

# Artificial Intelligence EDAP01

## Lecture 8.1: Natural Language Processing

Pierre Nugues

Lund University

[Pierre.Nugues@cs.lth.se](mailto:Pierre.Nugues@cs.lth.se)

[http://cs.lth.se/pierre\\_nugues/](http://cs.lth.se/pierre_nugues/)

February 12, 2021



# Applications of Language Processing

- Spelling and grammatical checkers: *MS Word*
- Text indexing and information retrieval on the Internet: *Google, Microsoft Bing, Yahoo*
- Telephone information that understands some spoken questions
- Speech dictation of letters or reports
- Translation: *Google Translate, SYSTRAN*



# Applications of Language Processing (ctn'd)

- Direct translation from spoken English to spoken Swedish in a restricted domain: *SRI* and *SICS*
- Voice control of domestic devices
- Conversational agents able to dialogue and to plan
- Spoken navigation in virtual worlds: *Ulysse*, *Higgins*
- Generation of 3D scenes from text: *Carsim*
- Question answering systems: *IBM Watson*

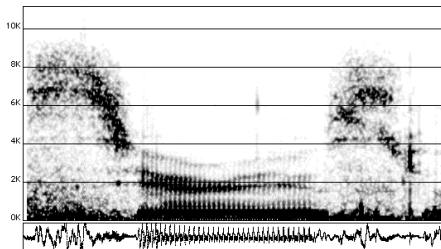


# Linguistics Layers

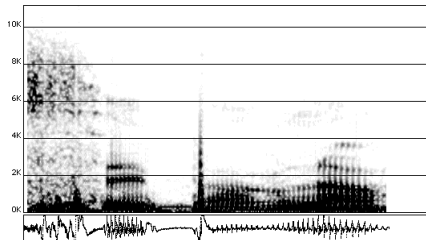
- Sounds
- Phonemes
- Words and morphology
- Syntax and functions
- Semantics
- Dialogue



# Sounds and Phonemes



*Serious*



*C'est par là 'It is that way'*



# Lexicon and Parts of Speech

*The big cat ate the gray mouse*

*The/article big/adjective cat/noun ate/verb the/article gray/adjective mouse/noun*

*Le/article gros/adjectif chat/nom mange/verbe la/article souris/nom grise/adjectif*

*Die/Artikel große/Adjektiv Katze/Substantiv ißt/Verb die/Artikel graue/Adjektiv Maus/Substantiv*

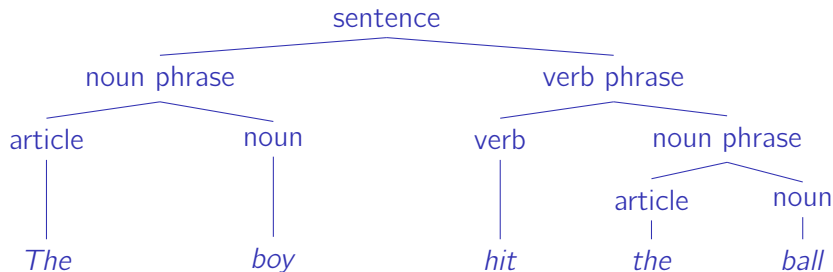


# Morphology

Word	Root form
<i>worked</i>	<i>to work</i> + verb + preterit
<i>travaillé</i>	<i>travailler</i> + verb + past participle
<i>gearbeitet</i>	<i>arbeiten</i> + verb + past participle



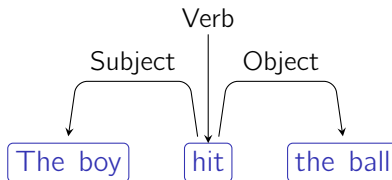
# Syntactic Tree





# Syntax: A Classical View

A graph of dependencies and functions



# Semantics

As opposed to syntax:

- 1 Colorless green ideas sleep furiously.
- 2 \*Furiously sleep ideas green colorless.

