Artificial Intelligence EDAP01

Lecture 12.1: Natural Language Processing

Pierre Nugues

Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

February 24, 2023

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*, *Bing Translator*

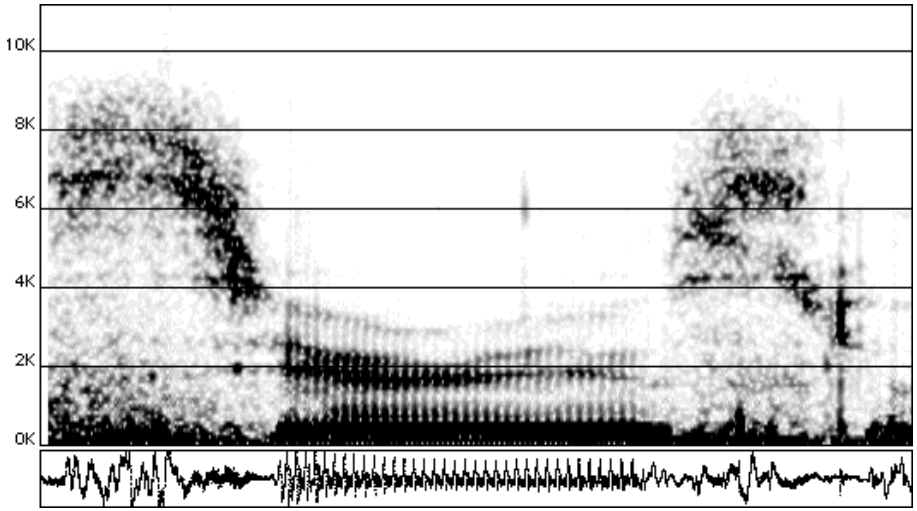
Applications of Language Processing (ctn'd)

