Natural Language Processing: Introduction

Roberto Navigli

Dipartimento di Informatica

Sapienza Università di Roma

Your Instructor

- Associate Professor in the Department of Computer Science (Sapienza)
- Home page: http://wwwusers.di.uniroma1.it/~navigli
- Email: navigli@di.uniroma1.it

What is Natural Language Processing (NLP)?

- The branch of information science that deals with natural language information [WordNet]

But… what is Information Science?

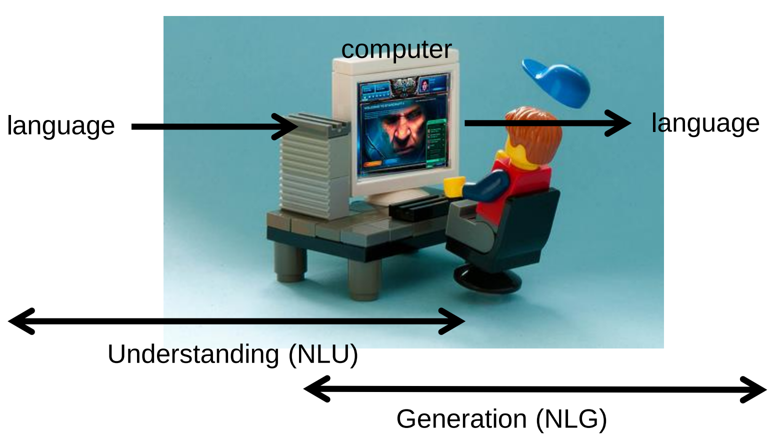
- Information science is an interdisciplinary science primarily concerned with the analysis, collection, classification, manipulation, storage, retrieval and dissemination of information [Wikipedia]

NLP: an Interdisciplinary Area

- Artificial Intelligence
- Computer Science
- Linguistics
- Psychology
- Logic
- Statistics
- Cognitive science
- Neurobiology
- …

What is Natural Language Processing II

- The use of natural language by computers as input and/or output



Natural Language Processing and Artificial Intelligence

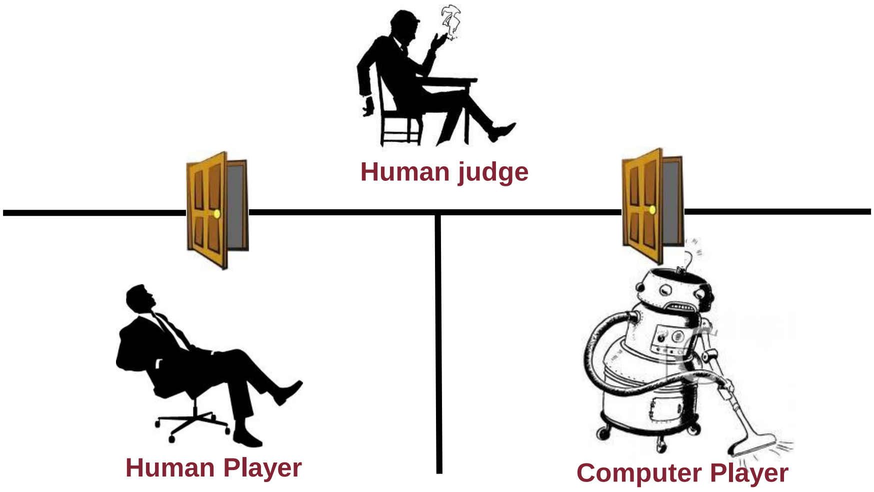
- NLP is a branch of Artificial Intelligence (AI)
- Better: NLP is the branch of AI dealing with human language
- Intelligence comprises capacities for:
  - Abstract thought
  - Understanding
  - Communication
  - Reasoning
  - Learning
  - Planning
  - Emotions
  - Problem solving
- How do we know whether a living being/system is **intelligent**?
- Idea: use language to test!

Turing Test (1950)

- A test of a machine’s ability to demonstrate intelligence
- Introduced in 1950 by Alan Turing
- "I propose to consider the question, 'Can machines think?'" Since "thinking" is difficult to define, Turing chooses to "replace the question by another, which is closely related to it and is expressed in relatively unambiguous words. […] Are there imaginable digital computers which would do well in the imitation game?"
  - Alan Turing, “Computing Machinery and Intelligence” (1950)
- Inspired by a party game, known as the “imitation game" (a man vs. a woman)

Turing Test (1950)

- A test of a machine's ability to demonstrate intelligence

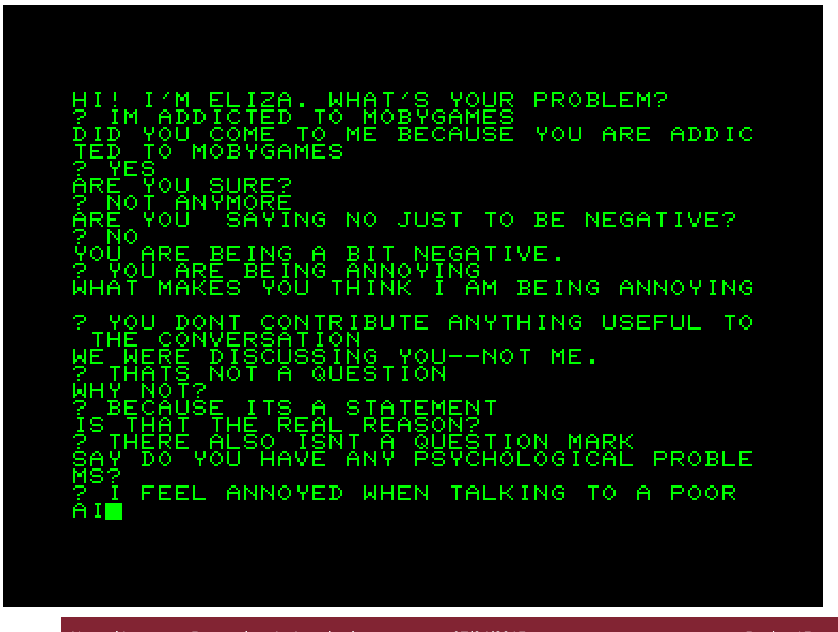


Turing Test (1950)

- A human judge engages in a (written) natural language conversation with one human and one machine
- The players try to appear human
- All participants are separated from each another
- The judge tries to determine which player is a computer and which is a human
- Assumption: NLP is AI-complete!
- In other words, if we solve NLP, we are able to solve AI

ELIZA (1966)

- An early example of computer program performing primitive natural language processing
  - Written at MIT by Joseph Weizenbaum (1966)
- Processes human replies to questions
- Uses simple parsing and substitutes keywords into template phrases



Loebner Prize Gold Medal

- $100,000 and a Gold Medal for the first computer whose responses were indistinguishable from a human's
- http://www.loebner.net/Prizef/loebner-prize.html

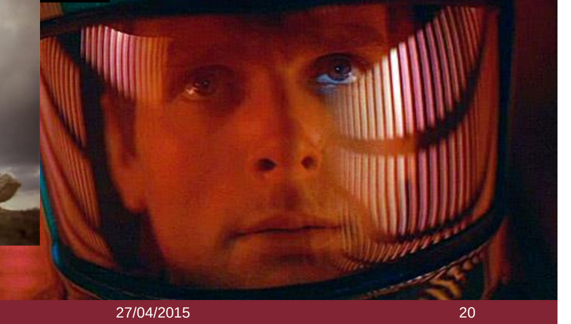
The Chinese Room (1980)

- John Searle argued against the Turing Test in his 1980 paper “Minds, Brains and Programs”
- Programs (e.g., ELIZA) could pass the Turing Test simply by manipulating symbols they do not understand
- Assume you act as a computer by manually executing a program that simulates the behavior of a native Chinese speaker

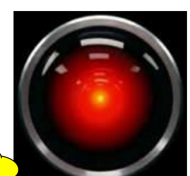
The Chinese Room (1980)

- Assume you are in a closed room with a book containing the computer program
- You receive Chinese characters through a slot in the door and process them according to the program's instructions and produce Chinese characters as output
- Would this mean that you understand?
- Would it mean you can speak Chinese?
- "I can have any formal program you like, but I still understand nothing."

The science fiction dream!



What knowledge does HAL 9000 need?



- **HAL** (**H**euristically programmed **AL**gorithmic computer)

syntax

phonetics & phonology

semantics

discourse

pragmatics

morphology

Dave Bowman: Hello, HAL. Do you read me, HAL?
HAL: Affirmative, Dave. I read you.
Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry, Dave. I'm afraid I can't do that.
Dave Bowman: What's the problem?
HAL: I think you know what the problem is just as well as I do.
Dave Bowman: What are you talking about, HAL?
HAL: This mission is too important for me to allow you to jeopardize it.
Dave Bowman: I don't know what you're talking about, HAL.

Why is NLP so hard?

- The answer is: ambiguity at the different levels of language
- Consider: “I made her duck”
  1. I cooked an animal for her
  2. I cooked an animal belonging to her
  3. I created the (plaster?) duck she owns
  4. I caused her to quickly lower her head or body
  5. I magically transformed her into a duck [ditransitive]
- Further ambiguity of spoken language:
  “eye made her duck”…

verb or noun?

dative or possessive pronoun?

create, cook or lower?

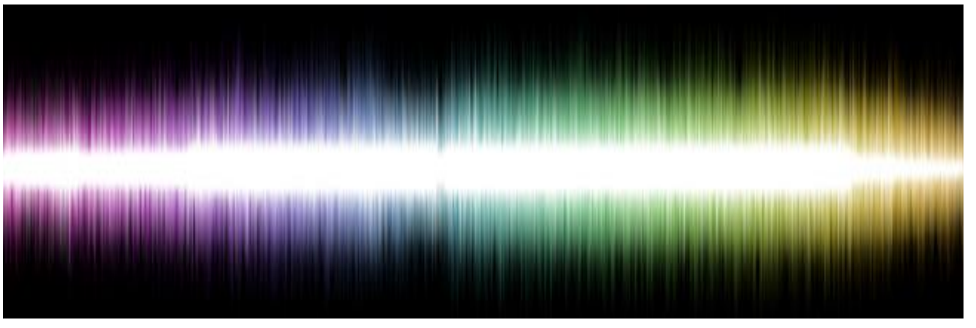
transitive or ditransitive?

The aim of NLP

- Resolving such ambiguities by means of computational models and algorithms
- For instance:
  – part-of-speech tagging resolves the ambiguity between duck as *verb* and *noun*
  – word sense disambiguation decides whether *make* means *create* or *cook*
  – probabilistic parsing decides whether *her* and *duck* are part of the same syntactic entity

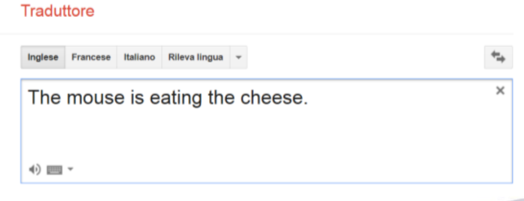
Why Aren’t We All Talking *With* Our Devices Yet?

- “Because it takes **more than just understanding a bunch of words** to provide a good voice user interface — especially on a mobile phone. We have to understand intent. But there are other factors at play here besides the technology of speech: output, interaction, and context.” [Chris Schmandt, MIT Media Lab]



Examples of the importance of NLP

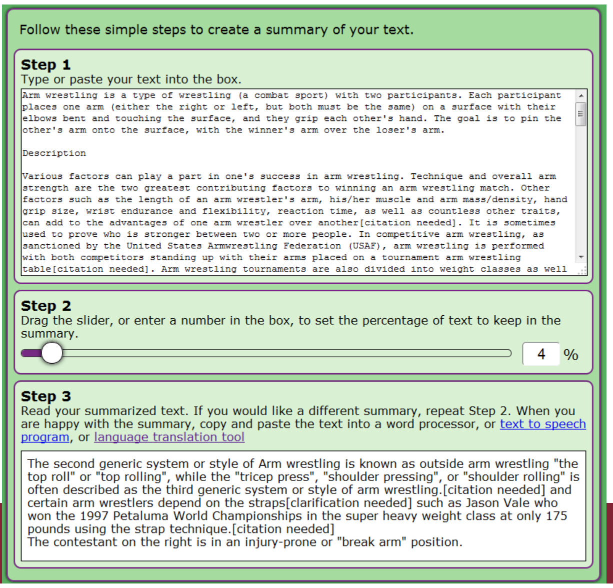
1. Machine Translation



Computer-Assisted Translation (CAT)