Determining the logical form:

Sentence	Logical representation
Frank is writing notes	writing(Frank, notes).
François écrit des notes	écrit(François, notes).
Franz schreibt Notizen	schreibt(Franz, Notizen).



# Lexical Semantics

Word senses:

- ① **note** (*noun*) short piece of writing;
- ② **note** (*noun*) a single sound at a particular level;
- ③ **note** (*noun*) a piece of paper money;
- ④ **note** (*verb*) to take notice of;
- ⑤ **note** (*noun*) of note: of importance.



# Reference

## 1. Sentence

*Pierre wrote notes*

## 2. Logical representation

`wrote(pierre, notes)`

## 3. Real world

Louis



Pierre

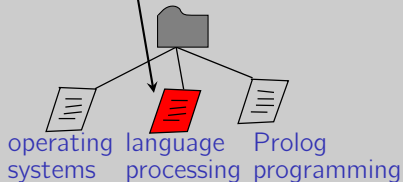


Charlotte



refers to

refers to



# Communication

Exchange of information between two parties

A dialogue is a set of linguistic interactions to carry out this exchange for instance to ask, inform, command, accept, etc.

It involves the generation of phrases/sentences by the speaker and their analysis by the hearer

Generation can be modeled as logical terms and then converted into sentences

Analysis involves the perception of the message, its syntactic and semantic parsing, and a pragmatic interpretation



# Ambiguity

Many analyses are ambiguous. It makes language processing difficult. Ambiguity occurs in any layer: speech recognition, part-of-speech tagging, parsing, etc.

Example of an ambiguous phonetic transcription:

*The boys eat the sandwiches*

That may correspond to:

*The boy seat the sandwiches; the boy seat this and which is; the buoys eat the sand which is*



# Models and Tools

Linguistics has produced an impressive set of theories and models

Language processing requires significant resources

Models and tools have matured. Resources are available.

Tools involve notably finite-state automata, regular expressions, rewriting rules, logic, statistics and machine learning.



# The Carsim System: A Text-to-Scene Converter

Texts

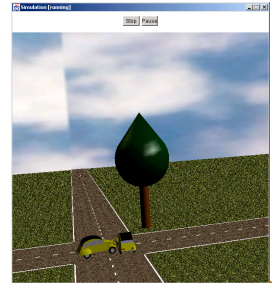
XML Templates

3D Animation

*Véhicule B venant de ma gauche, je me trouve dans le carrefour, à faible vitesse environ 40 km/h, quand le véhicule B, percute mon véhicule, et me refuse la priorité à droite. Le premier choc atteint mon aile arrière gauche,*

```
// Static Objects  
STATIC [  
ROAD  
TREE  
]
```

```
// Dynamic Objects  
DYNAMIC [  
VEHICLE [  
ID
```





# Corpora

A corpus is a collection of texts (written or spoken) or speech  
Corpora are balanced from different sources: news, novels, etc.

	English	French	German
<b>Most frequent words in a collection of contemporary running texts</b>	<i>the</i> <i>of</i> <i>to</i> <i>in</i> <i>and</i>	<i>de</i> <i>le</i> (article) <i>la</i> (article) <i>et</i> <i>les</i>	<i>der</i> <i>die</i> <i>und</i> <i>in</i> <i>des</i>
<b>Most frequent words in Genesis</b>	<i>and</i> <i>the</i> <i>of</i> <i>his</i> <i>he</i>	<i>et</i> <i>de</i> <i>la</i> <i>à</i> <i>il</i>	<i>und</i> <i>die</i> <i>der</i> <i>da</i> <i>er</i>



# Characteristics of Current Corpora

Big: The Bank of English (Collins and U Birmingham) has more than 500 million words

Available in many languages

Easy to collect: The web is the largest corpus ever built and within the reach of a mouse click

Parallel: same text in two languages: English/French (Canadian Hansards), European parliament (23 languages)

Annotated with part-of-speech or manually parsed (treebanks):

- Characteristics/N of/PREP Current/ADJ Corpora/N
- (NP (NP Characteristics) (PP of (NP Current Corpora)))



# Corpora as Knowledge Sources

Short term:

- Describe usage more accurately
- Assess tools: part-of-speech taggers, parsers.
- Learn statistical/machine learning models for speech recognition, taggers, parsers

Longer term:

- Semantic processing and knowledge extraction
- Texts are the main repository of human knowledge



# Counting Words and Word Sequences

Words have specific contexts of use.

