FH OBERÖSTERREICH
University of Applied Sciences

Web-Semantik-Technologien

Vorlesung WS 2019/20

thomas.kern@fh-hagenberg.at

Information Extraction

Outline

- IR versus IE
- IE-Subtasks
- Architecture of an IE-System
- NER
- NER-Pipeline
- Sequence Labeling as a Classification Task
- Features
- Evaluation & Review

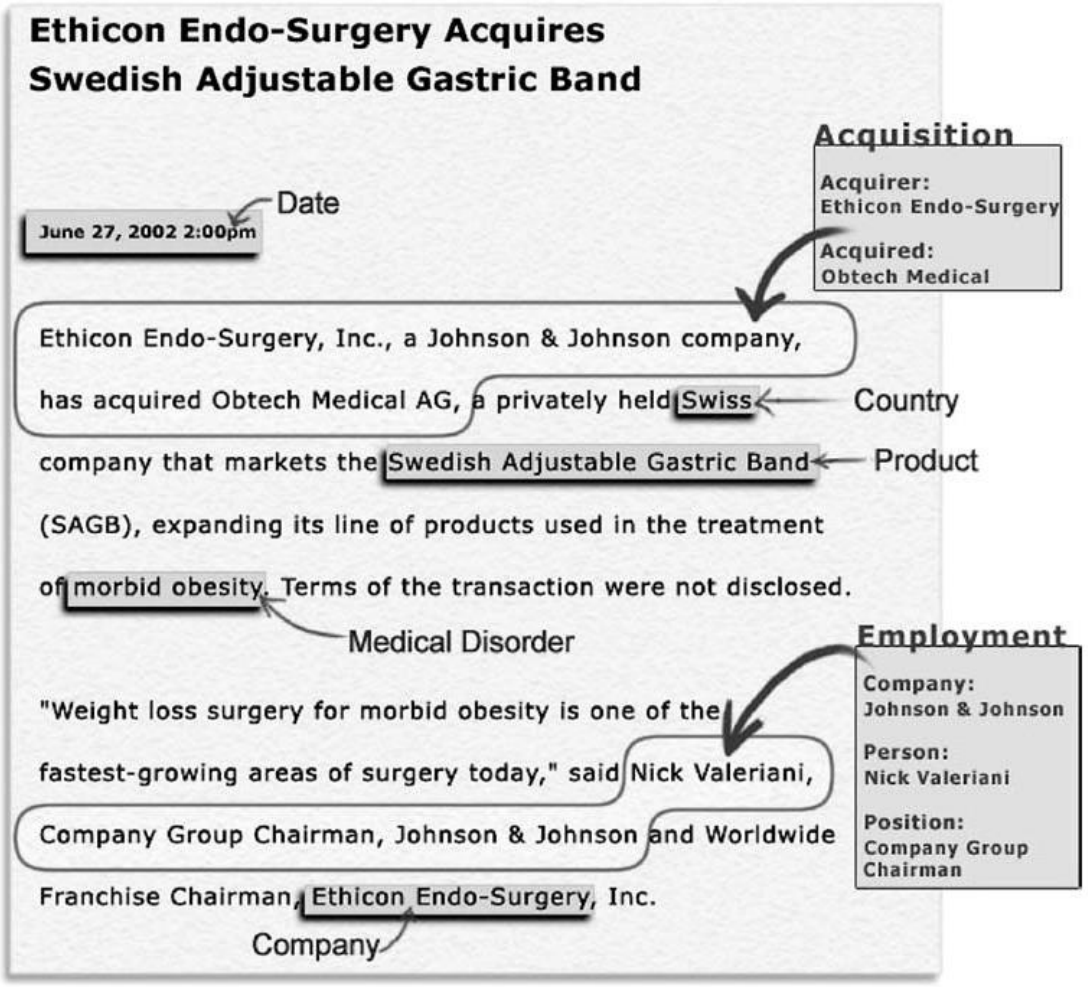
Information Extraction

- **Information retrieval (IR)**
  - IR treats documents as “bag of words”
    - order of words has no significance
    - semantic meaning of words not relevant
- **Information extraction (IE)**
  - **Identify specific pieces of information** (data) in a unstructured or semi-structured textual document.
  - **Transform** unstructured information of a corpus of documents or web pages **into a structured database**.
  - processing human language documents by means of **natural language processing** (NLP)
  - current approaches to IE focus on narrowly restricted domains: sports, medicine, business

Elements to be extracted from text

- **Entities.** Entities are the basic building blocks that can be found in text documents. Examples include people, dates, geographical locations, companies, desease and drugs.
- **Attributes.** Attributes are features of the extracted entities. Some examples of attributes are the title of a person, the age of a person, and the type of an organization.
- **Facts.** Facts are the relations that exist between entities. Some examples are an employment relationship between a person and a company (emplyoment) or a relationship between persons (marriage).
- **Events**: An event is an activity or occurrence of interest in which entities participate such as a terrorist act, a merger between two companies, a birthday etc.