- 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*

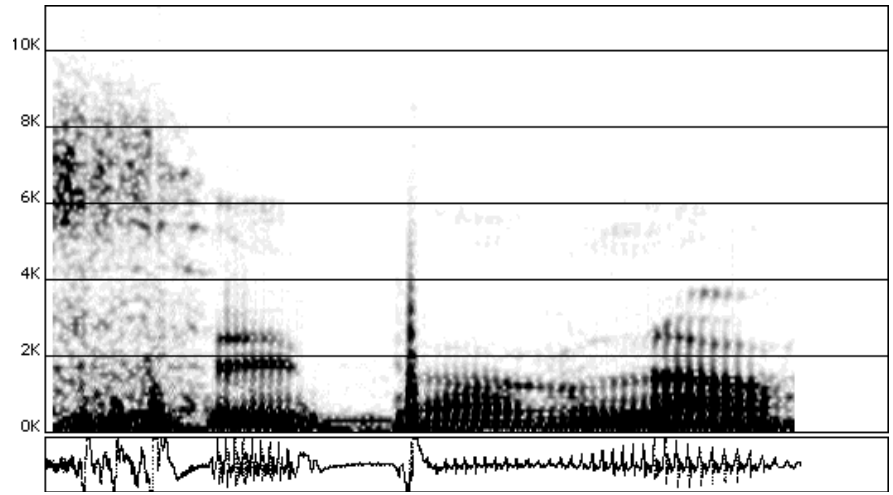
Linguistics Layers

- 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

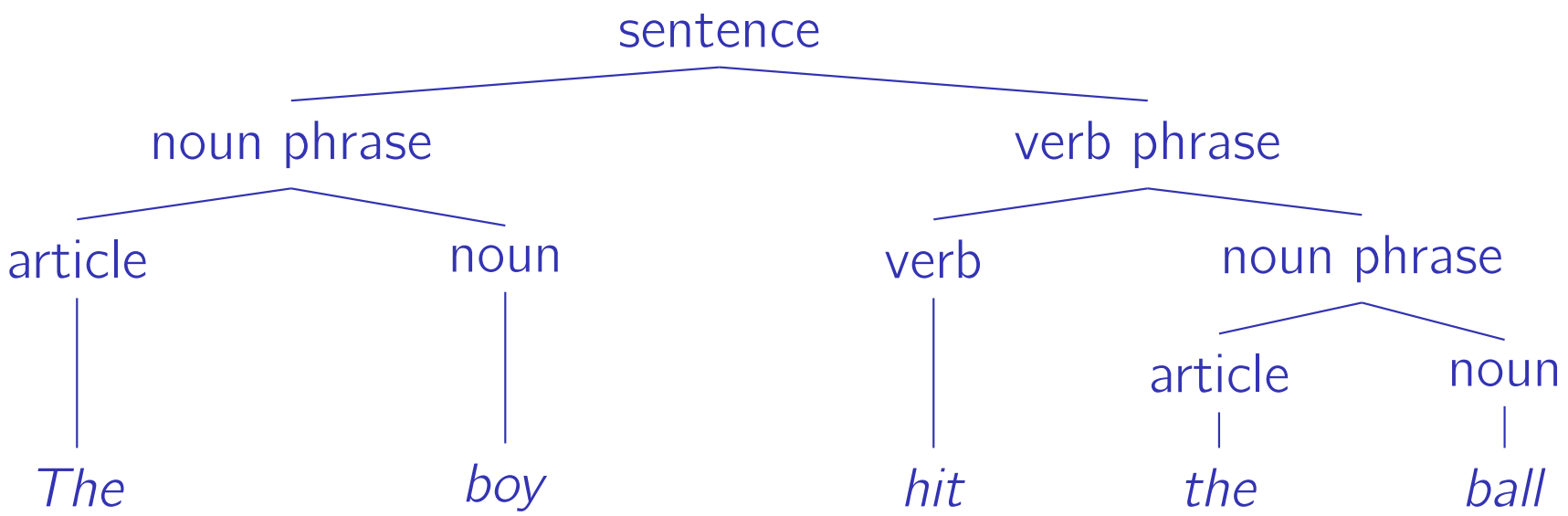
The big cat ate the gray mouse

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

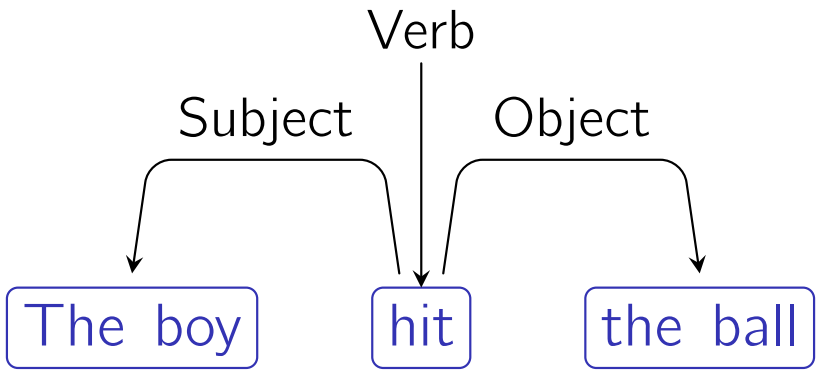
Word	Root form
worked	*to work* + verb + preterit
travaillé	*travailler* + verb + past participle
gearbeitet	*arbeiten* + verb + past participle

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:

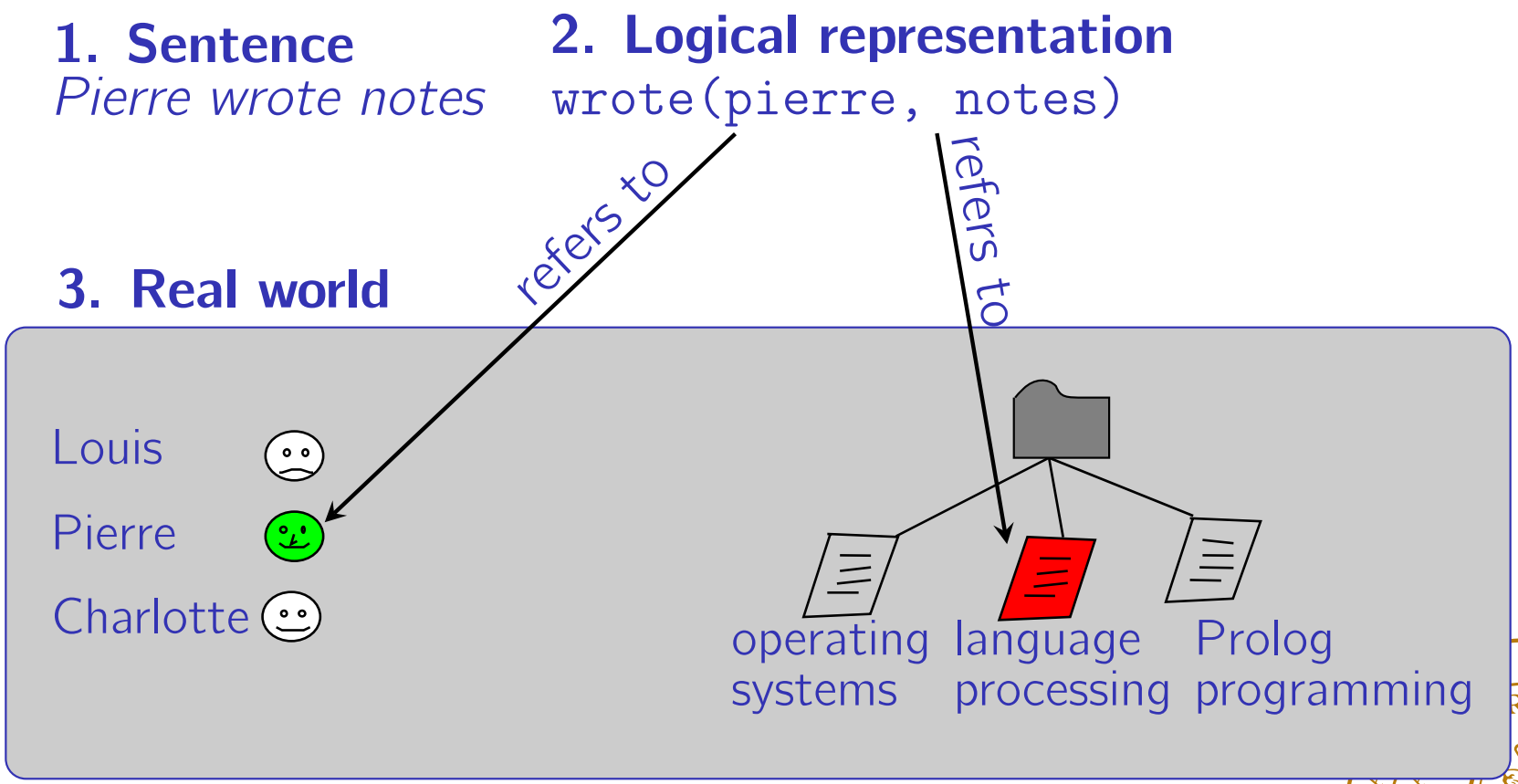
Sentence	Logical representation
Frank is writing notes	`writing(Frank, notes).`
François écrit des notes	`écrit(François, notes).`
Franz schreibt Notizen	`schreibt(Franz, Notizen).`

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)
refers to
refers to
3. Real world
Louis
Pierre
Charlotte
operating systems
language processing
Prolog programming

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;
- Inadequate theories in the beginning and lack of data: corpus, dictionaries, or reference (annotated) data;
- Models and tools have matured. Data has become available;
- Tools involve notably finite-state automata, regular expressions, logic, statistics, and machine learning;
- In general, language processing requires significant processing power;
- This overall resulted in massive improvements in most areas of NLP.

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 Genesis	*and*	*et*	*und*
	the	*de*	*die*
	of	*la*	*der*
	his	*à*	*da*
	he	*il*	*er*

Characteristics of Current Corpora

Big: starting with the Bank of English (Collins and U Birmingham) had more than 500 million words

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

Exhaustive: Common crawl: `https://commoncrawl.org/`

Multilingual: Wikipedia

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

Annotated: Part-of-speech or manually parsed (treebanks):
Characteristics/NOUN of/PREP Current/ADJ Corpora/NOUN

Corpora as Knowledge Sources

Traditional use:

- Describe usage more accurately

Machine learning

- Learn statistical/machine-learning models for speech and text processing
- Assess models and tools
- Derive automatically knowledge from annotated or unannotated corpora

Applications:

- Information extraction
- Question answering from textual sources
- Translation

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

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

Sentence Prediction with Trigrams

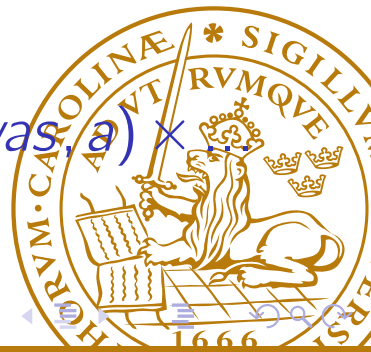
Word	Rank	More likely alternatives
We	9	*The This One Two A Three Please In*
need	7	*are will the would also do*
to	1	
resolve	85	*have know do. . .*
all	9	*the this these problems. . .*
of	2	*the*
the	1	
important	657	*document question first. . .*
issues	14	*thing point to. . .*
within	74	*to of and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

Probabilistic Models of a Word Sequence

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

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.

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



Approximations

Bigrams:

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

Trigrams:

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

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

$$\begin{aligned} P(S) \approx\ & P(It) \times P(was|It) \times P(a|It, was) \times P(bright|was, a) \times ... \\ & \times P(April|day, in). \end{aligned}$$

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.