Pairs of words like *strong* and *tea* or *powerful* and *computer* are not random associations.

Psychological linguistics tells us that it is difficult to make a difference between *writer* and *rider* without context

A listener will discard the improbable *rider of books* and prefer *writer of books*

A language model is the statistical estimate of a word sequence.

Originally developed for speech recognition

The language model component enables to predict the next word given a sequence of previous words: *the writer of books, novels, poetry etc* and not *the writer of hooks, nobles, poultry, ...*



# N-Grams

The types are the distinct words of a text while the tokens are all the words or symbols.

The phrases from *Nineteen Eighty-Four*

*War is peace*

*Freedom is slavery*

*Ignorance is strength*

have 9 tokens and 7 types.

Unigrams are single words

Bigrams are sequences of two words

Trigrams are sequences of three words



# Trigrams

Word	Rank	More likely alternatives
We	9	<i>The This One Two A Three Please In</i>
need	7	<i>are will the would also do</i>
to	1	
resolve	85	<i>have know do...</i>
all	9	<i>the this these problems...</i>
of	2	<i>the</i>
the	1	
important	657	<i>document question first...</i>
issues	14	<i>thing point to...</i>
within	74	<i>to of and in that...</i>
the	1	
next	2	<i>company</i>
two	5	<i>page exhibit meeting day</i>
days	5	<i>weeks years pages months</i>



# Probabilistic Models of a Word Sequence

$$\begin{aligned}P(S) &= P(w_1, \dots, w_n), \\&= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\dots P(w_n|w_1, \dots, w_{n-1}), \\&= \prod_{i=1}^n P(w_i|w_1, \dots, w_{i-1}).\end{aligned}$$

The probability  $P(\textit{It was a bright cold day in April})$  from *Nineteen Eighty-Four* corresponds to

*It* to begin the sentence, then *was* knowing that we have *It* before, then *a* knowing that we have *It was* before, and so on until the end of the sentence.

$$\begin{aligned}P(S) &= P(\textit{It}) \times P(\textit{was}|\textit{It}) \times P(\textit{a}|\textit{It, was}) \times P(\textit{bright}|\textit{It, was, a}) \times \\&\quad \times P(\textit{April}|\textit{It, was, a, bright, \dots, in}).\end{aligned}$$



# Approximations

Bigrams:

$$P(w_i|w_1, w_2, \dots, w_{i-1}) \approx P(w_i|w_{i-1}),$$

Trigrams:

$$P(w_i|w_1, w_2, \dots, w_{i-1}) \approx P(w_i|w_{i-2}, w_{i-1}).$$

Using a trigram language model,  $P(S)$  is approximated as:

$$P(S) \approx P(It) \times P(was|It) \times P(a|It, was) \times P(bright|was, a) \times \dots \\ \times P(April|day, in).$$





# Maximum Likelihood Estimate

Bigrams:

$$P_{\text{MLE}}(w_i | w_{i-1}) = \frac{C(w_{i-1}, w_i)}{\sum_w C(w_{i-1}, w)} = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}.$$

Trigrams:

$$P_{\text{MLE}}(w_i | w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})}.$$



# Text Categorization

The objective is to determine the type of a text with a set of predefined categories, for instance: {spam, no spam}

The Reuters corpus contains 800,00 economic newswires  
(<http://trec.nist.gov/data/reuters/reuters.html>)

Each newswire is manually annotated with a topic selected from a set of 103 predefined topics, for example:

C11: STRATEGY/PLANS,  
C12: LEGAL/JUDICIAL,  
C13: REGULATION/POLICY,  
C14: SHARE LISTINGS  
etc.