Example of a tagged news article



Ethicon Endo-Surgery Acquires Swedish Adjustable Gastric Band

June 27, 2002 2:00pm

Ethicon Endo-Surgery, Inc., a Johnson & Johnson company, has acquired Obtech Medical AG, a privately held Swiss company that markets the Swedish Adjustable Gastric Band (SAGB), expanding its line of products used in the treatment of morbid obesity. Terms of the transaction were not disclosed.

"Weight loss surgery for morbid obesity is one of the fastest-growing areas of surgery today," said Nick Valeriani, Company Group Chairman, Johnson & Johnson and Worldwide Franchise Chairman, Ethicon Endo-Surgery, Inc.

Information Extraction Subtasks

- **Named Entity Recognition Task (NE):**
  - basic task of any IE system
  - Mark in the text each string that represents:
    - a person, organization, or location name, or a date or time, or monetary amounts of currency or percentage figure
    - Example:

“It's a chance to think about new questions", said Ms. <enamex type= "PERSON">Simpson</enamex>, a partner in the <enamex type= "ORGANIZATION"> McSimpson & Sons</enamex> company in <enamex type= "LOCATION">San Francisco</enamex>, <enamex type= "LOCATION">California</enamex>

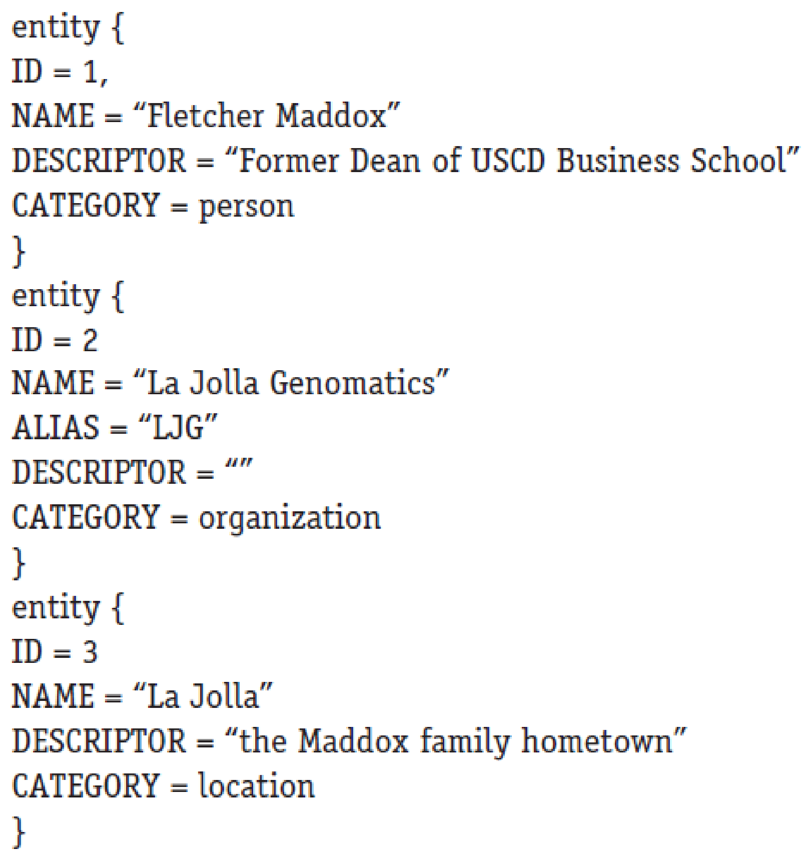
- **Template Element Task (TE)**
  - Each TE consists of a **generic object and some attributes** that describe it
  - Extract basic information related to organization, person, and artifact entities, drawing evidence **from everywhere** in the text.
  - Template element tasks (TEs) are independent or neutral with respect to domain.

