb	pron.
She	likes	big	snakes	but	I	hate	them.

Here is a sentence that contains every part of speech:

interjection	pron.	conj.	adj.	noun	verb	prep.	noun	adverb
Well,	she	and	young	John	walk	to	school	slowly.

POS tagging algorithms

- POS-Tagging generally requires:
  - Training phase where a manually annotated corpus is processed by a machine learning algorithm;
  - Tagging algorithm that processes texts using learned parameters.
- Performance is generally good (around 96%) when staying in the same domain.

Illustration with TreeTagger

- H. Schmid, University of Stuttgart, Germany
- English: “Mary has a cat.”
  - Mary NP Mary
  - has VHZ have
  - a DT a
  - white JJ white
  - cat NN cat
  - . SENT .
- French: “Mary a un chat.”
  - Mary NAM Mary
  - a VER avoir
  - un DET un
  - chat NOM chat
  - . SENT .

Rule-based taggers

- Using a set of rules to do the tagging
- Alternative to the probabilistic models
- Advantages:
  - Reduction in stored information required
  - Small set of meaningful rules
  - Better portability to other tag set / languages
- For instance: the Brill tagger [Brill, 1992]

Brill tagger

- The dataset (e.g., the Brown corpus) is split into 3 sets:
  - 90% (first) training set
  - 5% (second) training set
  - 5% test set
- Assigns initially the most likely tags
- Uses 2 basic procedures to improve performance
- Acquires patches to take the context into account

Brill tagger (con’t)

- 2 basic procedures for previously unseen words:

  capitalized words -> proper nouns

  ended with the same 3 letters -> same POS

  e.g.: *blahblahous* -> adjective
- Acquiring patches (rules) using templates:

  Change tag **a** to tag **b** when:
  - The preceding (following) word is tagged z
  - The word two before (after) is tagged z
  - One of the two preceding (following) words is tagged z
  - The current word is (is not) capitalized **etc.**

Brill tagger (con’t)

- For each error triple <tag a, tag b, number> and patch, the error reduction is calculated
- The patch with the best improvement is applied
- For instance:

  VB NN PREV-1-OR-2-TAG AT

  <noun, verb, 159> -> <noun, verb, 79>

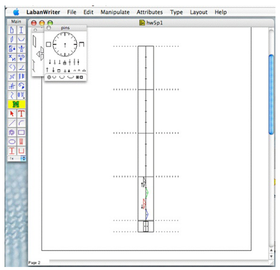
Brill tagger (con’t)

- 71 patches with the Brown corpus:
  – TO IN NEXT-TAG AT
  – VBN VBD PREV-WORD-IS-CAP YES
  – VBD VBN PREV-1-OR-2-OR-3-TAG HVD
  – VB NN PREV-1-OR-2-TAG AT

  etc.
- Performance comparable to probabilistic-based algorithmes, around 95%-97%

Tagging dance scores

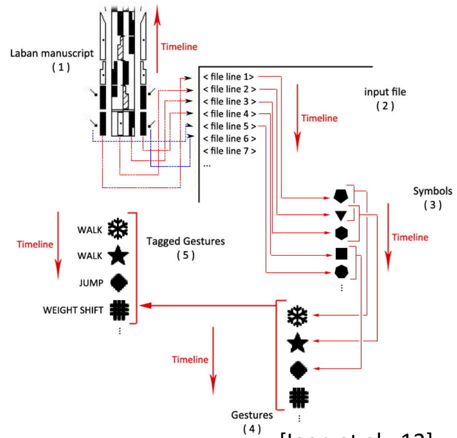
- Laban Notation: standardized graphical representation of choreography
- Used in many areas: dance, sport, physiotherapy etc.



```
LabanWriter
 4700
 no.no.no.no.no.no.no.no.no.no.no.no.no.no.no.
 1 792 612 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 10 0 0 0 0 0 0 6 Geneva ;
 2 184 27| 729 68 0 0 0 5 0 5 4 3 3 24 0 4 4 10 0 0 0 0 0 0 6 Geneva ;
 13 184 615 113 68 0 0 0 0 4 4 4 3 3 1 32 7 0 0.0 0 1 0 0 ;
 13 184 503 112 68 0 0 0 0 4 4 4 3 3 1 0 7 0 0.0 0 1 0 0 ;
 13 184 391 112 68 0 0 0 0 4 4 4 3 3 1 0 7 0 0.0 0 1 0 0 ;
 13 184 279 112 68 0 0 0 0 4 4 4 3 3 1 0 7 0 0.0 0 1 0 0 ;
 13 184 167 112 68 0 0 0 0 4 4 4 3 3 1 0 7 0 0.0 0 1 0 0 ;
 13 184 55 112 68 0 0 0 0 4 4 4 3 3 1 0 7 0 0.0 0 1 0 0 ;
 28 ;
```

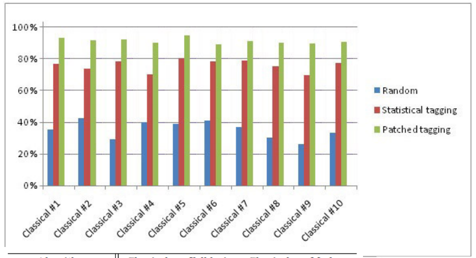

Tagging dance scores (con’t)

- Tagging with high-level (semantic) symbols: body actions (walk, run, jump, etc.), shape (arc-like, spoke-like, etc.), flow effort (bound or free) etc.
- Using these tags for describing the scores, some tasks are easier to deal with (e.g., genre classification).