# Text Representation

Most categorizers use the **bag-of-word** technique that represents each document as a vector of words.

The vector parameters denote the presence or absence of a word.

The documents:

D1: Chrysler plans new investment in Latin America.

D2: Chrysler plans major investments in Mexico.

are represented as:

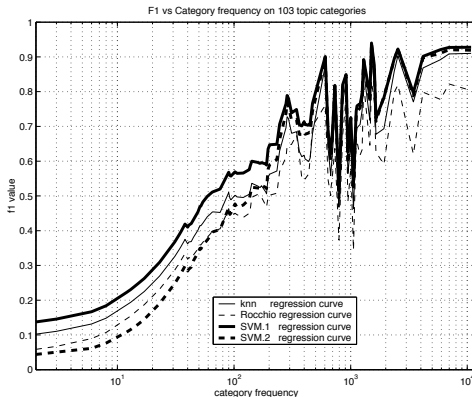
D\W	chrysler	plan	new	major	investment	latin	america	mexico
1	1	1	1	0	1	1	1	0
2	1	1	0	1	1	0	0	1

We can use supervised learning, where the classes are the categories and the features, the word vectors.



# Algorithms for Text Categorization

The performance depends on the number of samples



David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li, RCV1: A New Benchmark Collection for Text Categorization

Research, *Journal of Machine Learning Research* 5 (2004) 361-397.



# Information Retrieval

Astronomic number of available documents

Search engines – Google, Yahoo – are examples of tools to retrieve information on the web

Usually, we have:

- A document collection
- A query
- A result consisting of a set of documents

The simplest technique is to use a Boolean formula of conjunctions and disjunctions that will return the documents satisfying it.



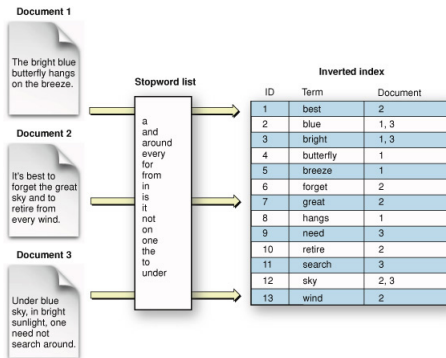
# The Vector Space Model

The vector space model represents a document in word space:

<b>Documents</b> <b>\ Words</b>	$w_1$	$w_2$	$w_3$	...	$w_m$
$D_1$	$C(w_1, D_1)$	$C(w_2, D_1)$	$C(w_3, D_1)$	...	$C(w_m, D_1)$
$D_2$	$C(w_1, D_2)$	$C(w_2, D_2)$	$C(w_3, D_2)$	...	$C(w_m, D_2)$
...					
$D_n$	$C(w_1, D_n)$	$C(w_2, D_n)$	$C(w_3, D_n)$	...	$C(w_m, D_n)$



# Inverted Index (Source Apple)



<http://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/SearchKitConcepts/index.html>  
Lucene is an outstanding program for document indexing and retrieval.  
<http://lucene.apache.org>



Word clouds give visual weights to words





# $TF \times IDF$

The frequency alone might be misleading

Document coordinates are in fact  $tf \times idf$ : Term frequency by inverted document frequency.

Term frequency  $tf_{i,j}$ : frequency of term  $j$  in document  $i$

Inverted document frequency:  $idf_j = \log\left(\frac{N}{n_j}\right)$



# Document Similarity

Documents are vectors where coordinates could be the count of each word:  $\vec{d} = (C(w_1), C(w_2), C(w_3), \dots, C(w_n))$

The similarity of documents is their cosine:

$$\cos(\vec{q}, \vec{d}) = \frac{\sum_{i=1}^n q_i d_i}{\sqrt{\sum_{i=1}^n q_i^2} \sqrt{\sum_{i=1}^n d_i^2}}.$$



# Evaluation

The Message Understanding Conferences have introduced a metric to evaluate the performance of information extraction systems using three figures.