Examples of the importance of NLP

2. Text Summarization

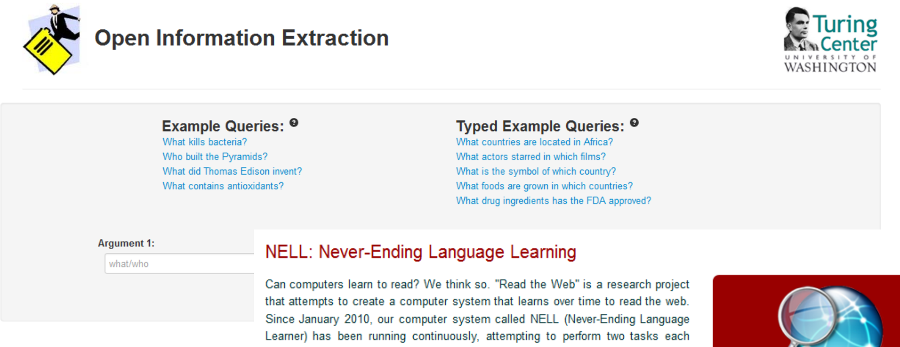


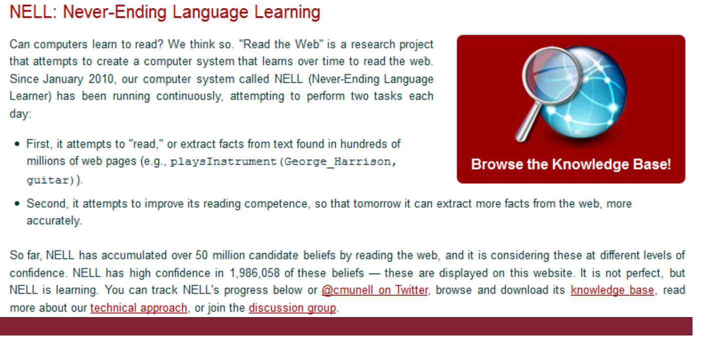
Examples of the importance of NLP

3. Personal assistance

Examples of the importance of NLP

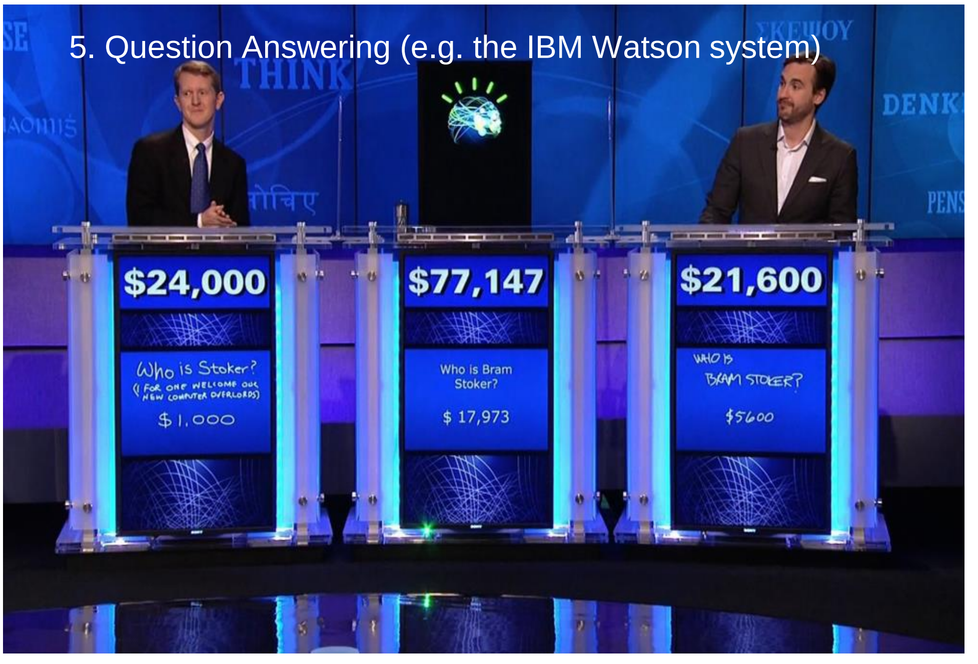
4. Information Extraction / Machine Reading





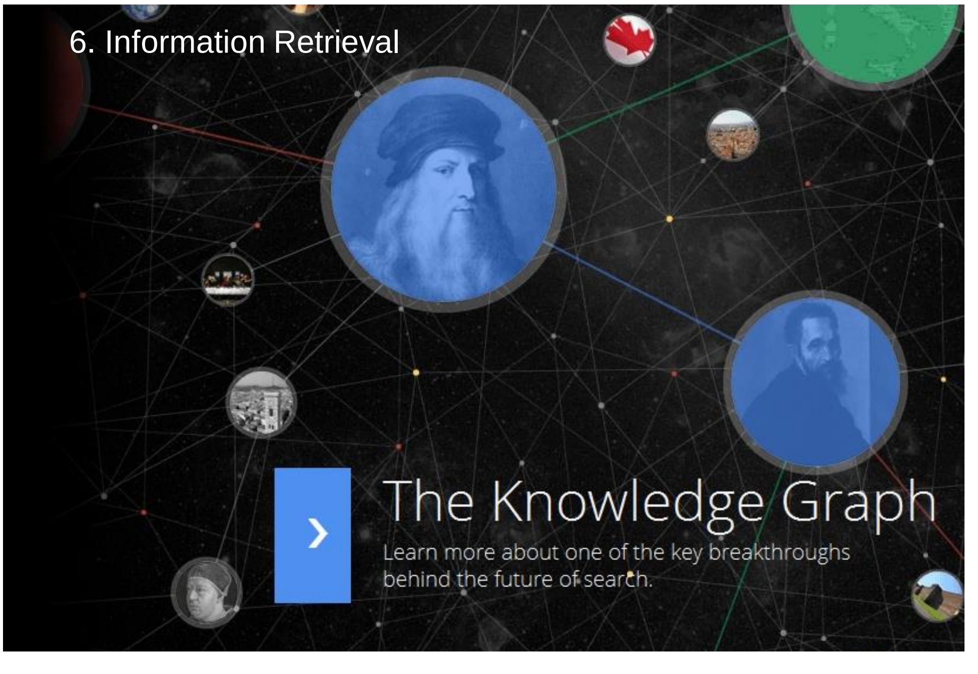
Examples of the Importance of NLP

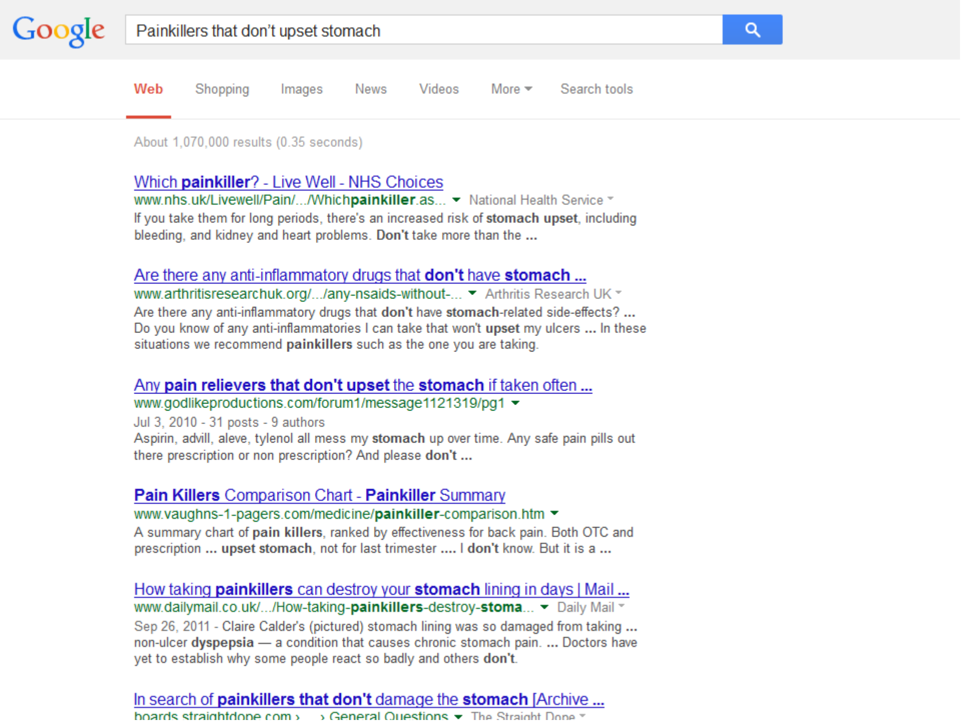
5. Question Answering (e.g. the IBM Watson system)



Examples of the importance of NLP

6. Information Retrieval





Google Painkillers that don't upset stomach

Web Shopping Images News Videos More Search tools

About 1,070,000 results (0.35 seconds)

Which painkiller? - Live Well - NHS Choices
www.nhs.uk/Livewell/Pain/.../Whichpainkiller.as... National Health Service
If you take them for long periods, there's an increased risk of **stomach upset**, including bleeding, and kidney and heart problems. **Don't** take more than the ...

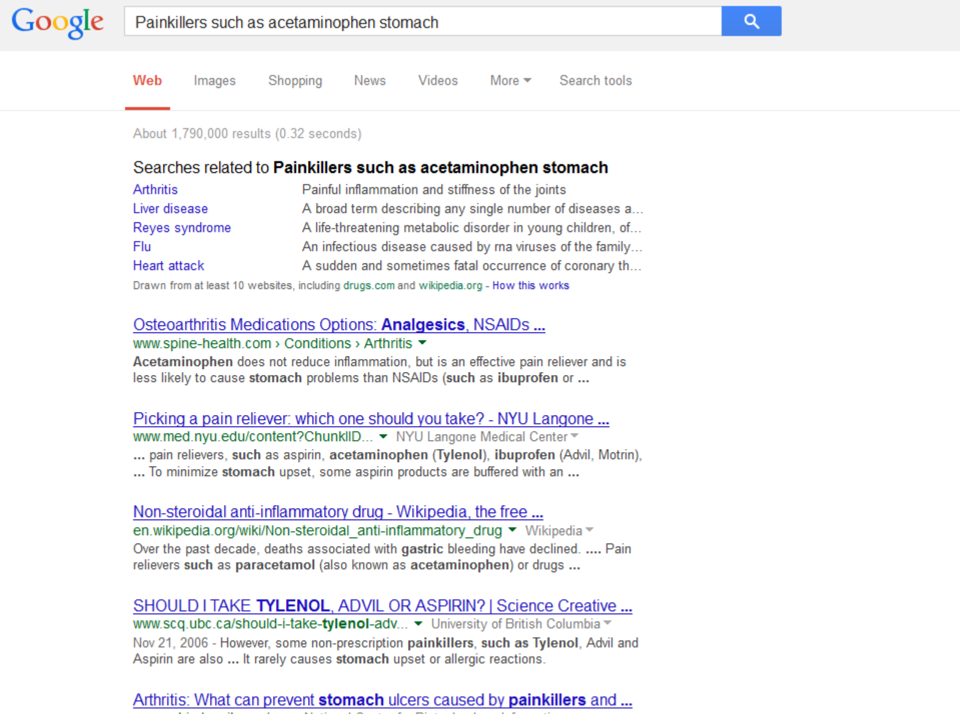
Are there any anti-inflammatory drugs that don't have stomach ...
www.arthritisresearchuk.org/.../any-nsaids-without-... Arthritis Research UK
Are there any anti-inflammatory drugs that **don't** have **stomach**-related side-effects? ... Do you know of any anti-inflammatories I can take that won't **upset** my ulcers ... In these situations we recommend **painkillers** such as the one you are taking.

Any pain relievers that don't upset the stomach if taken often ...
www.godlikeproductions.com/forum1/message1121319/pg1
Jul 3, 2010 - 31 posts - 9 authors
Aspirin, advill, aleve, tylenol all mess my **stomach** up over time. Any safe pain pills out there prescription or non prescription? And please **don't** ...

Pain Killers Comparison Chart - Painkiller Summary
www.vaughns-1-pagers.com/medicine/**painkiller**-comparison.htm
A summary chart of **pain killers**, ranked by effectiveness for back pain. Both OTC and prescription ... **upset stomach**, not for last trimester .... I **don't** know. But it is a ...

How taking painkillers can destroy your stomach lining in days | Mail ...
www.dailymail.co.uk/.../How-taking-**painkillers**-destroy-**stoma**... Daily Mail
Sep 26, 2011 - Claire Calder's (pictured) stomach lining was so damaged from taking ... non-ulcer **dyspepsia** — a condition that causes chronic stomach pain. ... Doctors have yet to establish why some people react so badly and others **don't**.

In search of painkillers that don't damage the stomach [Archive ...
boards.straightdope.com › ... › General Questions The Straight Dope



Google Painkillers such as acetaminophen stomach

Web Images Shopping News Videos More Search tools

About 1,790,000 results (0.32 seconds)

Searches related to **Painkillers such as acetaminophen stomach**

Arthritis	Painful inflammation and stiffness of the joints
Liver disease	A broad term describing any single number of diseases a...
Reyes syndrome	A life-threatening metabolic disorder in young children, of...
Flu	An infectious disease caused by rna viruses of the family...
Heart attack	A sudden and sometimes fatal occurrence of coronary th...

Drawn from at least 10 websites, including drugs.com and wikipedia.org - How this works

Osteoarthritis Medications Options: Analgesics, NSAIDs ...
www.spine-health.com › Conditions › Arthritis
Acetaminophen does not reduce inflammation, but is an effective pain reliever and is less likely to cause **stomach** problems than NSAIDs (**such as ibuprofen** or ...

Picking a pain reliever: which one should you take? - NYU Langone ...
www.med.nyu.edu/content?ChunkIID... NYU Langone Medical Center
... pain relievers, **such** as aspirin, **acetaminophen** (**Tylenol**), **ibuprofen** (Advil, Motrin), ... To minimize **stomach** upset, some aspirin products are buffered with an ...

Non-steroidal anti-inflammatory drug - Wikipedia, the free ...
en.wikipedia.org/wiki/Non-steroidal_anti-inflammatory_drug Wikipedia
Over the past decade, deaths associated with **gastric** bleeding have declined. .... Pain relievers **such** as **paracetamol** (also known as **acetaminophen**) or drugs ...

SHOULD I TAKE TYLENOL, ADVIL OR ASPIRIN? | Science Creative ...
www.scq.ubc.ca/should-i-take-**tylenol**-adv... University of British Columbia
Nov 21, 2006 - However, some non-prescription **painkillers**, **such as Tylenol**, Advil and Aspirin are also ... It rarely causes **stomach** upset or allergic reactions.

Arthritis: What can prevent stomach ulcers caused by painkillers and ...

I need to reason…

$$\begin{tikzcd}
\text{Painkiller} \arrow[r, "\text{not}"'] & \text{Upset stomach} \\
\text{Acetaminophen} \arrow[u, "\text{is-a}"'] \arrow[r, "\text{not}"'] & \text{Stomach problem} \arrow[u, "\text{is-a}"']
\end{tikzcd}$$

Natural Language Processing in Brief

- Morphological Analysis
- Language modeling
- Part-of-speech tagging
- Syntactic Parsing
- Computational Lexical Semantics
- Statistical Machine Translation
- Discourse and Dialogue
- Text Summarization
- Question Answering
- Information Extraction and Text Mining
- Speech Processing

We are talking about words!

What are words?