Template Task (TE) - Example

- **Text**

  Fletcher Maddox, former Dean of the UCSD Business School, announced the formation of La Jolla Genomatics together with his two sons. La Jolla Genomatics will release its product Geninfo in June 1999. L.J.G. is headquartered in the Maddox family's hometown of La Jolla, CA.

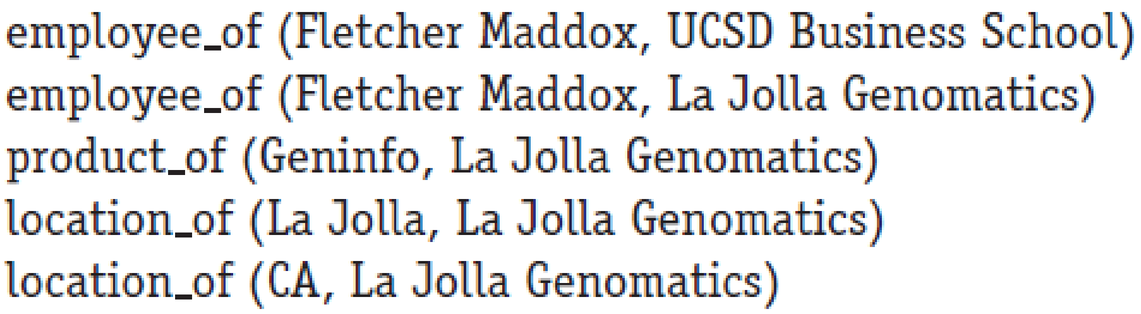
- **Templates**



```
entity {
ID = 1,
NAME = "Fletcher Maddox"
DESCRIPTOR = "Former Dean of USCD Business School"
CATEGORY = person
}
entity {
ID = 2
NAME = "La Jolla Genomatics"
ALIAS = "LJG"
DESCRIPTOR = ""
CATEGORY = organization
}
entity {
ID = 3
NAME = "La Jolla"
DESCRIPTOR = "the Maddox family hometown"
CATEGORY = location
}
```

Information Extraction Subtasks

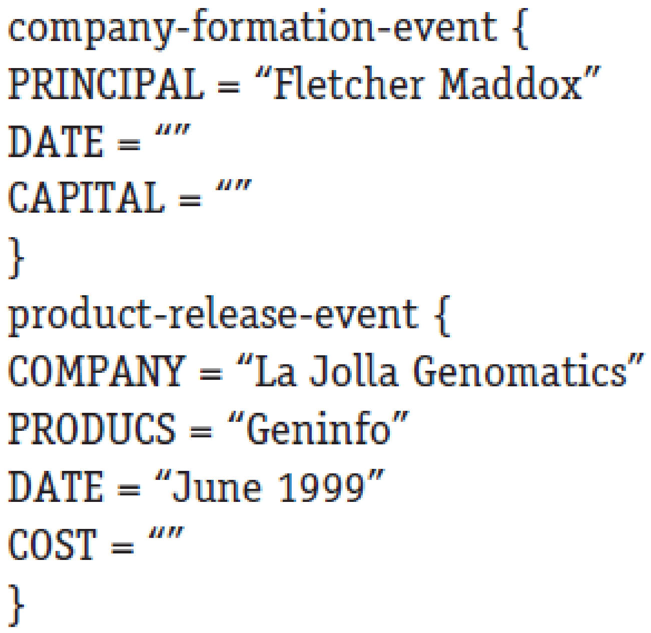
- **Template Relation task (TR):**
  - Goal of the TR is to **find the relationships** that exist between template elements
  - Extract relational information on *employee_of*, *manufacture_of*, *location_of* relations etc.
  - expresses a domain-independent relationship between entities. (TE just identifies entities themselves)
  - Examples (TR) extracted from the sample text:



employee_of (Fletcher Maddox, UCSD Business School)
employee_of (Fletcher Maddox, La Jolla Genomatics)
product_of (Geninfo, La Jolla Genomatics)
location_of (La Jolla, La Jolla Genomatics)
location_of (CA, La Jolla Genomatics)