They are borrowed from library science

	Relevant documents	Irrelevant documents
Retrieved	<i>A</i>	<i>B</i>
Not retrieved	<i>C</i>	<i>D</i>



# Recall, Precision, and the F-Measure

**Recall** measures how much relevant information the system has retrieved.

$$\text{Recall} = \frac{A}{A \cup C}.$$

**Precision** is the accuracy of what has been returned

$$\text{Precision} = \frac{A}{A \cup B}.$$

Recall and precision are combined into the **F-measure**, which is defined as the harmonic mean of both numbers:

$$F = \frac{2PR}{P+R}.$$



# Implementation Details

Very frequent words (stop words) can be removed

Words can be stemmed or lemmatized, for instance *table*, *tables*, *tabled*, *tabling* would have the same representation

Search can be extended to synonyms

Some systems use spell checkers



# Google's PageRank

Google's PageRank algorithm does not use word frequencies, but the page popularity through the “backlinks”, the links pointing to a page. Each backlink has a specific weight, which is related to the rank of the page it comes from.

The page rank is defined as the sum of the weights of all its backlinks:

$$PR(p_j) = \frac{1-d}{N} + d \sum_{p_i \in \text{Ref}(p_j)} \frac{PR(p_i)}{C(p_i)}.$$



## Google's PageRank (II)

The importance of a page is spread through its forward links and contributes to the popularity of the pages it points to.

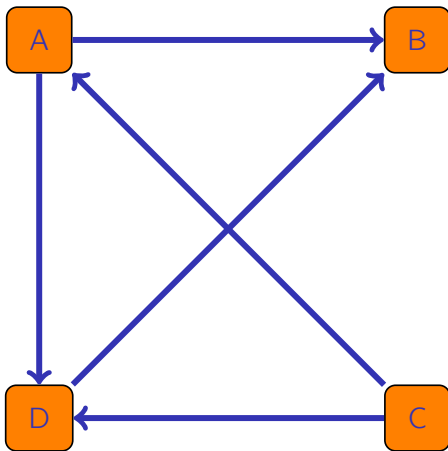
The weight of each of these forward links is the page rank divided by the count of the outgoing links.

The ranks are propagated in a document collection until they converge:

$$PR(p_j, t+1) = \frac{1-d}{N} + d \sum_{p_i \in \text{Ref}(p_j)} \frac{PR(p_i, t)}{C(p_i)}$$



# Pagerank Example



$$d = 0.85; \frac{1-d}{N} = \frac{0.15}{4} = 0.0375$$

A	B	C	D
1	1	1	1

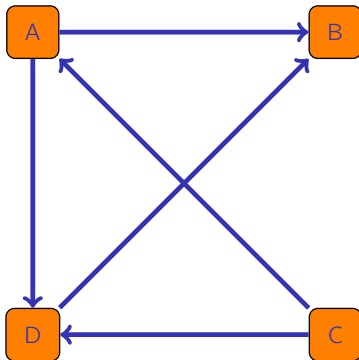
Table: Initial pageranks

→	A	B	C	D
A	0	1	0	1
B	0	0	0	0
C	1	0	0	0
D	0	1	0	0





# Pagerank Example



→	A	B	C	D
A	0	0.5	0	0.5
B	0	0	0	0
C	0.5	0	0	0.5
D	0	1	0	0
Sum	0.5	1.5	0	1

Table: Initial weighted links

	A	B	C	D
Sum	0.5	1.5	0	1
Damped = $0.85 \times \text{Sum}$	0.425	1.275	0	0.85
PR = $0.0375 + \text{Damped}$	0.4625	1.3125	0.0375	0.8875



# Message Understanding Conferences

The Message Understanding Conferences (MUCs) measure the performance of information extraction systems.

They are competitions organized by an agency of the US department of defense, the DARPA

The competitions have been held regularly until MUC-7 in 1997.

The performances improved dramatically in the beginning and stabilized then.

MUCs are divided into a set of tasks that have been changing over time.