- Basic building block of language
- Every human language, either spoken or written, is composed of words
- A word is the smallest free form that can be uttered in isolation with semantic and pragmatic content
- Made up of morphemes (smallest component of word that has semantic meaning)
  - tree
  - tree+s
  - sub+tree
  - un+important

affixes

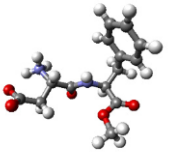
root

We need to perform a morphological analysis

- "Morphology is the study of the way words are built up from smaller meaning-bearing units" (Jurafsky & Martin, 2000)
- The meaning-bearing units are called morphemes
- Two main types of morphemes:
  - Stem or root: the main morpheme of a word
  - Affixes: prefixes (re-write), suffixes (beauti-ful-ly), infixes and circumfixes
- In order to detect these components we need to perform morphological parsing

The concept of parsing

Input



Structure for the input

Morphological Parsing

Input	Morphologically Parsed Output
beagles	beagle +N +PL
cities	city +N +PL
buy	buy +N +SG or buy +V
buying	buy +V +PRES-PART
bought	buy +V +PAST-PART or buy +V +PAST

- What do we need to build a morphological parser?

Ingredients for a Morphological Parser

- Lexicon: the list of stems and affixes

Stem	Part of speech

- Morphotactics: the model of morpheme ordering in the language of interest (e.g., main stem+plural)
- Orthographic rules: spelling rules about how morphemes combine to form a word (e.g., city +s -> cities)

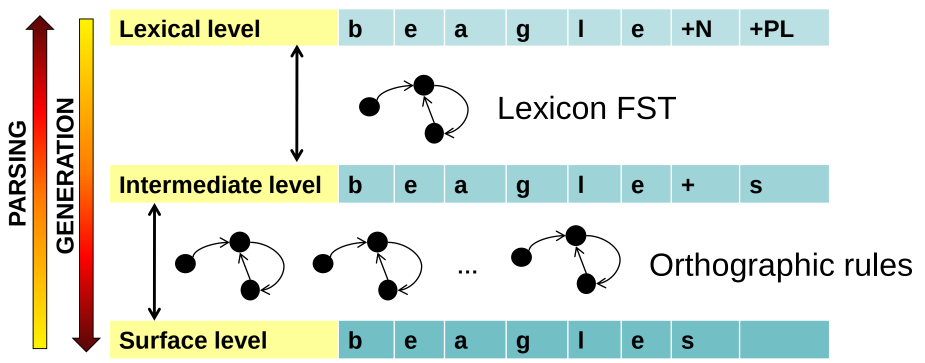
Ingredients for a Morphological Parser

- Lexicon: the list of stems and affixes

Stem	Part of speech
beach	N
beachwards	ADV
beachwear	N
beachy	ADJ
beacon	N

- Morphotactics: the model of morpheme o the language of interest (e.g., main stem+
- Orthographic rules: spelling rules about how morphemes combine to form a word (e.g., city -> cities)

Three levels: lexical, intermediate and surface level



Morphological Parsing with Finite State Transducers

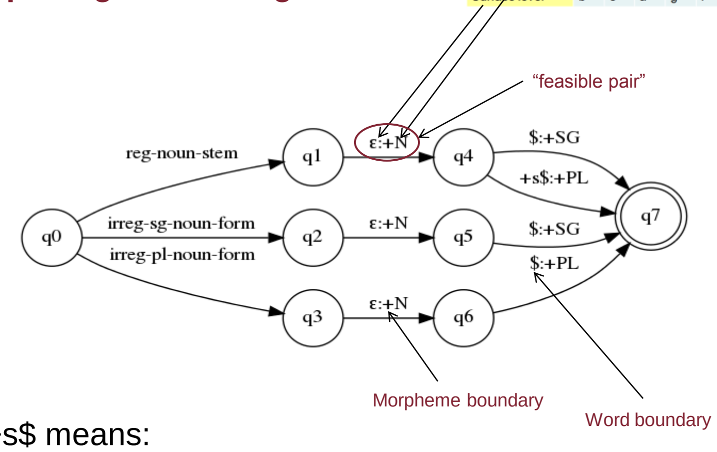
- We would like to keep distinct the surface and the lexical levels
- We need to build mapping rules between concatenation of letters and morpheme+feature sequences

Lexical level	**b**	**e**	**a**	**g**	**l**	**e**	**+N**	**+PL**
Surface level	b	e	a	g	l	e	s	

- A finite-state transducer implements two-level morphology and maps between one set of symbols to another
  - Done using a (two-tape) finite-state automaton
  - Recognizes or generates **pairs** of strings

Morphological Parsing with FSTs

Lexical level	**b**	**e**	**a**	**g**	**l**	**e**	**+N**	**+PL**
Surface level	b	e	a	g	l	e	s	



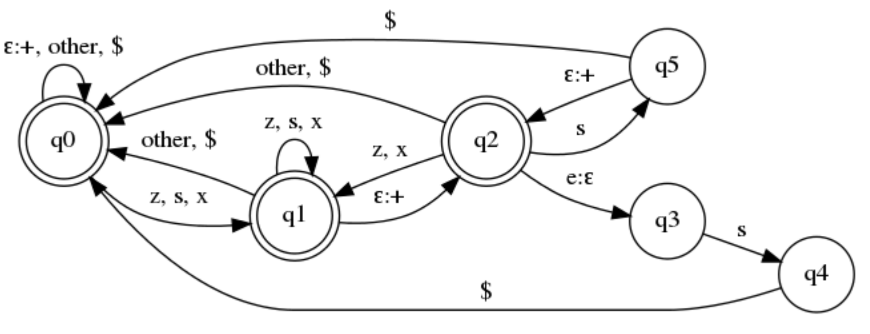
- +s$ means:
  - "+" (morpheme boundary), "s" (the morpheme), "$" word boundary

But: Just concatenating morphemes doesn't always work!

- **box+s** = **boxs**, rather than **boxes**
- **boxes** woudn't be recognized!
- Why? Because of a spelling change at morpheme boundaries
- We need to introduce spelling (or orthographical) rules
  - And implement these rules as FSTs

Rule	Description	Example
Consonant doubling	1 consonant doubled before –ing or –ed	beg/begging, embed/embedded
E deletion	e taken out before –ing or -ed	make/making
E insertion	e added after –s, -z, -x, -ch, -sh before s	watch/watches
Y replacement	-y replaced by –ies before –s, -i before -ed	try/tries, try/tried, city/cities

Example: transducer for the "E insertion" rule



- So one can build a transducer for each spelling and orthographical rule
- For example: foxes -> fox+s

  (q0 -> q0 -> q1 -> q2 -> q3 -> q4 -> q0)

Where do we go from here?

- We are now able to process text at the morphological level
- We can work on word combinations
- For instance, from The Telegraph:
  - Escort claims Berlusconi's 'bunga bunga' parties full of young…
- What comes next?
  - Old women? Boys? Girls?
  - **It depends! On what?**

1-grams (unigrams): just a single word

- Absolute count of each word (e.g. on the Web):

</S>	95119665584
<S>	95119665584
,	30578667846
.	22077031422
<UNK>	21594821357
the	19401194714
-	16337125274
of	12765289150
and	12522922536
:	12255665115
to	11557321584
)	9036544694
(	8912668768
a	7841087012
in	7490628883

2-grams (bigrams): sequences of two words

- Absolute count of two words (e.g. on the Web):

young gipsy	267
young gir	1203
young gir.s	79
young giraffe	817
young giraffes	288
young giral	77
young girel	245
young girels	227
young girils	266
young giris	59
young girks	379
young girl	1716008
young girl'	64
young girl.I	138
young girla	117

3-grams (trigrams): sequences of 3 words

of young bone	75
of young bones	56
of young boobs	66
of young book	177
of young born	362
of young botanists	41
of young bovine	41
of young bowlers	147
of young boxers	184
of young boy	11928
of young boys	40490
of young brain	60
of young brains	220
of young branches	421
of young branchlets	65

of young german	80
of young giant	245
of young giants	55
of young gibbons	48
of young gifted	589
of young ginger	96
of young girl	11631
of young girlfriends	153
of young girlhood	48
of young girls	86186
of young girlsnon	101
of young global	119
of young globular	166

Word prediction and N-gram models

- We can create language models, called N-gram models
  - they predict the next word from the previous N-1 words
  - they define probability distributions over strings of text
- Useful in:
  - Speech recognition
  - Handwriting recognition
  - Machine translation
  - Spelling correction
  - Part-of-speech tagging

Simple N-grams

- Our aim is to compute the probability of a word given some history:

$$P(w|h)$$

- For instance:
  - P(rapa|qui non l'ha capito nessuno che questa è una) = C(qui non l'ha capito nessuno che questa è una rapa)/ C(qui non l'ha capito nessuno che questa è una)
- How easily can we calculate this probability?

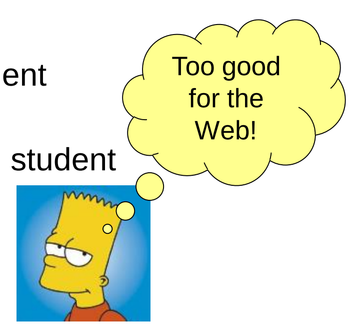
1) It depends on the corpus

- With a large corpus, such as the Web, we can compute these counts
- But: try yourself!

- Bad luck?
- The Web is not big enough **(!)** to provide good estimates for most counts

2) Language is infinite

- You are a good student
  – About 4,630,000 results (0.19 seconds)
- You are a very good student
  – About 2,040,000 results (0.36 seconds)
- You are a very very good student
  – 7 results (0.26 seconds)
- You are a very very very good student
  – 1 result (0.25 seconds)
- You are a very very very very good student
  – 0 results!



So what is a language model?

A language model is a probability distribution over word sequences

- P("You are a good student") will be high
- P("You are a very very very very good student") will be very low

We need to estimate probabilities

- Chain rule of probability:

$$P(w_1 ... w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)...P(w_n \mid w_1^{n-1})$$

$$= \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

- Not enough - we need to approximate:

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-1})$$

- Independence assumption: Markov assumption of order N-1
- How to estimate these bigram probabilities (N=2)?

Calculating the Relative Frequency Estimate

- Bracket sentences with <s> and </s>
- Count the frequency of each bigram
- We estimate the bigram probabilities by normalizing counts from a corpus:

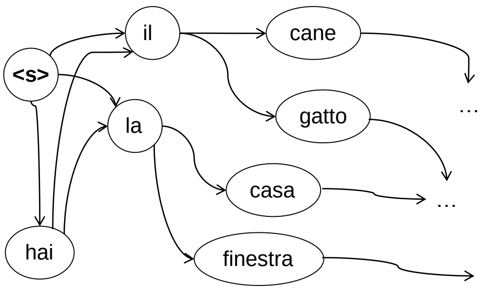
$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)} = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

- General case with N-gram probabilities:

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

Bigram Models are Markov chains

	<s>	**il**	**…**	**cane**
<s>	0	0.3	…	0.01
il	0	0	…	0.2
…	0	…	…	…
cane	0	0.1	…	0.01



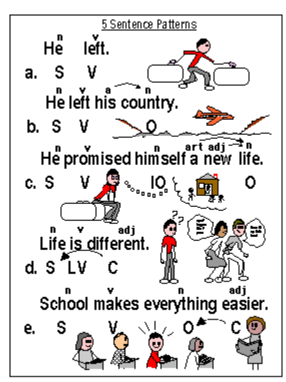
A random process usually characterized as memoryless: the next state depends only on the current state

Word classes / Parts Of Speech (POS) / Lexical Tags

- Why do we need **word classes**?
- They give important information about the word and its neighbours
- He is running the race He decided to race for the job

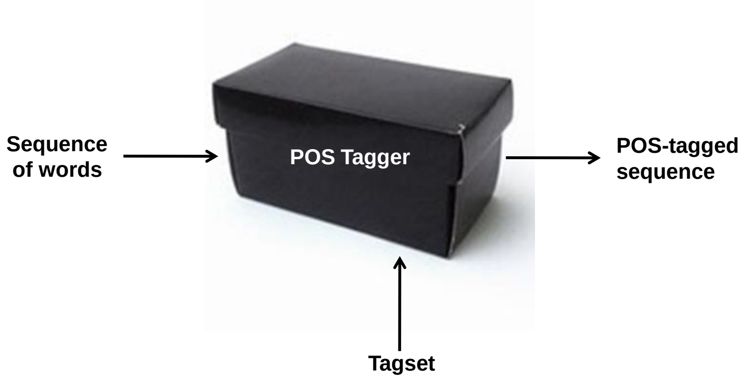
Word classes / Parts Of Speech (POS) / Lexical Tags

- Why do we need **word classes**?
- They give important information about the word and its neighbours
- Useful for recognizing speech and correcting spelling errors:
  - What is likely to come after an adjective?
  - a verb? a preposition? a noun?



Part-of-Speech Tagging

- It consists of assigning a part-of-speech tag to each word in a text automatically



Part-of-Speech Ambiguity

- Unfortunately, the same word can be tagged with different POS tags:
  - How to increase the water pressure from a **well**?
  - Tears **well** in her eyes
  - The wound is nearly **well**
  - The party went **well**
- Part-of-Speech tagging is a disambiguation task!

An Example: an English sentence

sentence:	The	oboist	Heinz	Holliger	has	taken	a	hard	line	about	the	problems	.
original:	DT	NN	NNP	NNP	VBZ	VBN	DT	JJ	NN	IN	DT	NNS	.
universal:	DET	NOUN	NOUN	NOUN	VERB	VERB	DET	ADJ	NOUN	ADP	DET	NOUN	.

Example English sentence with its language specific and corresponding universal POS tags.

Stochastic Part-of-Speech Tagging (since 1970s)

- Stochastic POS tagging uses probabilities to tag
- **Idea:** use Hidden Markov Models to select the most-likely tags for a given sequence of words

$$\hat{t}_1^n = \underset{t_1^n \in Tagset^n}{\arg\max}\, P(t_1^n \mid w_1^n)$$

- But how can we calculate these probabilities?

Holy Bayes!

- Remember?

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{P(y)}$$

- Let's apply Bayes' Theorem to our formula:

$$\hat{t}_1^n = \underset{t_1^n \in Tagset^n}{\arg\max} \frac{P(w_1^n \mid t_1^n)P(t_1^n)}{P(w_1^n)} = \underset{t_1^n \in Tagset^n}{\arg\max} \underbrace{P(w_1^n \mid t_1^n)}_{\text{likelihood}} \underbrace{P(t_1^n)}_{\text{prior}}$$

- Still hard to compute!

