Introduction to Natural Language Processing

DATE15/EDA132 – Lecture 4
Information Extraction and Retrieval

Pierre Nugues

Pierre.Nugues@cs.lth.se

www.cs.lth.se/~pierre

LUNDS TEKNISKA HÖGSKOLA
Lunds universitet

LUNDS TEKNISKA HÖGSKOLA

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
Most frequent words in a collection of contemporary running texts	*the*	*de*	*der*
	of	*le (article)*	*die*
	to	*la (article)*	*und*
	in	*et*	*in*
	and	*les*	*des*
Most frequent words in the Genesis	*and*	*et*	*und*
	the	*de*	*die*
	of	*la*	*der*
	his	*à*	*da*
	he	*il*	*er*

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 (12 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)))

Lexicography

Writing dictionaries

Dictionaries for language learners should be build on real usage

- *They're just trying to score **brownie points** with politicians*
- *The boss is pleased – that's another **brownie point***

Bank of English: *brownie point* (6 occs) *brownie points* (76 occs)

Extensive use of corpora to

Find **concordances** and cite real examples

Extract of **collocations** and describe frequent pairs of words

Concordances

A word and its context:

Language	Concordances
English	s beginning of miracles did Je
	n they saw the miracles which
	n can do these miracles that t
	ain the second miracle that Je
	e they saw his miracles which
French	le premier des miracles que fi
	i dirent: Quel miracle nous mo
	om, voyant les miracles qu'il
	peut faire ces miracles que tu
	s ne voyez des miracles et des

Collocations

Word preferences: Words that occur together

	English	French	German
You say	*Strong tea*	*Thé fort*	*Kräftiger Tee*
	Powerful computer	*Ordinateur puissant*	*Starker Computer*
You don't say	*Strong computer*	*Thé puissant*	*Starker Tee*
	Powerful tea	*Ordinateur fort*	*Kräftiger Computer*

Word Preferences

Strong w			Powerful w		
strong w	*powerful w*	*w*	*strong w*	*powerful w*	*w*
161	0	*showing*	1	32	*than*
175	2	*support*	1	32	*figure*
106	0	*defense*	3	31	*minority*
…					

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
- Derive automatically symbolic rules from annotated corpora

Longer term:

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

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	*The This One Two A Three Please In*
need	7	*are will the also do*
to	1	
resolve	85	*have know do*
all	9	*the this these problems*
of	2	*the*
the	1	
important	657	*document question first*
issues	14	*thing point to*

Probabilistic Models of a Word Sequence

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

The probability $P(It\ was\ a\ bright\ cold\ day\ in\ April)$ 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.

$$
\begin{aligned}
P(S) &= P(It) \times P(was \mid It) \times P(a \mid It, was) \times P(bright \mid It, was, a) \times \\
&\quad \ldots \times P(April \mid It, was, a, bright, \ldots, in)
\end{aligned}
$$

Approximations

Bigrams: $P(w_i \mid w_1, w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$

Trigrams: $P(w_i \mid w_1, w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-2}, w_{i-1})$

Using a trigram language model:

$$P(S) \approx P(It) \times P(was \mid It) \times P(a \mid It, was) \times P(bright \mid was, a) \times \ldots \times P(April \mid day, in)$$

Maximum Likelihood Estimate

Bigrams:

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

Trigrams:

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

POS Annotation with Statistical Methods

$$t_1, t_2, t_3, ..., t_n \rightarrow noisy\_channel \rightarrow w_1, w_2, w_3, ..., w_n$$

The optimal part of speech sequence is

$$P(t_1, t_2, t_3, ..., t_n \mid w_1, w_2, w_3, ..., w_n)$$

The Bayes' rule on conditional probabilities:

$$P(A \mid B)P(B) = P(B \mid A)P(A)$$

$$\hat{T} = \arg\max P(T)P(W \mid T)$$

P(*T*) and *P*(*W*|*T*) are simplified and estimated on hand-annotated corpora, the "gold standard".

The First Term: *N*-Gram Approximation

$$P(T) = P(t_1, t_2, t_3, \ldots, t_n) \approx P(t_1)P(t_2 \mid t_1)\prod_{i=3}^{n} P(t_i \mid t_{i-2}, t_{i-1})$$

If we use a start-of-sentence delimiter **<s>**, the product initialization $P(t_1)P(t_2 \mid t_1)$ is rewritten as $P(< s >)P(t_1 \mid < s >)P(t_2 \mid < s >, t_1)$ where $P(< s >) = 1$.