Information Extraction Subtasks

- **Scenario Template task (ST):**
  - Extract specified **event information** and relate the event information to particular organization, person, or artifact entities (ST identifies domain and task specific entities and relations)



```
company-formation-event {
PRINCIPAL = "Fletcher Maddox"
DATE = ""
CAPITAL = ""
}
product-release-event {
COMPANY = "La Jolla Genomatics"
PRODUCS = "Geninfo"
DATE = "June 1999"
COST = ""
}
```

Information Extraction Subtasks

- **Coreference Task (CO):**
  - The coreference task (CO) captures information on **coreferring expressions** (e.g., pronouns or any other mentions of a given entity), including those tagged in the NE, TE tasks. This CO focuses on the IDENTITY (IDENT) relation, which is symmetrical and transitive.
  - Example

    David$_1$ came home from school, and saw his$_1$ mother$_2$ Rachel$_2$. She$_2$ told him$_1$ that his$_1$ father will be late.

    Correctly identified pronominal coreference chains are (David$_1$, his$_1$, him$_1$, his$_1$) and (mother$_2$, Rachel$_2$, She$_2$).

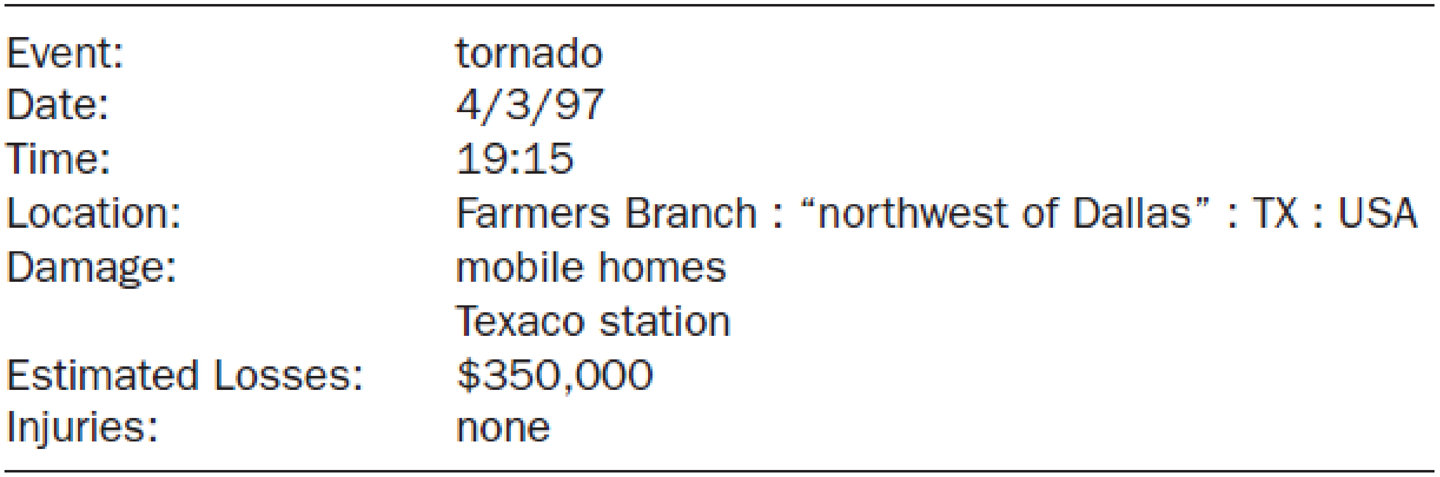
IE - Examples

- Domain specific IE-Systems:
  - Business events from news articles
  - Natural Disaster Events
  - Terror-Related Events
  - Technology-Related Evens
  - …….