HMM taggers make two simplifying assumptions

1) The probability of a word depends only on its own part-of-speech tag

$$P(w_1^n \mid t_1^n) = P(w_1 \mid t_1^n)P(w_2 \mid w_1, t_1^n) \ldots P(w_n \mid w_1^{n-1}, t_1^n) \approx \prod_{i=1}^{n} P(w_i \mid t_i)$$

2) The probability of a tag appearing depends only on the previous tag (**bigram assumption**)

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-1})$$

The two simplifying assumptions in action

$$\hat{t}_1^n = \underset{t_1^n \in Tagset^n}{\arg\max}\, P(t_1^n \mid w_1^n) \approx \underset{t_1^n \in Tagset^n}{\arg\max} \prod_{i=1}^{n} P(w_i \mid t_i) P(t_i \mid t_{i-1})$$

- Now we can easily estimate these two probabilities from a part-of-speech tagged corpus

$$P(t_i \mid t_{i-1}) = \frac{c(t_{i-1}, t_i)}{c(t_{i-1})}$$

$$P(w_i \mid t_i) = \frac{c(t_i, w_i)}{c(t_i)}$$

Estimating Conditional Probabilities for Tags

$$P(t_i \mid t_{i-1}) = \frac{c(t_{i-1}, t_i)}{c(t_{i-1})}$$

- Examples:

$$P(NN \mid DT) = \frac{c(DT, NN)}{c(DT)} = \frac{58{,}800}{120{,}000} = 0.49$$

$$P(JJ \mid DT) = \frac{c(DT, JJ)}{c(DT)} = \frac{52{,}800}{120{,}000} = 0.42$$

$$P(IN \mid DT) = \frac{c(DT, IN)}{c(DT)} = \frac{120}{120{,}000} = 0.001$$

Estimating Conditional Probabilities for Words

$$P(w_i \mid t_i) = \frac{c(t_i, w_i)}{c(t_i)}$$

- Examples:

$$P(is \mid VBZ) = \frac{c(VBZ, is)}{c(VBZ)} = \frac{9{,}600}{20{,}000} = 0.48$$

$$P(are \mid VBZ) = \frac{c(VBZ, are)}{c(VBZ)} = \frac{2}{20{,}000} = 0.0001$$

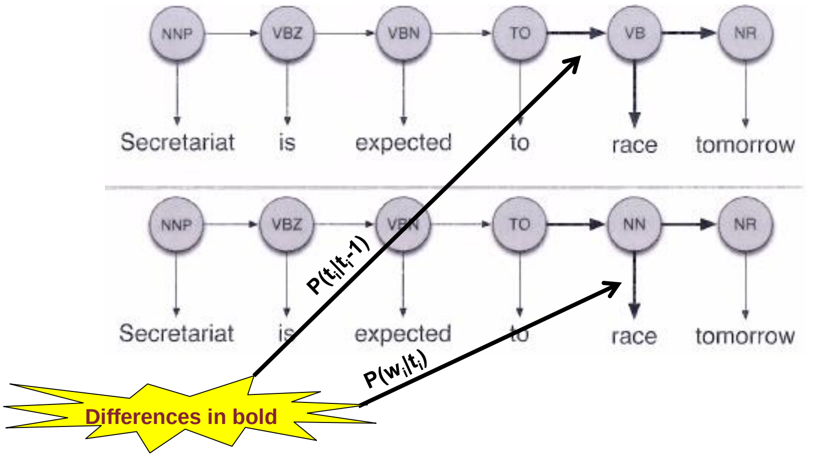
An Example

- You can book your flight
  – P(book|VB)=0.0004
  – P(book|NN)=0.0002
  – P(VB|MD)=0.5
  – P(NN|MD)=0.001

– So what is the most likely tag for book?

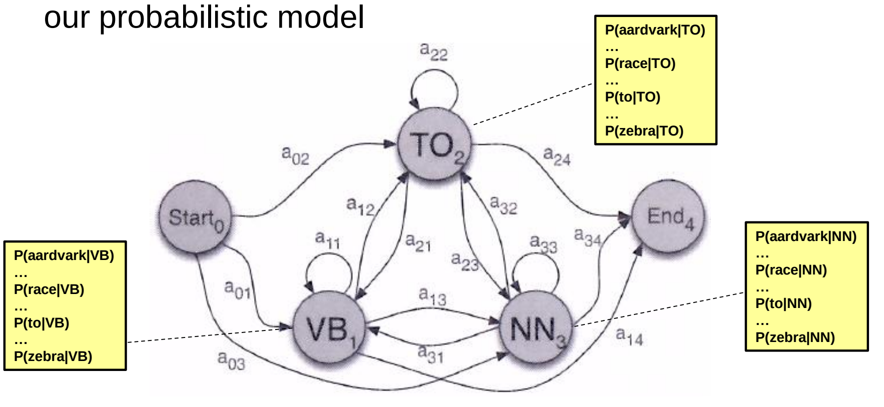
Another Example from Jurafsky & Martin 2008

- How to choose the correct global tagging sequence?



Hidden Markov Models (HMM)

- A HMM allows us to talk both about observed events (word sequence in input) and unobserved (hidden) events (part-of-speech tags) that are causal factors in our probabilistic model



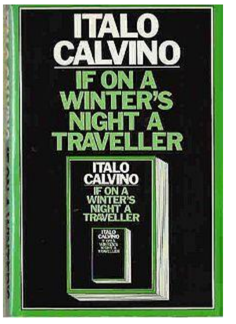
Go to the Machine Learning class!!!

So far about word ordering…

- Morphological analysis: Finite-state transducers
- N-gram models: Computing probabilities for word sequences
- Part-of-speech classes: equivalence classes for words
- We now move to… formal grammars!

Example

- Se una notte d'inverno un viaggiatore
- *Se notte una d'inverno un viaggiatore
- Una notte se d'inverno un viaggiatore
- *Se un notte d'inverno una viaggiatore
- Se una notte un viaggiatore d'inverno
- Se un viaggiatore d'inverno una notte
- *Se un una notte viaggiatore d'inverno
- *Se un una d'notte viaggiatore inverno
- ~Se un inverno d'notte un viaggiatore
- Se d'inverno un viaggiatore una notte
- Se d'inverno una notte un viaggiatore
- …



Context-Free Grammars (CFGs)

- A context-free grammar (CFG) or phrase-structure grammar is a formal grammar defined as a 4-tuple:

$$G = (N, T, P, S)$$

- where:
  - N is the set of nonterminal symbols (phrases or clauses)
  - T is the set of terminal symbols (lexicon)
  - P is the set of productions (rules), a relation $\subseteq N \times (N \cup T)^*$
  - S is the start symbol such that $S \in N$, $\exists(S, \beta) \in P$

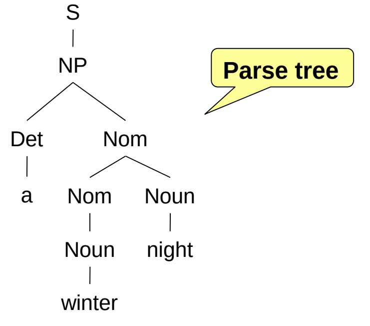
Example

- N = { S, NP, Nom, Det, Noun },
- T = { a, the, winter, night },
- P = {
  - S → NP
  - NP → Det Nom
  - Nom → Noun | Nom Noun
  - Det → a | the
  - Noun → winter
  - Noun → night
  - },
- S

CFG as tools for…

N = { S, NP, Nom, Det, Noun },
T = { a, the, winter, night },
P = {
S → NP
NP → Det Nom
Nom → Noun | Nom Noun
Det → a | the
Noun → winter
Noun → night
},
S

- Generating sentences
  - G generates "a winter night"
  - There exists a derivation (sequence of rule expansions)



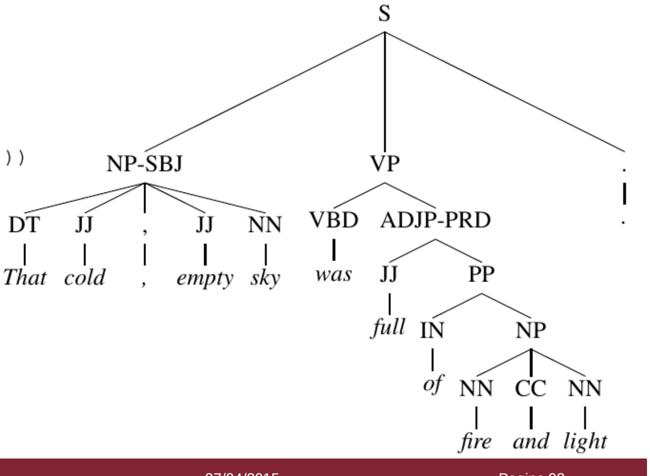
- Assigning a structure to a sentence
  - What is the structure for "a winter night"?

Treebanks

- CFGs can be used to assign a parse tree to any valid sentence
- We can build a corpus, called treebank, whose sentences are annotated with parse trees
- The most popular project of this kind is the Penn Treebank
  - From the Brown, Switchboard, ATIS and Wall Street Journal corpora of English
    - Wall Street Journal: 1.3 million words
    - Brown Corpus: 1 million words
    - Switchboard: 1 million words
  - All tagged with Part of Speech & syntactic structure
  - Developed 1988-1994

"That cold, empty sky was full of fire and light."

```
((S
   (NP-SBJ (DT That)
     (JJ cold) (, ,)
     (JJ empty) (NN sky) )
   (VP (VBD was)
     (ADJP-PRD (JJ full)
       (PP (IN of)
         (NP (NN fire)
           (CC and)
           (NN light) ))))
   (. .) ))
```



Viewing Treebanks as Grammars

- The sentences in a treebank can be viewed as a grammar of the language
- We can extract the rules from the parsed sentences
- For example:
  - NP $\rightarrow$ DT JJ NN
  - NP $\rightarrow$ DT JJ NNS
  - NP $\rightarrow$ DT JJ NN NN
  - NP $\rightarrow$ DT JJ CD NNS
  - NP $\rightarrow$ RB DT JJ NN NN
  - NP $\rightarrow$ RB DT JJ JJ NNS
  - NP $\rightarrow$ DT JJ JJ NNP NNS

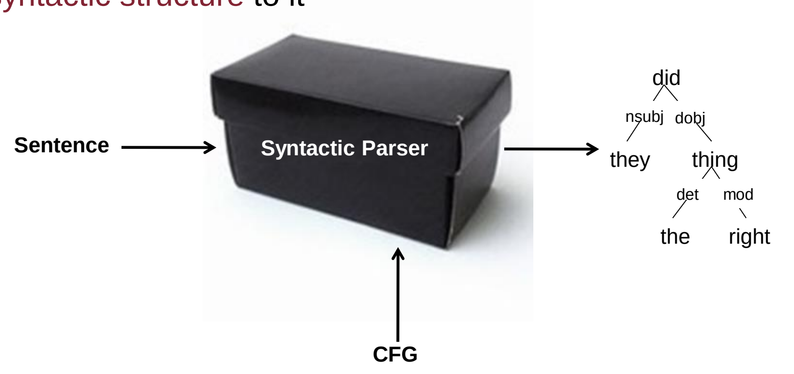
Cardinal number

Adverb

Proper noun, sing.

Syntactic Parsing

- The task of recognizing a sentence and assigning a syntactic structure to it



- However: CFGs do not specify how to calculate the parse tree for a sentence

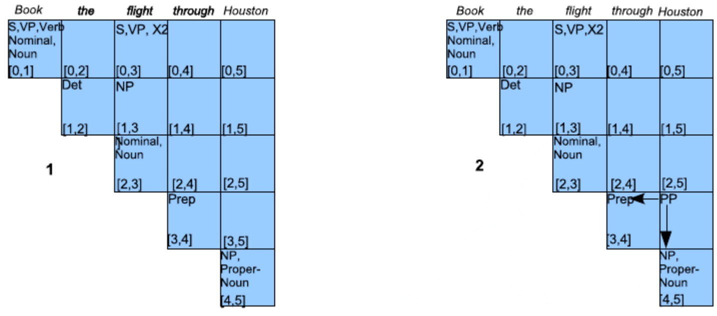
The Cocke-Kasami-Younger (CKY) Algorithm

- A bottom-up dynamic programming parsing approach
- Takes as input a CFG in Chomsky Normal Form
- Given a sentence of n words, we need an (n+1)x(n+1) matrix
- Cell (i,j) contains the set of non-terminals that produce all the constituents spanning positions from i to j of the input sentence
- The cell that represents the entire sentence is (0,n)
- Main idea: if a non-terminal A is in (i,j), there is a production A $\rightarrow$ B C, so there must be an intermediate position k with B in (i,k) and C in (k,j)

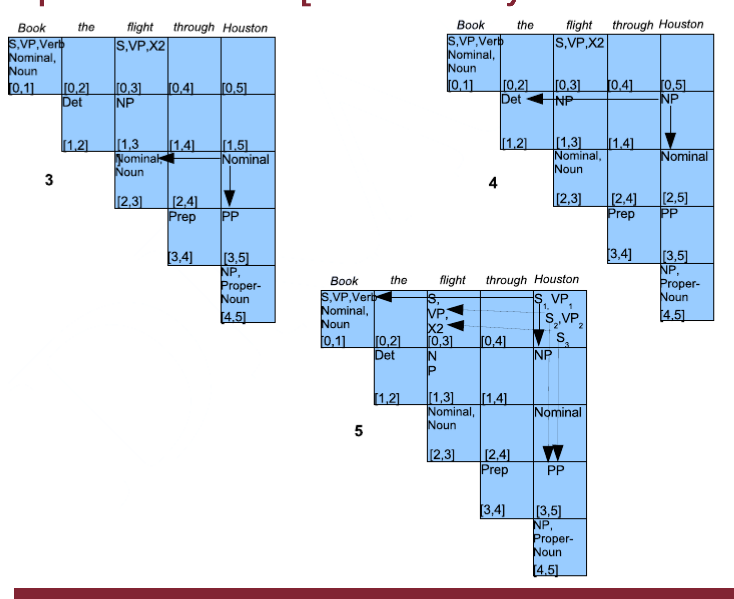
Example of CKY Table [from Jurafsky & Martin book]