We estimate the probabilities with the Maximum Likelihood:

$$P_{MLE}(t_i \mid t_{i-2}, t_{i-1}) = \frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})}$$

The Second Term

The complete word sequence knowing the part-of-speech sequence is usually approximated as:

$$P(W \mid T) = P(w_1, w_2, w_3, \ldots, w_n \mid t_1, t_2, t_3, \ldots, t_n) \approx \prod_{i=1}^{n} P(w_i \mid t_i)$$

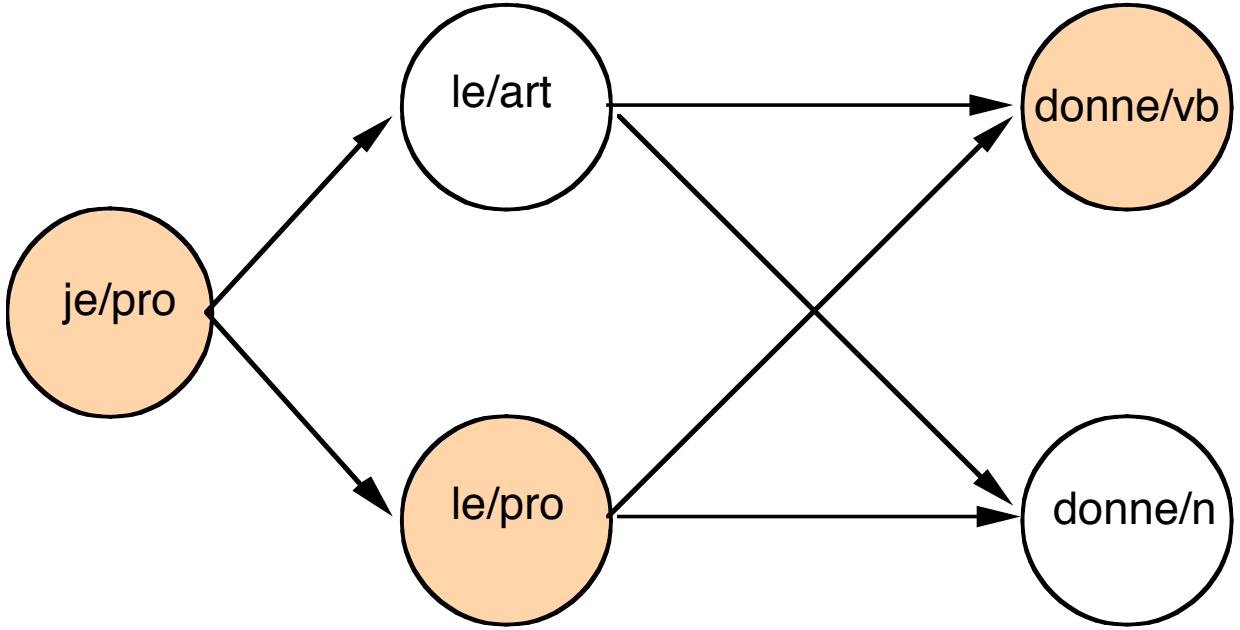
Like the previous probabilities, $P(w_i \mid t_i)$ is estimated from hand-annotated corpora using the Maximum Likelihood:

$$P_{MLE}(w_i \mid t_i) = \frac{C(w_i, t_i)}{C(t_i)}$$

For N_w different words, there are $N_p \times N_w$ values to obtain. But in this case, many of the estimates will be 0.

An Example

Je le donne ‘I give it’



1.P(pro|∅) . P(art|pro, ∅) . P(verb|pro, art) . P(*je*|pro) . P(*le*|art) . P(*donne*|verb)

2.P(pro|∅) . P(art|pro, ∅) . P(noun|pro, art) . P(*je*|pro) . P(*le*|art) . P(*donne*|noun)

3.P(pro|∅) . P(pro|pro, ∅) . P(verb|pro, pro) . P(*je*|pro) . P(*le*|pro) . P(*donne*|verb)

4.P(pro|∅) . P(pro|pro, ∅) . P(noun|pro, pro) . P(*je*|pro) . P(*le*|pro) . P(*donne*|noun)

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 then stabilized.