[Ioan et al., 12]

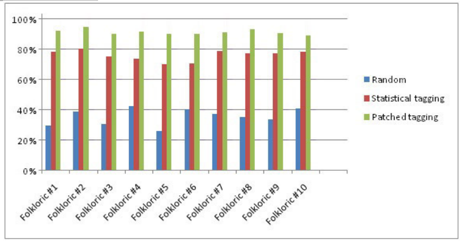
Tagging dance scores (3)



- Average accuracy:
  - Statistical tagging: 70 – 80%
  - Patched tagging: 89 – 94%
  - Classification: 98%

Algorithm	Classical vs. Folkloric	Classical vs. Modern
Naive Bayes	65.3%	51.0%
ID3	70.8%	56.4%
AdaBoost	72.0%	64.8%
K-Nearest Neighbor	59.7%	48.2%
C4.5	75.1%	66.3%
Random forest	68.8%	47.3%

Algorithm	Classical vs. Folkloric	Classical vs. Modern
Naive Bayes	98%	72%
ID3	97.3%	74%
AdaBoost	98.5%	75%
K-Nearest Neighbor	96%	70.2%
C4.5	99%	72.4%
Random forest	94.3%	69.5%



Chunking

- Finding Syntactic constituents like Noun Phrases (NPs) or Verb Groups (VGs) within a sentence
- **Less costly** than full parsing
- More **robust** to novel words, bad tokenization, wrong sentence split etc.
- Very useful in finding **named entities** (persons, companies, locations, patents…)

Chunking (an illustration)

- Based on rules of context-free grammar:

  e.g.: GN --> Det Adj N
- Examples of extract patterns:

  { Det="the", Adj="good", N="wife" }

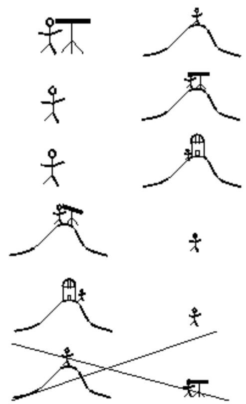
  { Det="a", Adj="broad", N="ship" }

  { Det="the", Adj="red", N="balloon" }

  etc.

Ambiguity

- Natural language is highly ambiguous and must be *disambiguated*.
  - I saw the man on the hill with a telescope.
  - I saw the Grand Canyon flying to LA.
  - Time flies like an arrow.
  - Horse flies like a sugar cube.
  - Time runners like a coach.
  - Time cars like a Porsche.



Meaning of a sentence

- Compare these 3 sentences [Chomsky]:
  - Colorless green ideas sleep furiously
  - Furiously sleep ideas green colorless
  - Ideas furiously colorless sleep green
- Languages have rules => constraints the way in which words can be combined into an **acceptable sentences**

Word Sense Disambiguation

- WSD problem: find out the most probable meaning
  - Supervised WSD (carried out with the help of a dictionary or a thesaurus)
  - Unsupervised WSD (the different senses of the word are not known).
- Consider the context (e.g., get the grammatical category of a word)

Lesk's algorithm

- Simple algorithm for WSD [Lesk, 86]
- Assumption:
  "Words in a given neighborhood will tend to share a common topic."
- For each word in a sentence:
  - look in a dictionary for the different definitions
  - look for the definitions of the close words
  - sense is chosen if it maximizes the common words

Illustration of Lesk

- Example with "pine cone"
- Definitions of "pine":
  - pine#1: "kinds of evergreen tree with needle-shaped leaves"
  - pine#2: "waste away through sorrow or illness"
- Definitions of "cone":
  - cone#1: "solid body which narrows to a point"
  - cone#2: "something of this shape whether solid or hollow"
  - cone#3: "fruit of certain evergreen trees"
- The best intersection is:
  - pine#1: "kinds of **evergreen tree** with needle-shaped leaves"
  - cone#3: "fruit of certain **evergreen trees**"

Simplified Lesk Algorithm

[Kilgarriff and Rosenzweig, 2000]

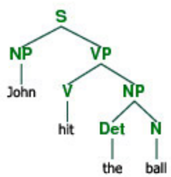
```
function SIMPLIFIED LESK(word,sentence)
returns best sense of word
  best-sense <- most frequent sense for word
  max-overlap <- 0
  context <- set of words in sentence
  for each sense in senses of word do
      signature <- set of words in the gloss and
                   examples of sense
      overlap <- COMPUTEOVERLAP(signature,context)
      if overlap > max-overlap then
         max-overlap <- overlap
         best-sense <- sense
end return (best-sense)
```

≈ 58% precision for Senseval-2 english

Limitations of Lesk-based methods

- Sensitive to the exact wording of definitions
  -> the absence of a word can **drastically** change the results
- Overlaps only among the glosses
  -> not sufficient vocabulary to fine-grained sense distinctions
- Task more difficult than PoS tagging
- Modern approaches for WSD:
  – Dictionary/knowledge-based (e.g., Lesk)
  – Supervised learning (e.g., ANN, SVM, CRF)
  – Semi-supervised learning (e.g., Yarowsky algorithm)
  – Unsupervised learning -> WSI (I=Inducion)

Full-parsing level



- Parsing provides maximum structura information per sentence
- On the input we get a sentence, on the output we generate a parse tree
- For most of the methods dealing with the text data the information in parse trees is too complex

Syntax

- Order of words in the query
  – the woman bought the funny game
  – the funny woman bought the game
- The parsing of a sentence could start
  – by the beginning or
  – by the end or even
  – by the main verb
- To go further -> NLP!