Book	*the*	*flight*	*through*	*Houston*
S,VP,Verb Nominal, Noun [0,1]	[0,2]	S,VP,X2 [0,3]	[0,4]	S, VP [0,5]
	Det [1,2]	NP [1,3]	[1,4]	NP [1,5]
		Nominal, Noun [2,3]	[2,4]	Nominal [2,5]
			Prep [3,4]	PP [3,5]
				NP, Proper-Noun [4,5]

Example of CKY Table [from Jurafsky & Martin book]



Example of CKY Table [from Jurafsky & Martin book]



Probabilistic (or Stochastic) CFGs

- First proposed by Taylor Booth (1969)
- In a probabilistic CFG G = (N, T, P, S), each production

$$A \rightarrow w\ [p]$$

is assigned a probability p = P(w|A) = P(A $\rightarrow$ w)

- For each left-hand-side non-terminal A, it must hold:

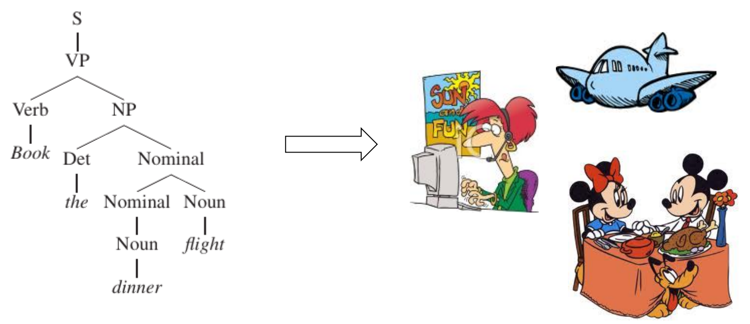
$$\sum_w P(A \rightarrow w) = 1$$

An Example of PCFG (from Jurafsky & Martin)

$S \rightarrow NP\ VP$	[.80]	$Det \rightarrow that$ [.10] | a [.30] | the [.60]	
$S \rightarrow Aux\ NP\ VP$	[.15]	$Noun \rightarrow book$ [.10] | $flight$ [.30]	
$S \rightarrow VP$	[.05]	| $meal$ [.15] | $money$ [.05]	
$NP \rightarrow Pronoun$	[.35]	| $flights$ [.40] | $dinner$ [.10]	
$NP \rightarrow Proper\text{-}Noun$	[.30]	$Verb \rightarrow book$ [.30] | $include$ [.30]	
$NP \rightarrow Det\ Nominal$	[.20]	| $prefer$; [.40]	
$NP \rightarrow Nominal$	[.15]	$Pronoun \rightarrow I$ [.40] | she [.05]	
$Nominal \rightarrow Noun$	[.75]	| me [.15] | you [.40]	
$Nominal \rightarrow Nominal\ Noun$	[.20]	$Proper\text{-}Noun \rightarrow Houston$ [.60]	
$Nominal \rightarrow Nominal\ PP$	[.05]	| TWA [.40]	
$VP \rightarrow Verb$	[.35]	$Aux \rightarrow does$ [.60] | can [40]	
$VP \rightarrow Verb\ NP$	[.20]	$Preposition \rightarrow from$ [.30] | to [.30]	
$VP \rightarrow Verb\ NP\ PP$	[.10]	| on [.20] | $near$ [.15]	
$VP \rightarrow Verb\ PP$	[.15]	| $through$ [.05]	
$VP \rightarrow Verb\ NP\ NP$	[.05]		
$VP \rightarrow VP\ PP$	[.15]		
$PP \rightarrow Preposition\ NP$	[1.0]		

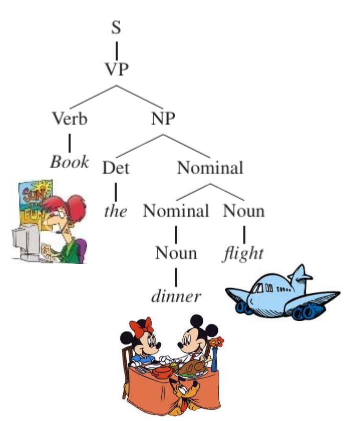
Meaning, meaning, meaning!

- We are now moving from syntax to semantics



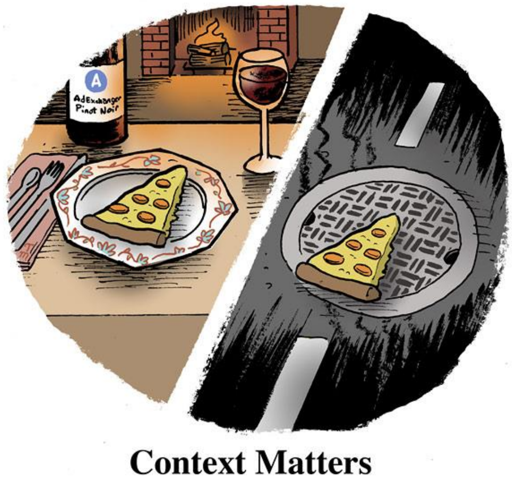
Meaning, meaning, meaning!

- We are now moving from syntax to semantics



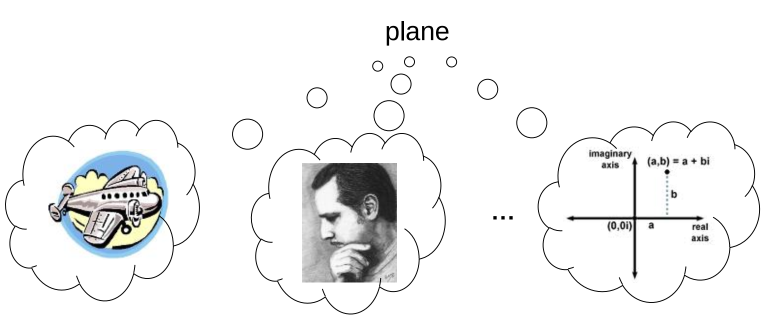
Word Senses

- The meaning of a word depends on the context in which it occurs



Word Senses

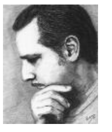
- The meaning of a word depends on the context in which it occurs
- We call each meaning of a word a word sense



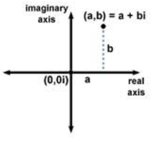
Word Senses in Context

- I am catching the earliest **plane** to Brussels.

- This area probably lies more on the spiritual **plane** than the mental one.



- Let's represent three-dimensional structures on a two-dimensional **plane**

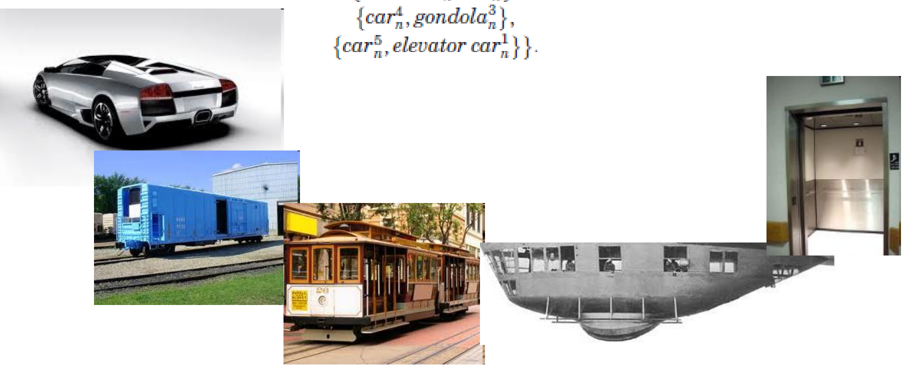


WordNet [Miller et al. 1990]

- The most popular computational lexicon of English
  - Based on psycholinguistic theories
- Concepts expressed as sets of synonyms (synsets)
  - { car_n^1, auto_n^1, automobile_n^1, machine_n^4, motorcar_n^1 }
- A word sense is a word occurring in a synset
  - $\mathbf{machine_n^4}$ is the fourth sense of noun machine

WordNet: the "car" example

$$
\begin{aligned}
Senses_{WN}(car_n) = \{ & \{car^1_n, auto^1_n, automobile^1_n, machine^4_n, motorcar^1_n\},\\
& \{car^2_n, rail\ car^1_n, rail\ way\ car^1_n, rail\ road\ car^1_n\},\\
& \{cable\ car^1_n, car^3_n\},\\
& \{car^4_n, gondola^3_n\},\\
& \{car^5_n, elevator\ car^1_n\}\}.
\end{aligned}
$$



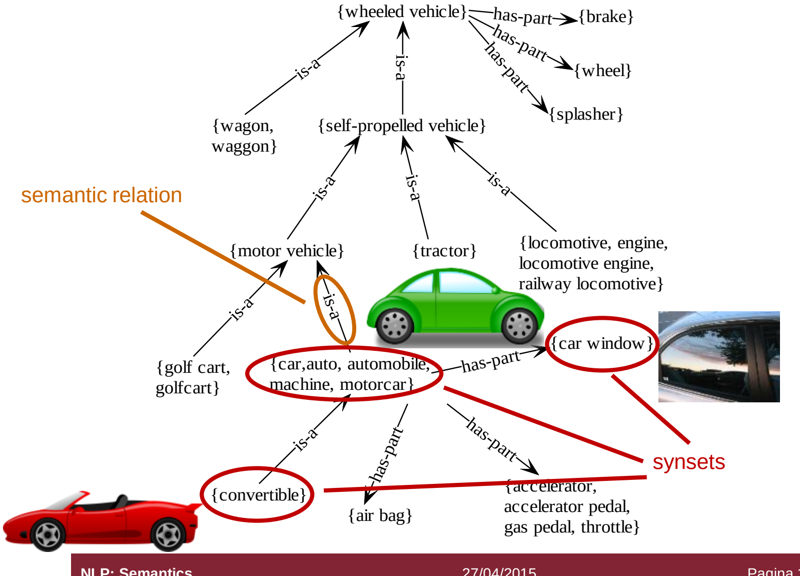
WordNet provides textual definitions

- Called **glosses**
- A textual definition is provided for each synset
- Gloss of car_n^1: 
  - "a 4-wheeled motor vehicle; usually propelled by an internal combustion engine; 'he needs a car to get to work' "
- Gloss of car_n^2: 
  - "a wheeled vehicle adapted to the rails of railroad; 'three cars had jumped the rails' "
- Also available in quasi-logical form

WordNet encodes relations!

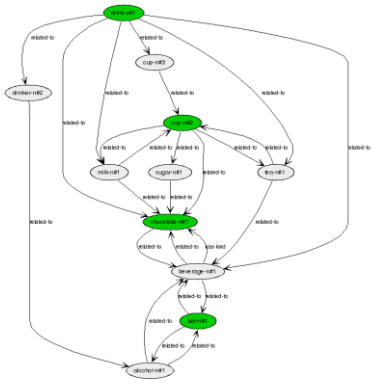
- Semantic relations between synsets
  - **Hypernymy** (car_n^1 is-a motor $vehicle_n^1$)
  - **Meronymy** (car_n^1 has-a car $door_n^1$)
  - **Entailment**, **similarity**, **attribute**, etc.
- Lexical relations between word senses
  - **Synonymy** (i.e., words that belong to the same synset)
  - **Antonymy** ($good_a^1$ antonym of bad_a^1)
  - **Pertainymy** ($dental_a^1$ pertains to $tooth_n^1$)
  - **Nominalization** ($service_n^2$ nominalizes $serve_v^4$)

WordNet as a Graph



But WordNet is more than Simply a Graph!

- It is a semantic network!
- A semantic network is a network which represents semantic relations among concepts
- It is often used as a form of knowledge representation



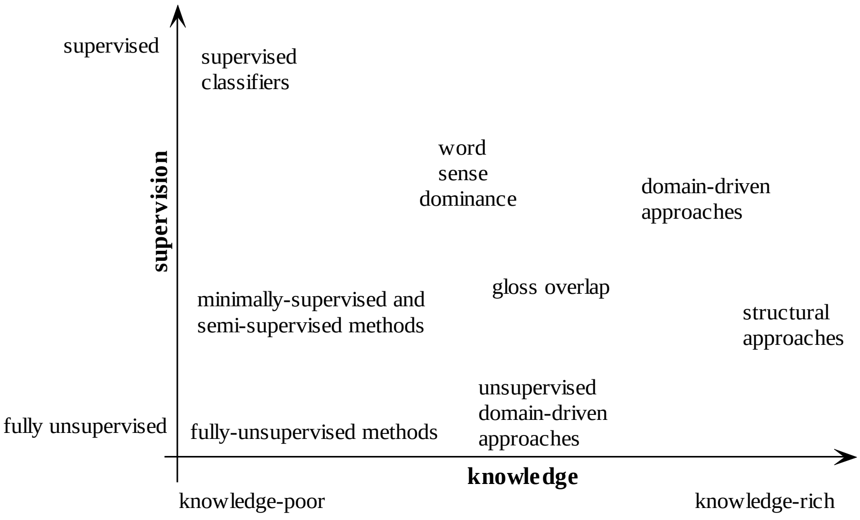
Word Sense Disambiguation (WSD)

- WSD is the task of computationally determining which sense of a word is activated by its use in a particular context [Ide and Véronis, 1998; Navigli, 2009]
- It is basically a classification task
  - The objective is to learn how to classify words into word senses
  - This task is strongly tied to Machine Learning