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	*urban guerrillas*
Perpetrator: Organization ID	*FMLN*
Perpetrator: Organization Confidence	Suspected or Accused by Authorities: *FMLN*
Physical Target: Description	*Vehicle*
Physical Target: Effect	Some Damage: *vehicle*
Human Target: Name	*Roberto Garcia Alvarado*
Human Target: Description	*attorney general*: *Roberto Garcia Alvarado* *driver* *bodyguards*
Human Target: Effect	Death: *Roberto Garcia Alvarado* No Injury: *driver* Injury: *bodyguards*

FASTUS

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

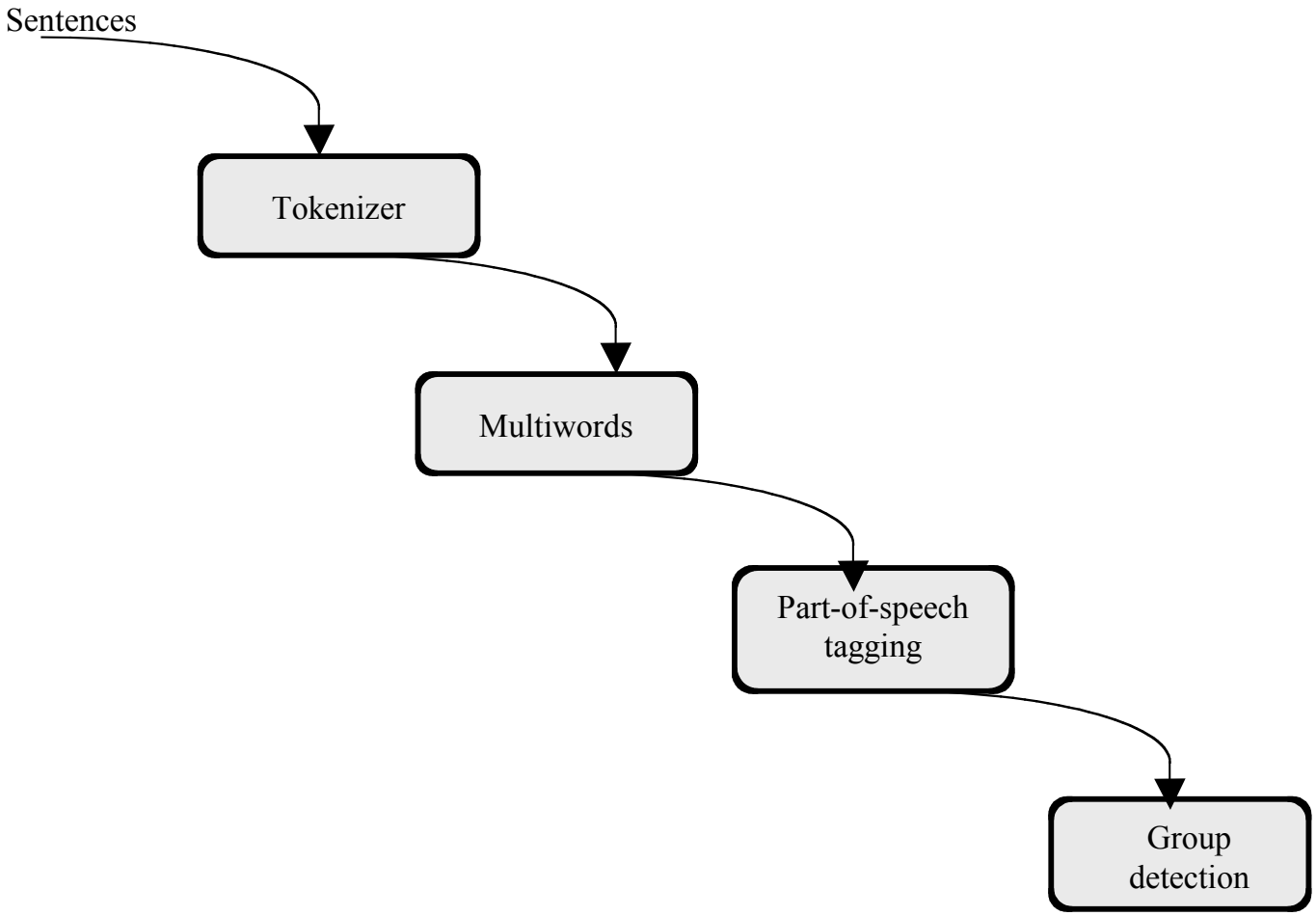
FASTUS uses partial parsers that are organized as a cascade of finite-state automata.

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.