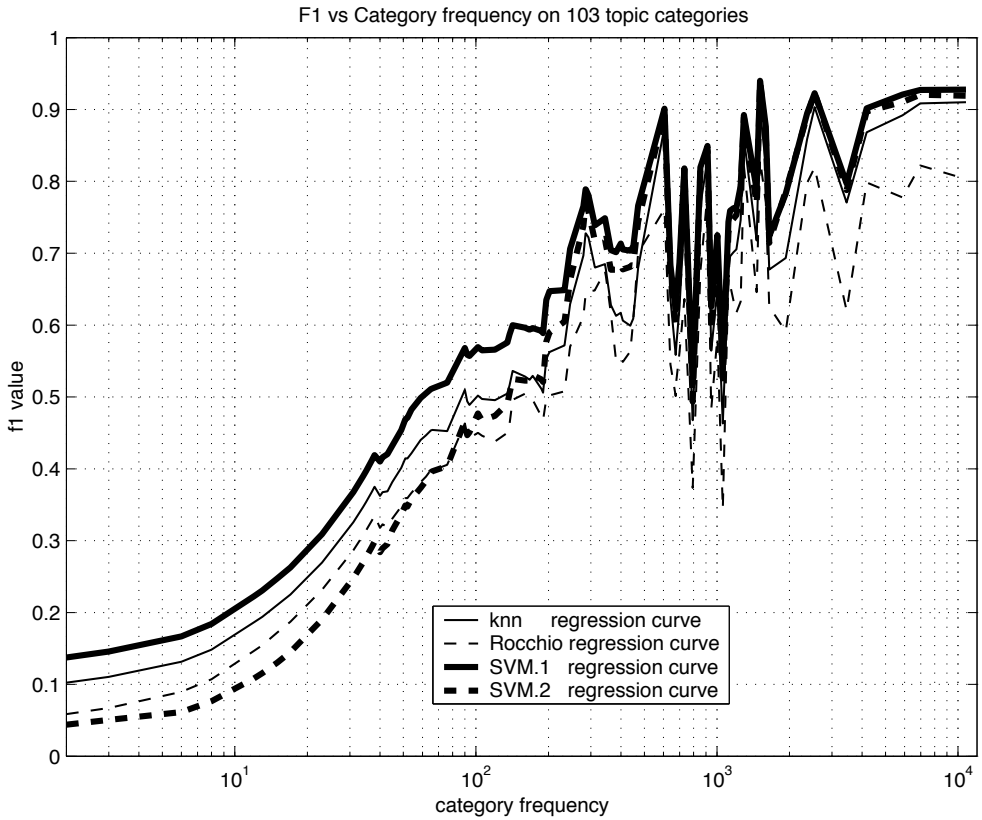
are represented as:

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



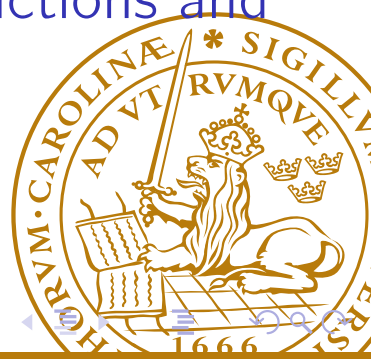
David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li, RCV1: A New Benchmark Collection for Text Categorization Research, *Journal of Machine Learning Research* 5 (2004) 361-397.

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.