I drank a cup of chocolate at the **bar**

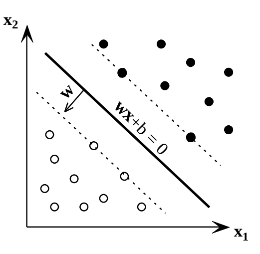
bar =

Supervision and Knowledge

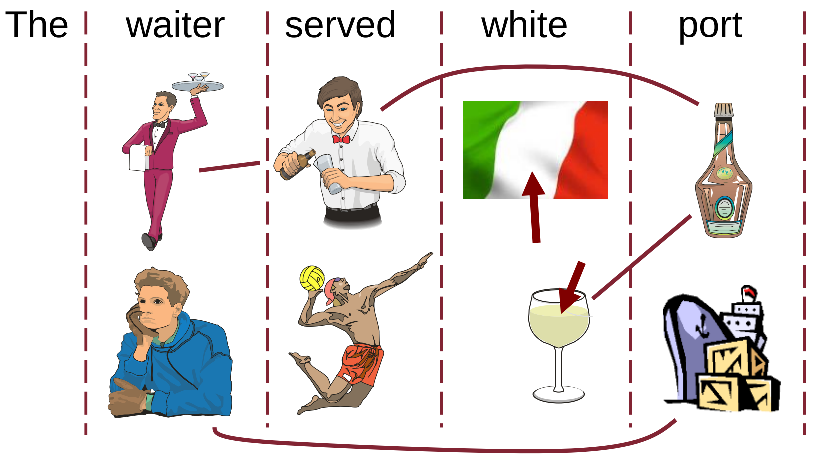


Supervised WSD: Support Vector Machines

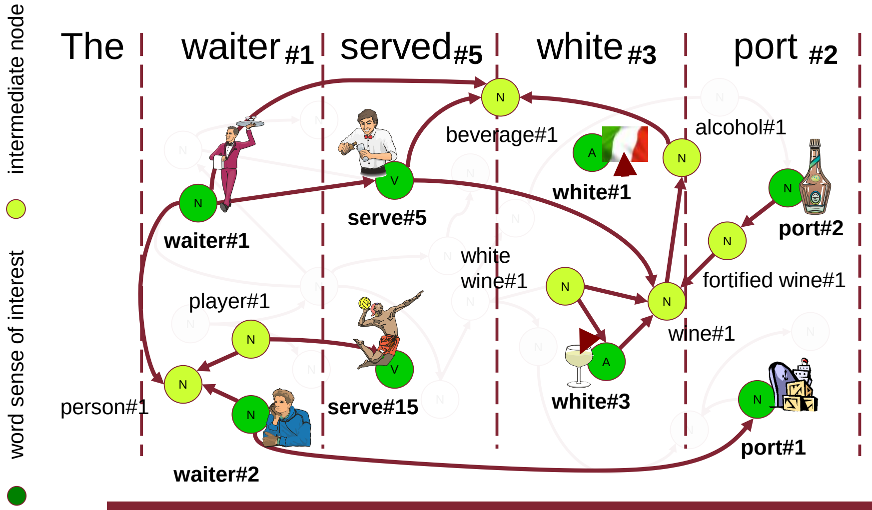
- SVM learns a linear hyperplane from the training set that separates positive from negative examples
- The hyperplane maximizes the distance to the closest positive and negative examples (**support vectors**)
- Achieves state-of-the-art performance in WSD [Keok and Ng, 2002]



Knowledge-based Word Sense Disambiguation

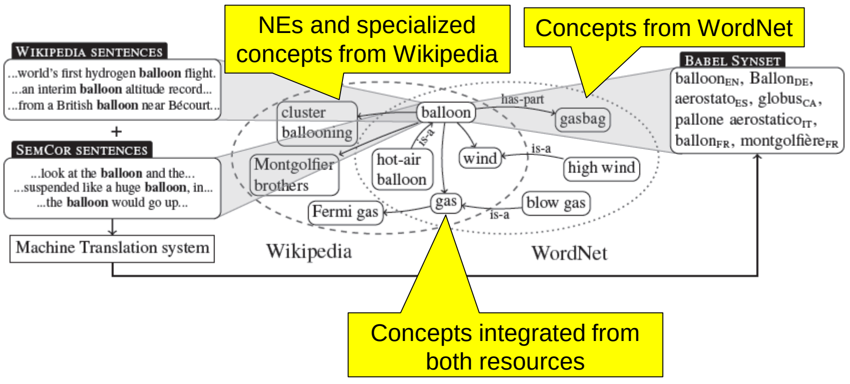


Knowledge-based Word Sense Disambiguation



BabelNet [Navigli and Ponzetto, AIJ 2012]

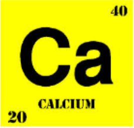
- A wide-coverage multilingual semantic network including both encyclopedic (from Wikipedia) and lexicographic (from WordNet) entries



BabelNet as a Multilingual Inventory for:

Concepts

Calcio in Italian can denote different concepts:



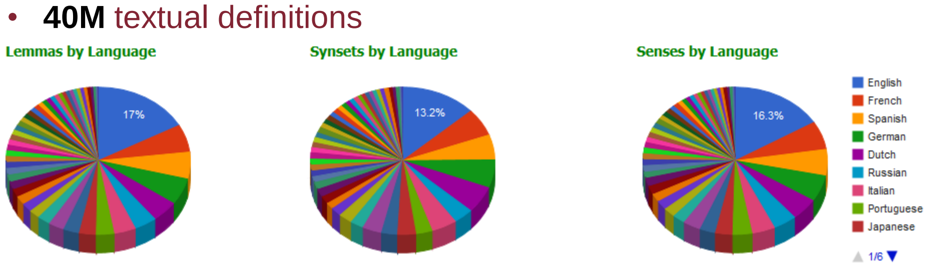
Named Entities

The word *Mario* can be used to represent different things such as the video game charachter or a soccer player (Gomez) or even a music album

BabelNet 3.0 is online: http://babelnet.org

Anatomy of BabelNet 3.0

- **271** languages covered (including Latin!)
- **13.8M** Babel synsets
  - (6.4M concepts and 7.4M named entities)
- **117M** word senses
- **355M** semantic relations (26 edges per synset on avg.)
- **11M** synset-associated images
- **40M** textual definitions



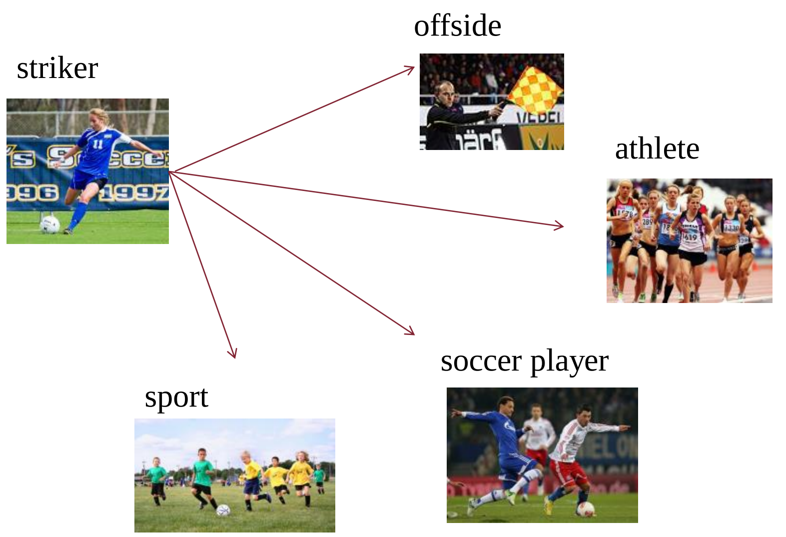
New 3.0 version out!

- Seamless integration of:
  - **WordNet** 3.0
  - **Wikipedia**
  - **Wikidata**
  - **Wiktionary**
  - **OmegaWiki**
  - **Open Multilingual WordNet** [Bond and Foster, 2013]
- Translations for all open-class parts of speech
- **2B** RDF triples available via SPARQL endpoint

So what?

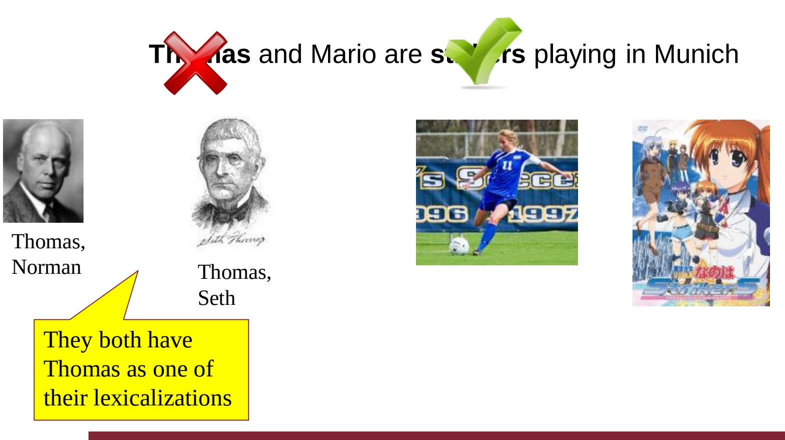
Babelfy

Step 1: Semantic Signatures



Step 2: Find all possible meanings of words

1. Exact Matching (good for WSD, bad for EL)



Step 2: Find all possible meanings of words

2. Partial Matching (good for EL)

T **as** and Mario are **s** **rs** playing in Munich

Thomas, Norman

Thomas, Seth

Thomas Müller

It has Thomas as a substring of one of its lexicalizations

Step 2: Find all possible meanings of words

- "Thomas and Mario are strikers playing in Munich"

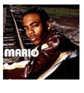
Seth Thomas

Thomas Müller

Thomas (novel)

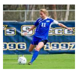
Mario (Character)

Mario (Album)



Mario Gómez

striker (Sport)



Striker (Video Game)

Striker (Movie)

Munich (City)

FC Bayern Munich

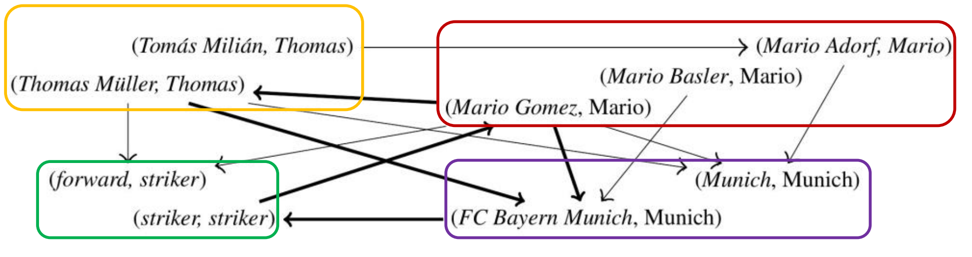
Munich (Song)

Step 2: Find all possible meanings of words

- "Thomas and Mario are strikers playing in Munich"

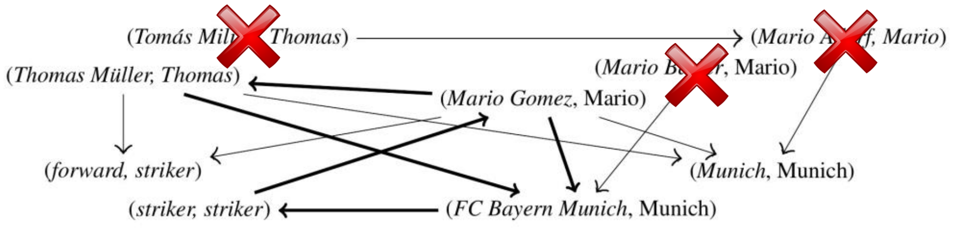
Step 3: Connect all the candidate meanings

- Thomas and Mario are strikers playing in Munich



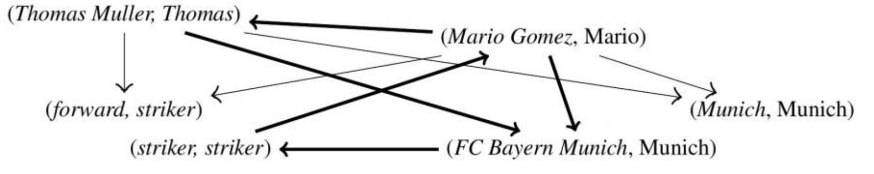
Step 4: Extract a dense subgraph

- **Thomas** and **Mario** are **strikers** playing in **Munich**



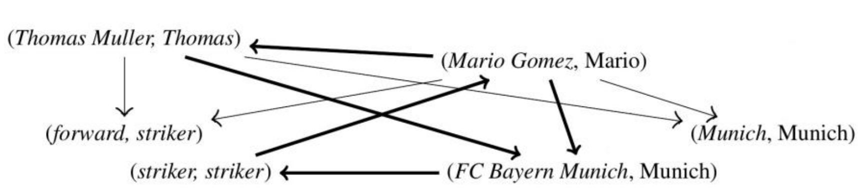
Step 4: Extract a dense subgraph

- **Thomas** and **Mario** are **strikers** playing in **Munich**



Step 5: Select the most reliable meanings

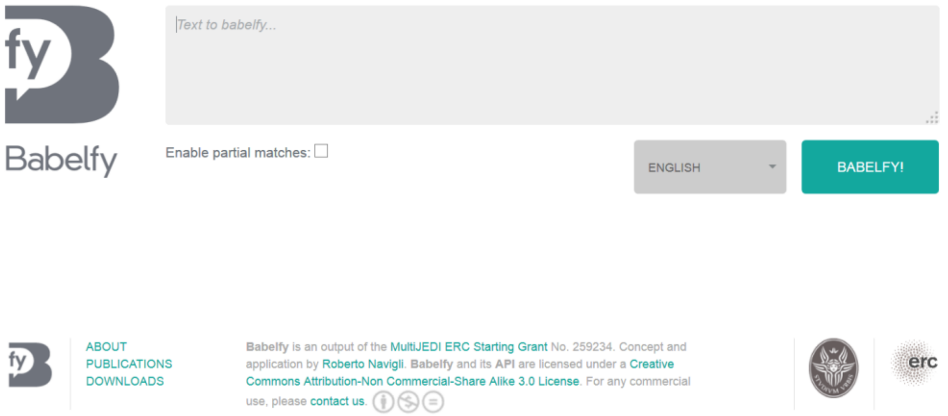
- **Thomas** and **Mario** are **strikers** playing in **Munich**