IE-Examples: Natural Disaster Events

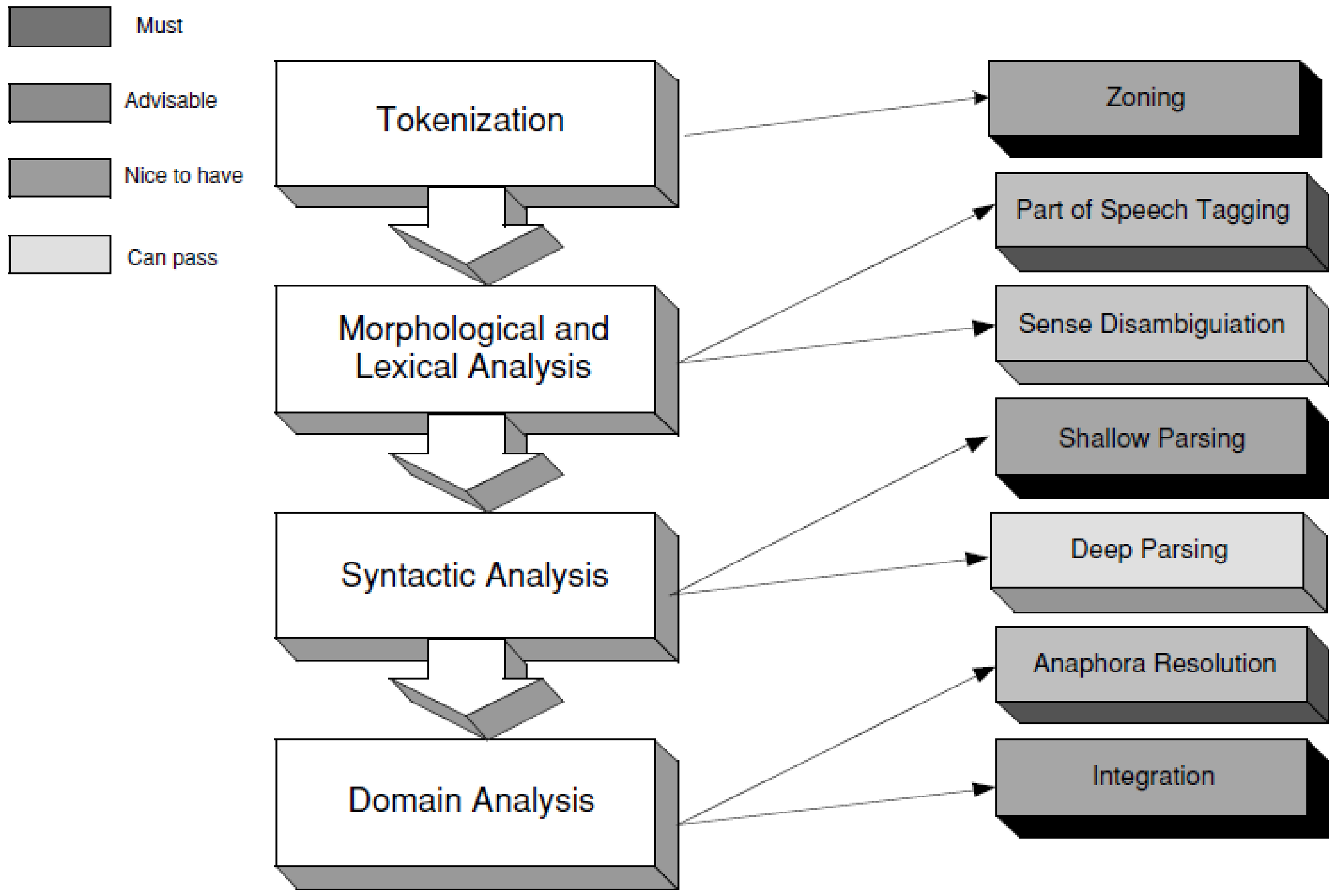
Text:

4 Apr Dallas – Early last evening, a tornado swept through an area northwest of Dallas, causing extensive damage. Witnesses confirm that the twister occurred without warning at approximately 7:15 p.m. and destroyed the mobile homes. The Texaco station, at 102 Main Street, Farmers Branch, TX, was severely damaged, but no injuries were reported. Total property damages are estimated at $350,000.



Event:	tornado
Date:	4/3/97
Time:	19:15
Location:	Farmers Branch : “northwest of Dallas” : TX : USA
Damage:	mobile homes
	Texaco station
Estimated Losses:	$350,000
Injuries:	none

Architecture of a information extraction system



Components of an information extraction system

- **Tokenization or Zoning**
  - Splits an input document into its basic building blocks.
  - Typical building blocks are words, sentences and paragraphs.
  - Rarely used higher building blocks like sections and chapters
- **Morphological and lexical analysis**
  - handles activities such as
    - assigning **POS (Part of speech) tags** to the document's various words
    - parts of speech: assign to each word if it is a noun, verb, adjective, etc
  - **disambiguating** the sense of ambiguous words
    - Example: flies (noun or verb ?)
- **Syntactic analysis**
  - establishes the **connection** between the different parts of each sentence.
  - Done either by deep parsing or shallow parsing