The most basic task is to extract people and company names.

The most challenging one is referred to as information extraction



# Information Extraction

Information extraction consists of:

- The analysis of pieces of text ranging from one to two pages,
- The identification of entities or events of a specified type,
- The filling of a pre-defined template with relevant information from the text.

Information extraction then transforms free texts into tabulated information.



## An Example

*San Salvador, 19 Apr 89 (ACAN-EFE) – [TEXT] Salvadoran President-elect Alfredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of the crime...*

*Garcia Alvarado, 56, was killed when a bomb placed by urban guerrillas on his vehicle exploded as it came to a halt at an intersection in downtown San Salvador...*

*Vice President-elect Francisco Merino said that when the attorney general's car stopped at a light on a street in downtown San Salvador, an individual placed a bomb on the roof of the armored vehicle...*

*According to the police and Garcia Alvarado's driver, who escaped unscathed, the attorney general was traveling with two bodyguards. One of them was injured.*



# The Template

Template slots	Information extracted from the text
Incident: Date	19 Apr 89
Incident: Location	El Salvador: San Salvador (city)
Incident: Type	Bombing
Perpetrator: Individual ID	<i>urban guerrillas</i>
Perpetrator: Organization ID	<i>FMLN</i>
Perpetrator: Organization confidence	Suspected or accused by authorities: <i>FMLN</i>
Physical target: Description	<i>vehicle</i>
Physical target: Effect	Some damage: <i>vehicle</i>
Human target: Name	<i>Roberto Garcia Alvarado</i>
Human target: Description	Attorney general: <i>Roberto Garcia Alvarado</i> <i>driver</i> <i>bodyguards</i>
Human target: Effect	Death: <i>Roberto Garcia Alvarado</i> No injury: <i>driver</i>



# FASTUS

The FASTUS system has been designed at the Stanford Research Institute to extract information from free-running text

FASTUS uses partial parsers that are organized as a cascade of finite-state automata.

It includes a tokenizer, a multiword detector, and a group detector as first layers.

Verb groups are tagged with active, passive, gerund, and infinitive features.

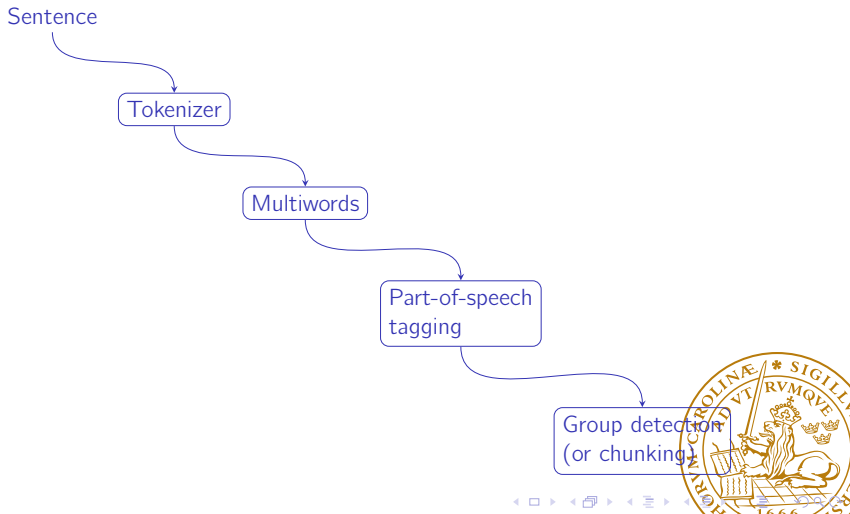
Then FASTUS combines some groups into more complex phrases and uses extraction patterns to fill the template slots.

See

<http://www.ai.sri.com/natural-language/projects/fastus.html>



# FASTUS' Architecture



# Probabilistic Models for Information Extraction

It is possible to use statistical tagging techniques to carry out information extraction.

An example with three tapes corresponding to the text (input), speaker, and date (both output).