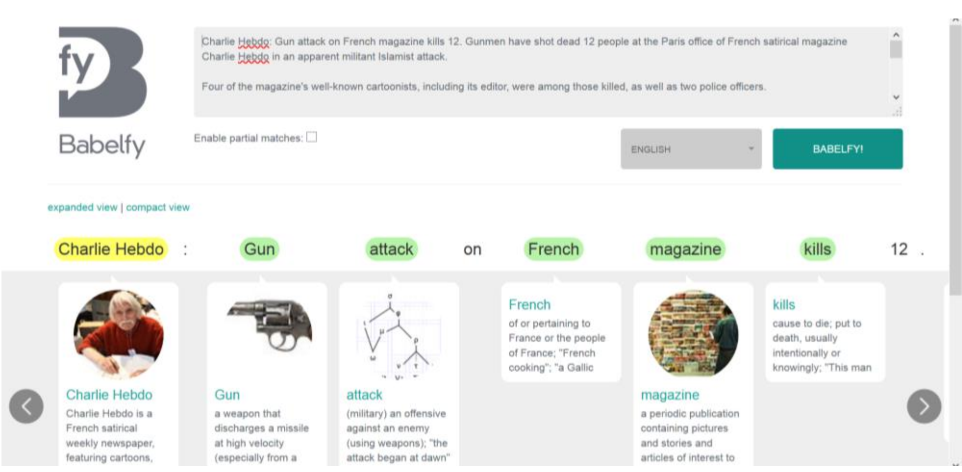
Step 5: Select the most reliable meanings

- “Thomas and Mario are strikers playing in Munich”

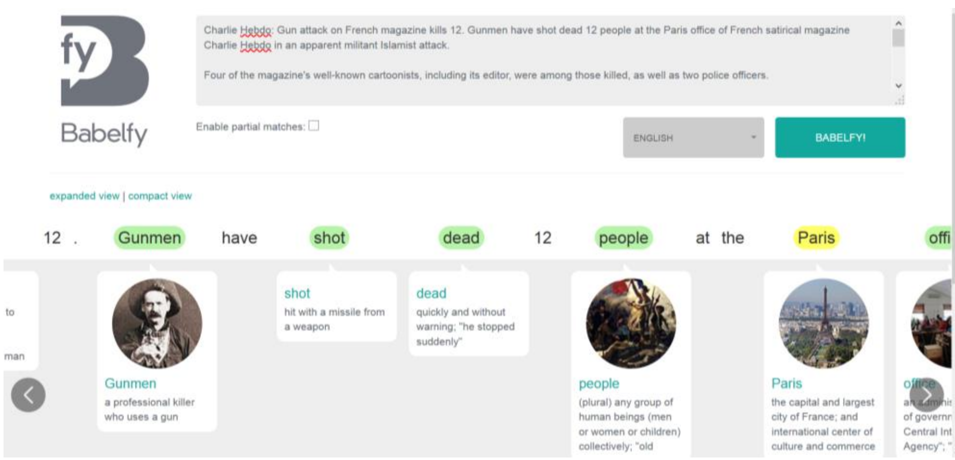
http://babelfy.org



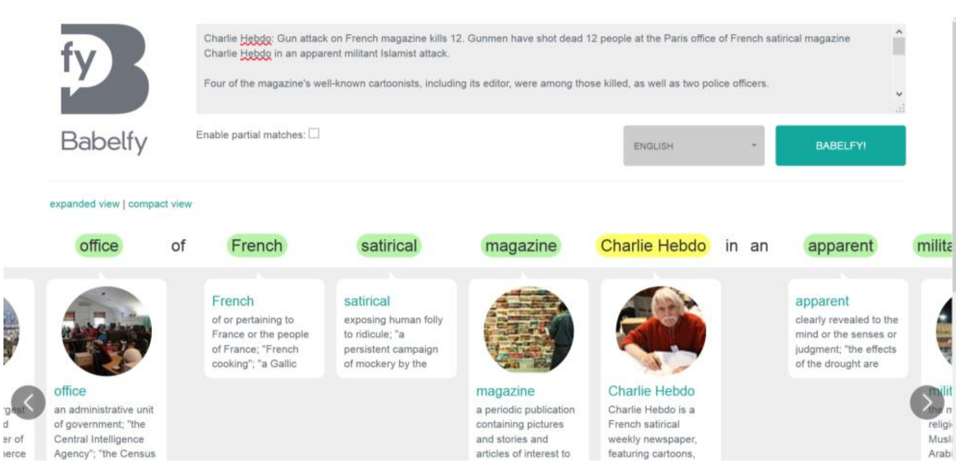
The Charlie Hebdo gun attack (English)



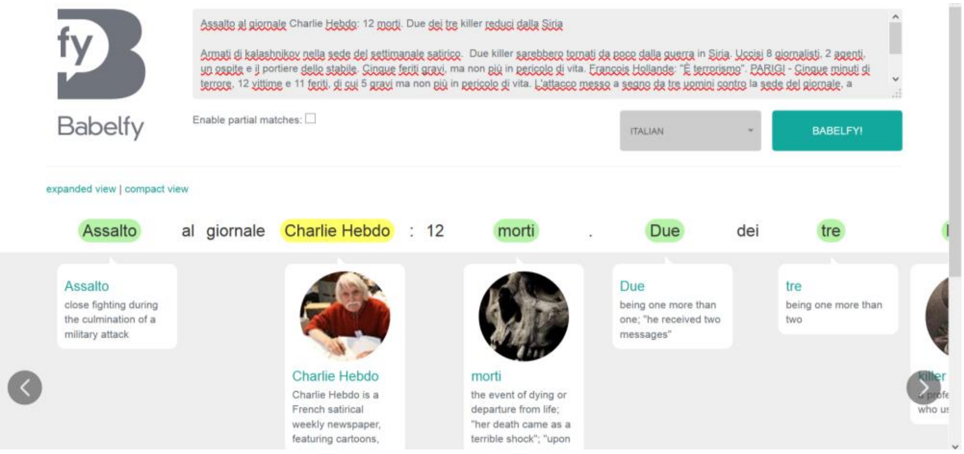
The Charlie Hebdo gun attack (English)



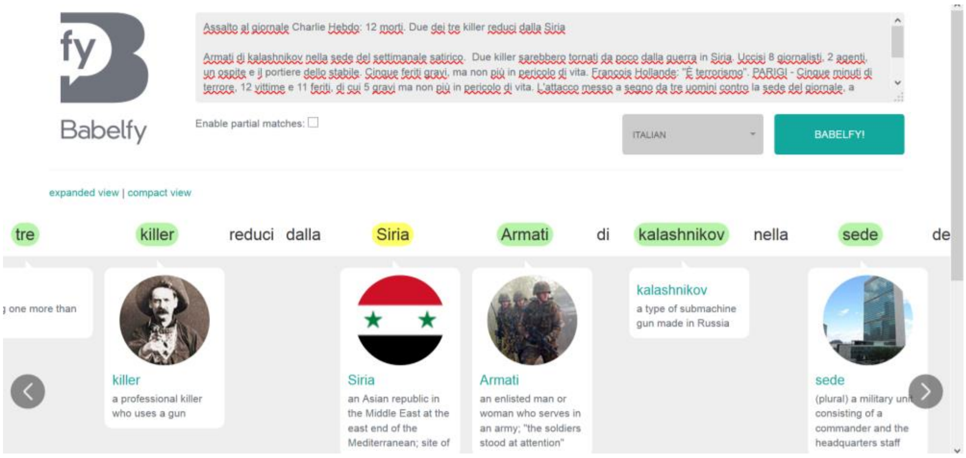
The Charlie Hebdo gun attack (English)



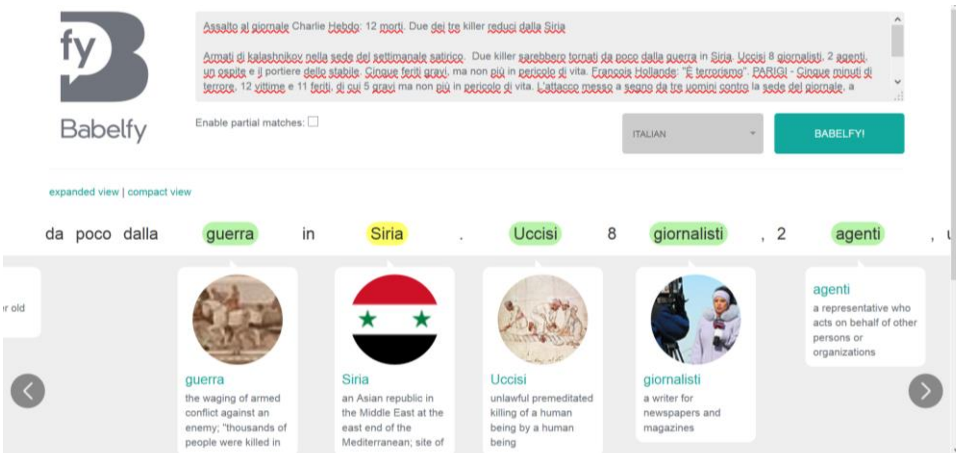
The Charlie Hebdo gun attack (Italian)



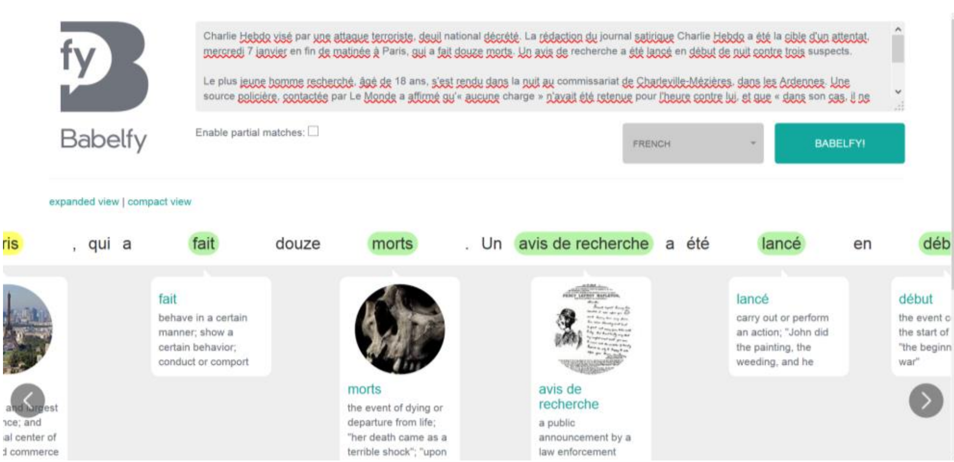
The Charlie Hebdo gun attack (Italian)



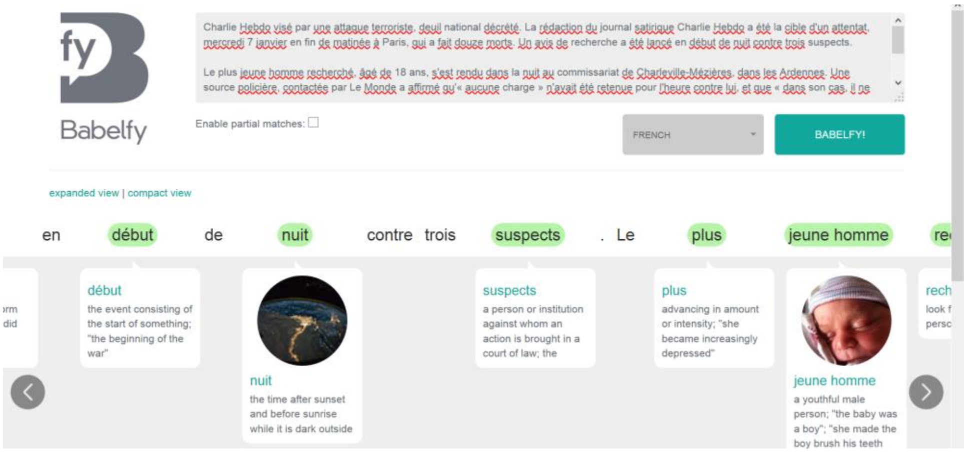
The Charlie Hebdo gun attack (Italian)



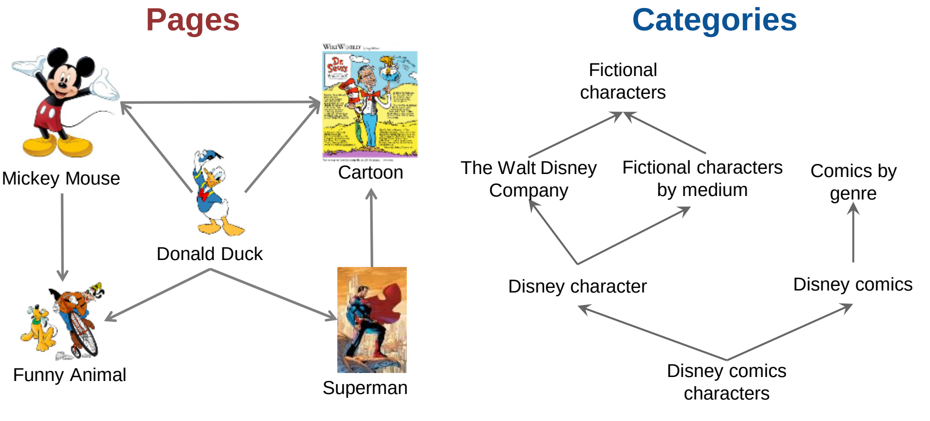
The Charlie Hebdo gun attack (French)



The Charlie Hebdo gun attack (French)

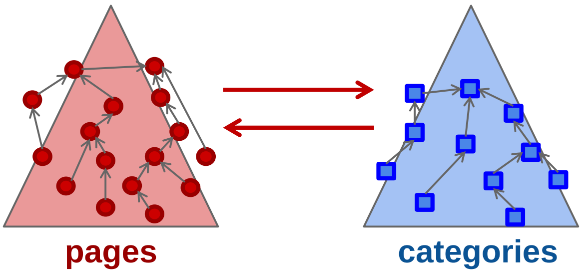


The Wikipedia structure: an example



Our goal

To **automatically** create a **Wi**kipedia **Bi**taxonomy for Wikipedia pages and categories in a simultaneous fashion.