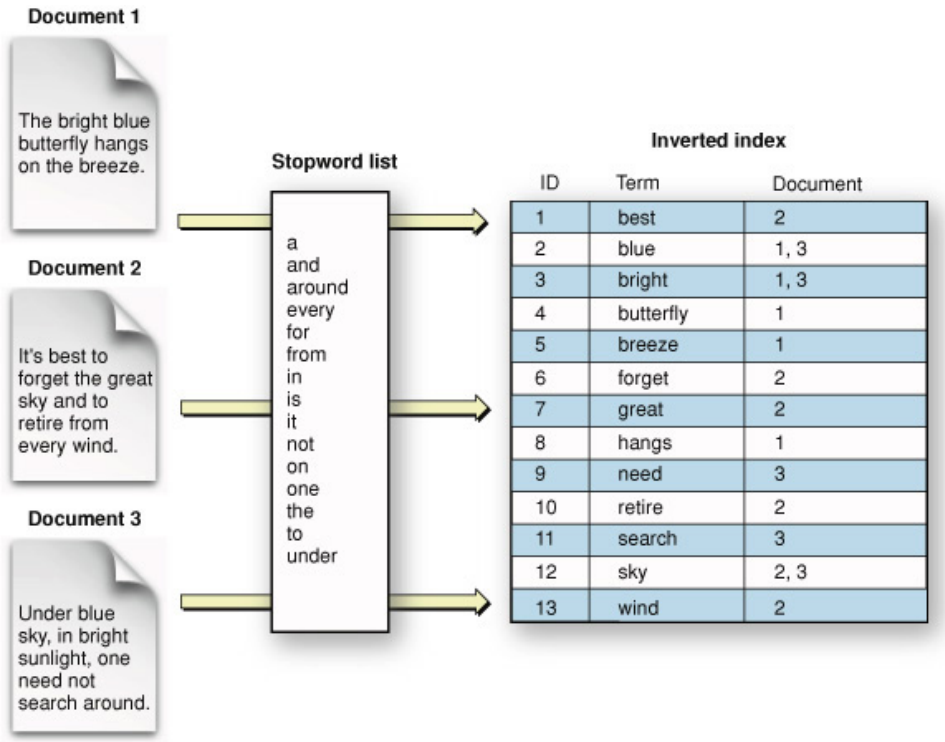


The Vector Space Model

The vector space model represents a document in word space:

Documents \Words	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

Giving a Weight

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

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:

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

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	*urban guerrillas*
Perpetrator: Organization ID	*FMLN*
Perpetrator: Organization confidence	Suspected or accused by authorities: *FM*
Physical target: Description	*vehicle*
Physical target: Effect	Some damage: *vehicle*
Human target: Name	*Roberto Garcia Alvarado*
Human target: Description	*Attorney general*: *Roberto Garcia Alvarad* *driver* *bodyguards*
Human target: Effect	Death: *Roberto Garcia Alvarado* No injury: *driver* *bodyguards*

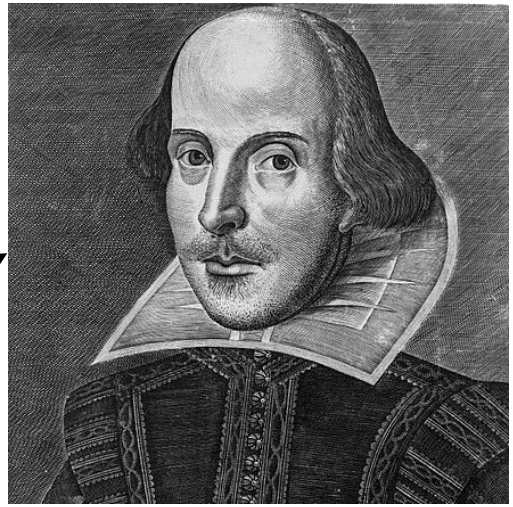
Probabilistic Models for Information Extraction

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).

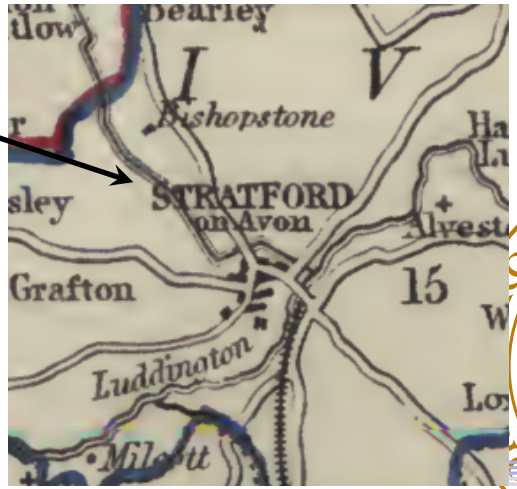
There	will	be	a	seminar	by	Andrew	McCallum	on	Friday
–	–	–	–	–	–	B-NAME	I-NAME	–	–
–	–	–	–	–	–	–	–	–	B-DATE

The speaker and date tapes are tagged by two separate hidden Markov models.
The procedure is similar to that of part-of-speech tagging.

Named Entities: Proper Nouns



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



Others Entities: Common Nouns

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

Research Directions

Text processing architectures have shifted from pipelines of linguistic modules to large language models fined-tuned on specific applications.
Regular benchmarks on popular applications:

- Classification and textual entailment
- Named entity recognition
- Question answering
- Translation
- Summarization and generation

See:

- GLUE (`https://gluebenchmark.com/`)
- SuperGLUE (`https://super.gluebenchmark.com/`)
- Machine translation (`https://www.statmt.org/wmt22/`)