(From the textbook, Stuart Russell and Peter Norvig, *Artificial Intelligence*, 3rd ed., 2010, page 876.)

There	will	be	a	seminar	by	Andrew	McCallum	on	Friday
–	–	–	–	PRE	PRE	TARGET	TARGET	POST	–
–	–	–	–	–	–	–	–	PRE	TARGET

The speaker and date tapes are tagged by two separate hidden Markov models.

The procedure is similar to that of part-of-speech tagging.





# Tagging Techniques to Extract Groups

Group detection – chunking – can be reframed as a tagging operation.

From:  $[_{NG}$  The government  $_{NG}$ ] has  $[_{NG}$  other agencies and instruments  $_{NG}$ ] for pursuing  $[_{NG}$  these other objectives  $_{NG}$ ] .

To: *The/I government/I has/0 other/I agencies/I and/I instruments/I for/0 pursuing/0 these/I other/I objectives/I ./0*

From: Even  $[_{NG}$  Mao Tse-tung  $_{NG}$ ]  $[_{NG}$  's China  $_{NG}$ ] began in  $[_{NG}$  1949  $_{NG}$ ] with  $[_{NG}$  a partnership  $_{NG}$ ] between  $[_{NG}$  the communists  $_{NG}$ ] and  $[_{NG}$  a number  $_{NG}$ ] of  $[_{NG}$  smaller, non-communists parties  $_{NG}$ ] .

To: *Even/0 Mao/I Tse-tung/I 's/B China/I began/0 in/0 1949/I with/0 a/I partnership/I between/0 the/I communists/I and/0 a/I number/I of/0 smaller/I ,/I*



# IOB Annotation for Named Entities

CoNLL 2002		CoNLL 2003			
Words	Named entities	Words	POS	Groups	Named entities
Wolff	B-PER	U.N.	NNP	I-NP	I-ORG
,	O	official	NN	I-NP	O
currently	O	Ekeus	NNP	I-NP	I-PER
a	O	heads	VBZ	I-VP	O
journalist	O	for	IN	I-PP	O
in	O	Baghdad	NNP	I-NP	I-LOC
Argentina	B-LOC	.	.	O	O
,	O				
played	O				
with	O				
Del	B-PER				
Bosque	I-PER				
in	O				
the	O				
final	O				
years	O				
of	O				
the	O				
seventies	O				
in	O				
Real	B-ORG				
Madrid	I-ORG				



# Multiple Categories of Chunks

Extendable to any type of chunks: nominal, verbal, etc.

For the IOB scheme, this means tags such as I.Type, O.Type, and B.Type, Types being NG, VG, PG, etc.

In CoNLL 2000, ten types of chunks

Word	POS	Group	Word	POS	Group
<i>He</i>	PRP	B-NP	<i>to</i>	TO	B-PP
<i>reckons</i>	VBZ	B-VP	<i>only</i>	RB	B-NP
<i>the</i>	DT	B-NP	<i>£</i>	#	I-NP
<i>current</i>	JJ	I-NP	<i>1.8</i>	CD	I-NP
<i>account</i>	NN	I-NP	<i>billion</i>	CD	I-NP
<i>deficit</i>	NN	I-NP	<i>in</i>	IN	B-PP
<i>will</i>	MD	B-VP	<i>September</i>	NNP	B-NP
<i>narrow</i>	VB	I-VP	<i>.</i>	.	O

Noun groups (NP) are in red and verb groups (VP) are in blue.



## Example from Kudoh and Matsumoto (2000)

Three lines or columns representing the words, the parts of speech, and the groups.