FASTUS' Architecture



Evaluation

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

They borrowed them from library science

	Relevant documents	Irrelevant documents
Retrieved	A	B
Not retrieved	C	D

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

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:

Documents/ words	D1	D2	D3	Dm
w1				
w2				
wn				

Document Similarity

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

The similarity of documents is their cosine

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

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 $idf_j = \log(\frac{N}{n_j})$

Implementation Details

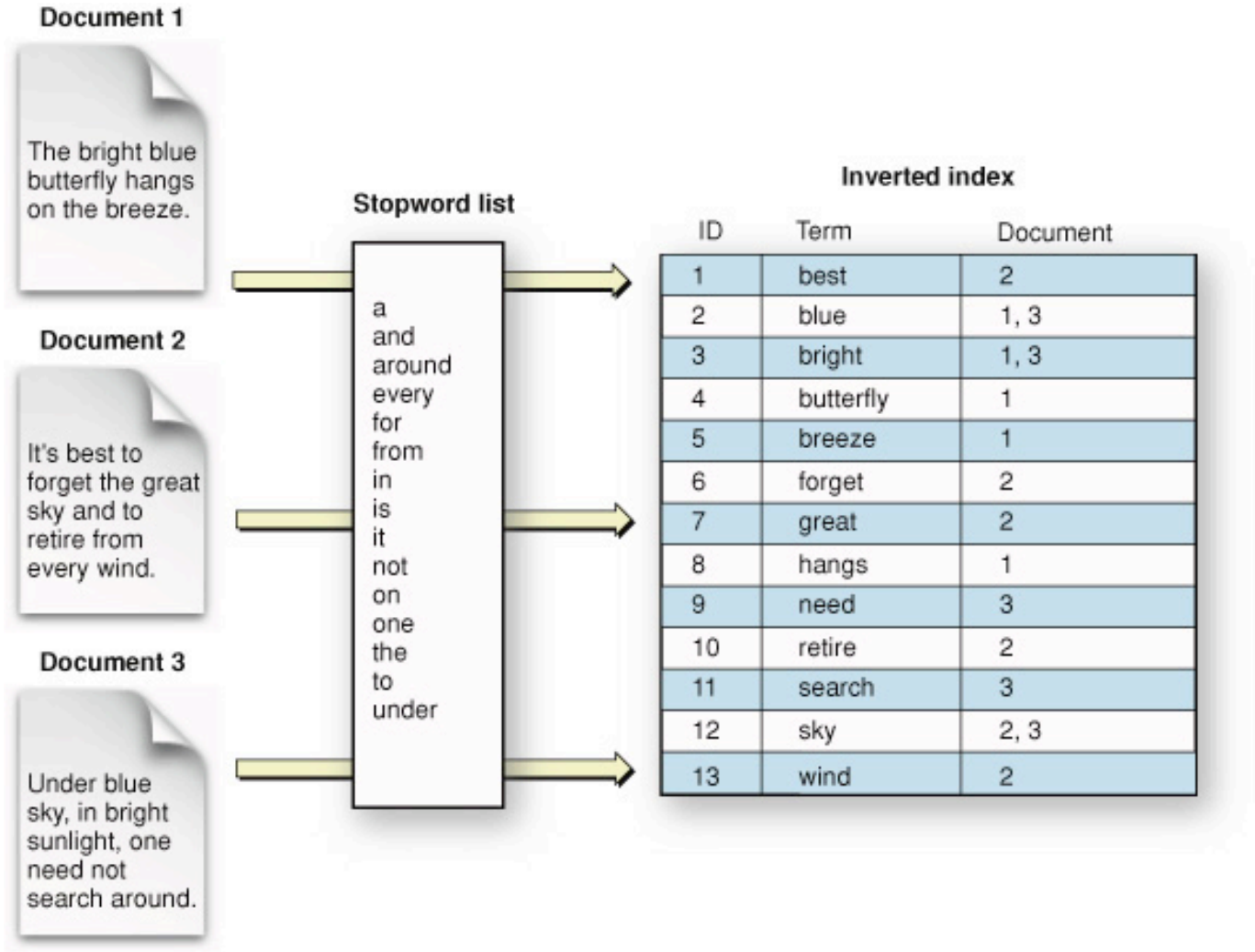
Very frequent words (stop words) can be removed

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

Inverted Index (Source Apple)



developer.apple.com/documentation/UserExperience/Conceptual/SearchKitConcepts/searchKit_basics/chapter_2_section_2.html