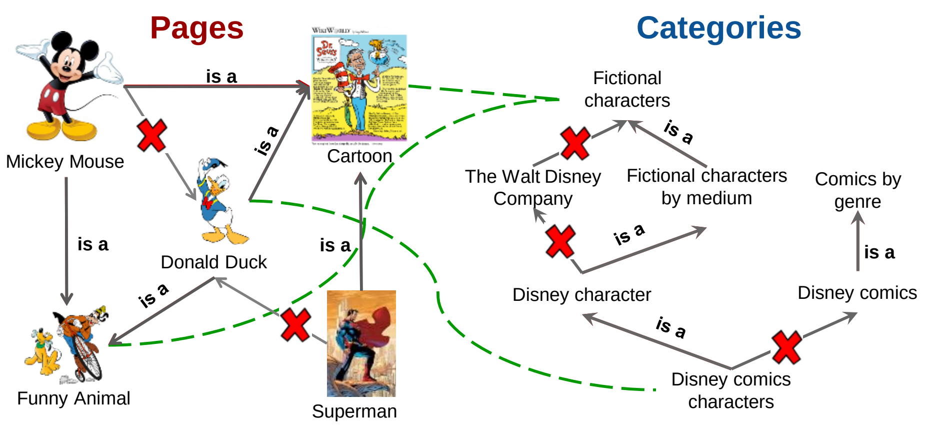
Our goal

To **automatically** create a **Wi**kipedia **Bi**taxonomy for Wikipedia pages and categories in a simultaneous fashion.

KEY IDEA

The page and category level are **mutually beneficial** for inducing a wide-coverage and fine-grained integrated taxonomy

The Wikipedia Bitaxonomy: an example



The WiBi Page taxonomy

Assumption

- The first sentence of a page is a good definition (also called gloss)

Scrooge McDuck

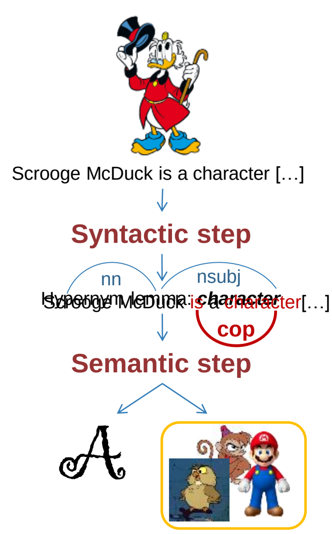
From Wikipedia, the free encyclopedia

Scrooge McDuck is a cartoon character created in 1947 by Carl Barks and licensed by The Walt Disney Company. Scrooge is an elderly Scottish anthropomorphic white duck with a yellow-orange bill, legs, and feet.

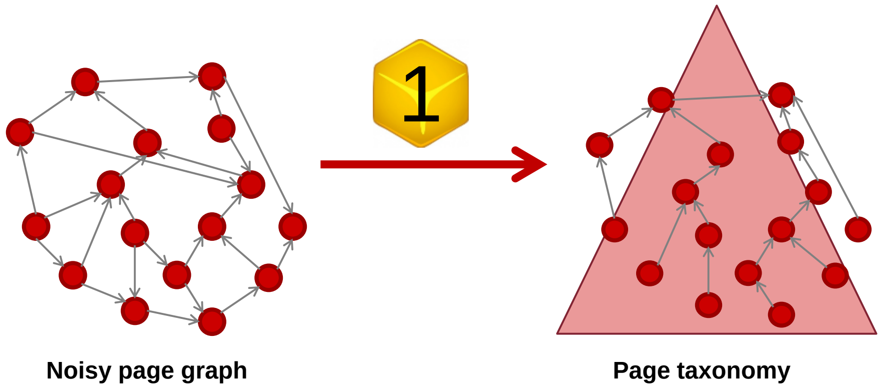
The WiBi Page taxonomy

1. **[Syntactic step]**
   Extract the hypernym lemma from a page definition using a syntactic parser;

2. **[Semantic step]**
   Apply a set of linking heuristics to disambiguate the extracted lemma.



The story so far



The Bitaxonomy algorithm

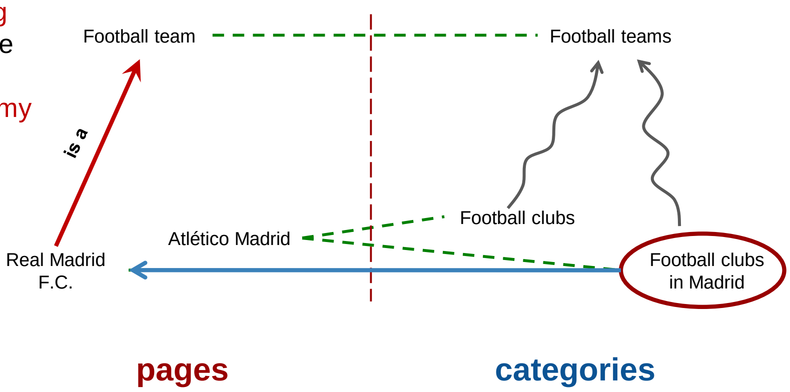
The Bitaxonomy algorithm

The information available in the two taxonomies is mutually beneficial

- At each step exploit one taxonomy to update the other and vice versa
- Repeat until convergence

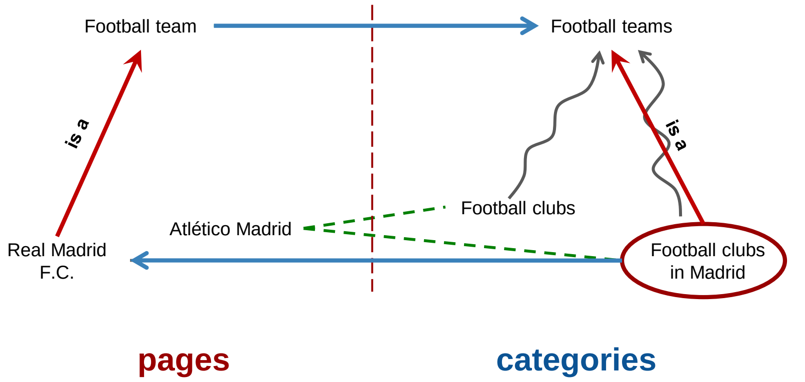
The Bitaxonomy algorithm

Starting from the page taxonomy

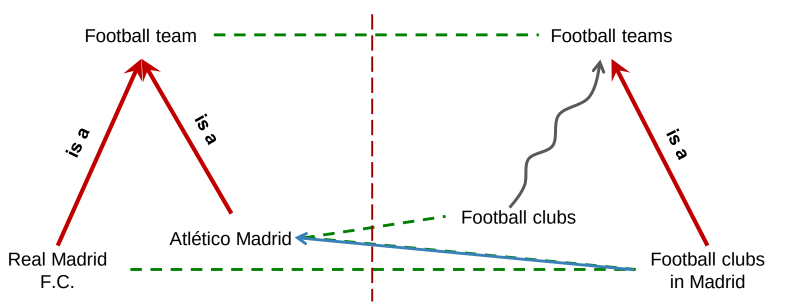


The Bitaxonomy algorithm

Exploit the cross links to infer hypernym relations in the category taxonomy



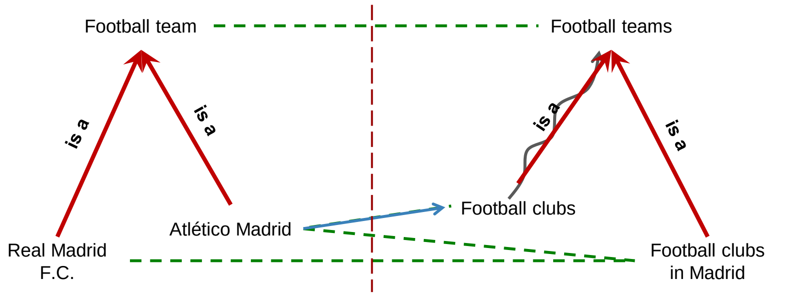
The Bitaxonomy algorithm



Take advantage of cross links to infer back is-a relations in the page taxonomy

pages **categories**

The Bitaxonomy algorithm

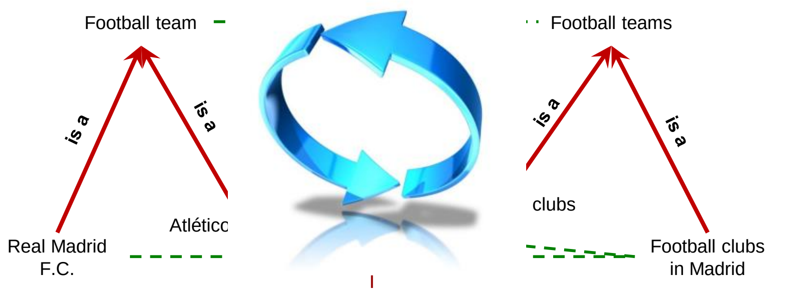


Use the relations found in previous step to infer new hypernym edges

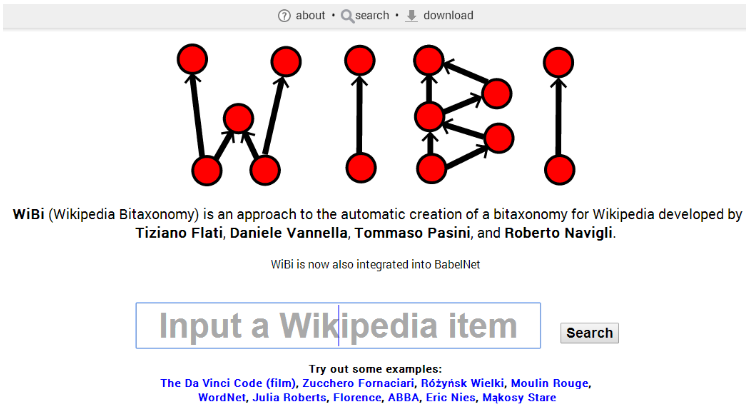
pages **categories**

The Bitaxonomy algorithm

Mutual enrichment of both taxonomies until convergence

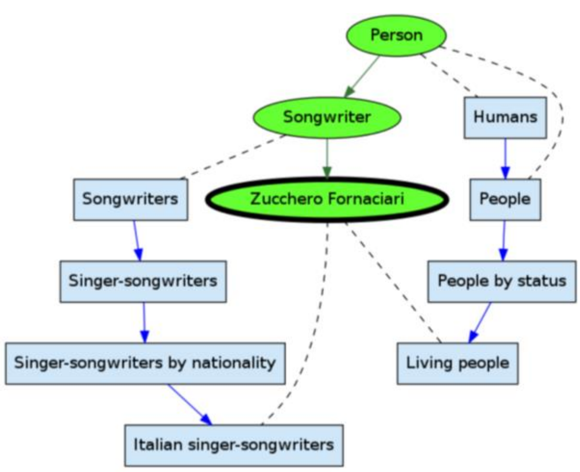


pages **categories**

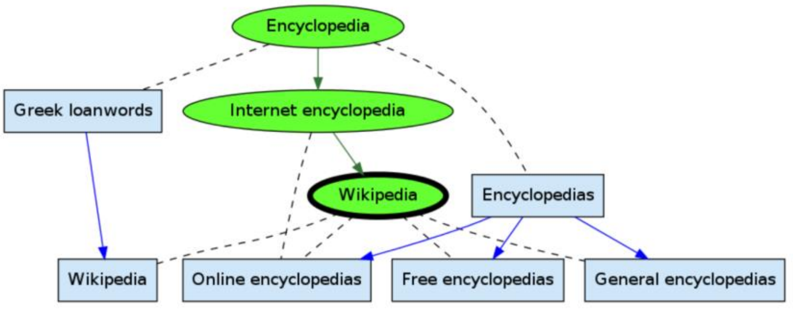


wibitaxonomy.org

Example from http://wibitaxonomy.org: WordNet

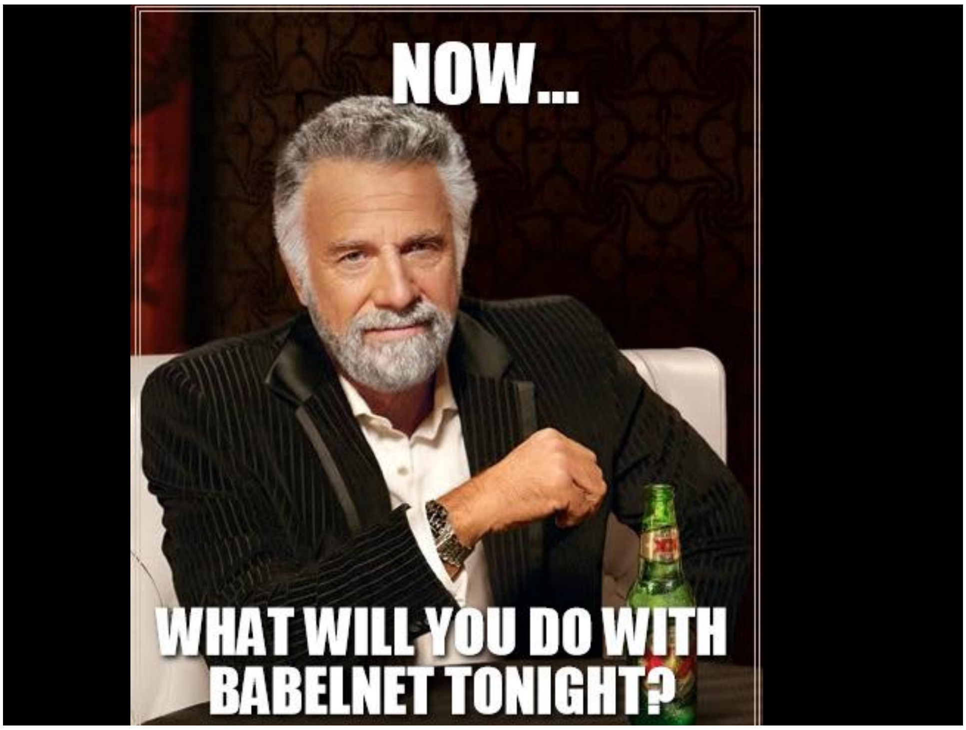


Example from http://wibitaxonomy.org: Wikipedia



THAT'S ALL FOLKS!!!

(Isn't it enough???)



Thanks or…

merci

(grazie)

Google
Focused Research Awards

SAPIENZA
UNIVERSITÀ DI ROMA

Roberto Navigli

Linguistic Computing Laboratory
http://lcl.uniroma1.it

Linguistic Computing Laboratory