<i>He</i>	<i>reckons</i>	<i>the</i>	<i>current</i>	<i>account</i>	<i>deficit</i>	<i>will</i>	<i>narrow</i>
PRP	VBZ	DT	JJ	NN	NN	MD	VB
B-NP	B-VP	B-NP	I-NP	I-NP	I-NP	B-VP	I-VP

<i>to</i>	<i>only</i>	<i>#</i>	<i>1.8</i>	<i>billion</i>	<i>in</i>	<i>September</i>	<i>.</i>
TO	RB	#	CD	CD	IN	NNP	.
B-PP	B-NP	I-NP	I-NP	I-NP	B-PP	B-NP	O



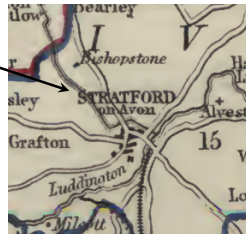
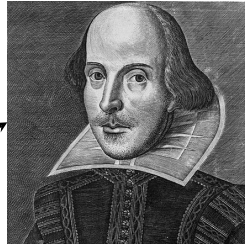
# Example from Kudoh and Matsumoto (2000)

Words	POS	Groups	
BOS	BOS	BOS	Padding
BOS	BOS	BOS	
He	PRP	B-NP	
reckons	VBZ	B-VP	
the	DT	B-NP	
current	JJ	I-NP	
account	NN	I-NP	
deficit	NN	I-NP	Input features
will	MD	B-VP	
narrow	VB	I-VP	Predicted tag
to	TO	B-PP	
only	RB	B-NP	↓
£	#	I-NP	
1.8	CD	I-NP	
billion	CD	I-NP	
in	IN	B-PP	
September	NNP	B-NP	
.	.	O	
EOS	EOS	EOS	Padding



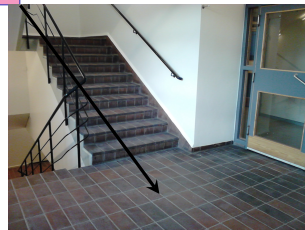
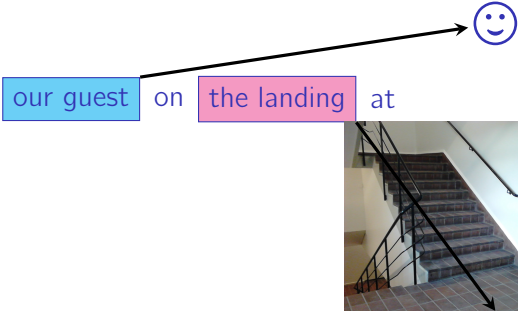
# Named Entities: Proper Nouns

William Shakespeare was born and brought  
up in Stratford-upon-Avon



## Others Entities: Common Nouns

Meeting with our guest on the landing at  
lunchtime



# Supervised Learning: A Summary

Needs a manually annotated corpus called the **gold standard**

The gold standard may contain errors (*errare humanum est*) that we ignore

A classifier is trained on a part of the corpus, the **training set**, and evaluated on another part, the **test set**, where automatic annotation is compared with the *gold standard*

**N-fold cross validation** is used avoid the influence of a particular division

Some algorithms may require additional optimization on a development set

Classifiers can use statistical or symbolic methods





# Conditional Random Fields

To carry out named entity tagging, time or location extraction, it is possible to use discriminative models such as support vector machines. Conditional random fields are tools that take into account sequences. Let  $X_1^N$  be the words and  $Y_1^N$ , the tags:

$$P(Y_1^N | X_1^N) = \alpha e^{\sum_{i=1}^N F(Y_{i-1}, Y_i, X_1^N, i)}$$

$F$  are feature functions:

$$F(Y_{i-1}, Y_i, X_1^N, i) = \sum_k \lambda_k f_k(Y_{i-1}, Y_i, X_1^N, i)$$

where:

$$f_1(Y_{i-1}, Y_i, X_1^N, i) = \begin{cases} 1 & Y_i = \text{SPEAKER and } X_i = \text{Andrew} \\ 0 & \text{otherwise.} \end{cases}$$

(From the textbook, Stuart Russell and Peter Norvig, *Artificial Intelligence*, 3rd ed., 2010, page 879.)



# Question Answering



Question parsing and classification: Syntactic parsing, entity recognition, answer classification

Document retrieval. Extraction and ranking of passages: Indexing, vector space model.

Extraction and ranking of answers: Answer parsing, entity recognition