Components of an information extraction system

- **Domain analysis:**
  - combines all information collected from the previous components and creates complete frames to describe relationships between entities
  - Advanced domain analysis modules also possess an **anaphora resolution** (CO) component.
  - *Anaphora resolution (Coreference Resolution):*
    - concerns itself with resolving indirect (and usually pronomic) references for entities that may appear in sentences other than the one containing the primary direct reference to an entity
    - Example:
      - *We gave the bananas to the monkeys because they were hungry*
      - ***Anaphor:*** *"they" refers to "monkeys"*
- **Integration:**
  - Merge results with other IE systems (Ontologies)

NER – Named Entity Recognition

- NER involves *identification* of *proper names* in texts, and *classification* into a set of predefined categories of interest.
- Three universally accepted categories: person, location and organisation
  - Example:

Person:	Albert Einstein
Organization:	FH-OOE
Location:	Hagenberg

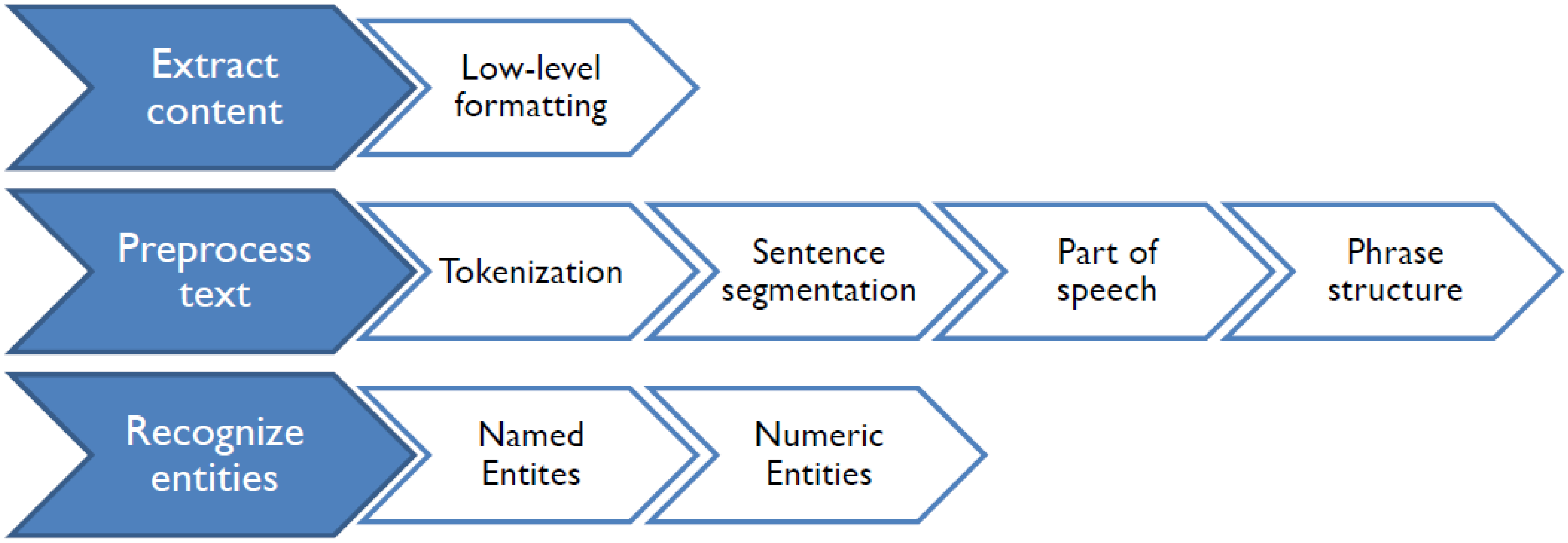
- Other common tasks: recognition of date/time expressions, measures (percent, money, weight etc), email addresses etc.
- Named-Entity-Recognition (NER)
  - Find named entities in a text
  - Assign predefined categories

NER – Named Entity Recognition

- What NER is NOT?
  - NER recognizes entities in text, and classifies them in some way, but **it does not create templates, nor does it perform co-reference or anaphora resolution.**
  - NER is not just matching text strings with pre-defined lists of names: **It recognizes text strings which are being used as entities in a given context.**

- Why do NER Recognition?
  - Key part of Information Extraction system
  - Robust handling of proper names essential for many applications
  - Pre-processing for subsequent information extraction tasks
  - Information filtering
  - Information linking

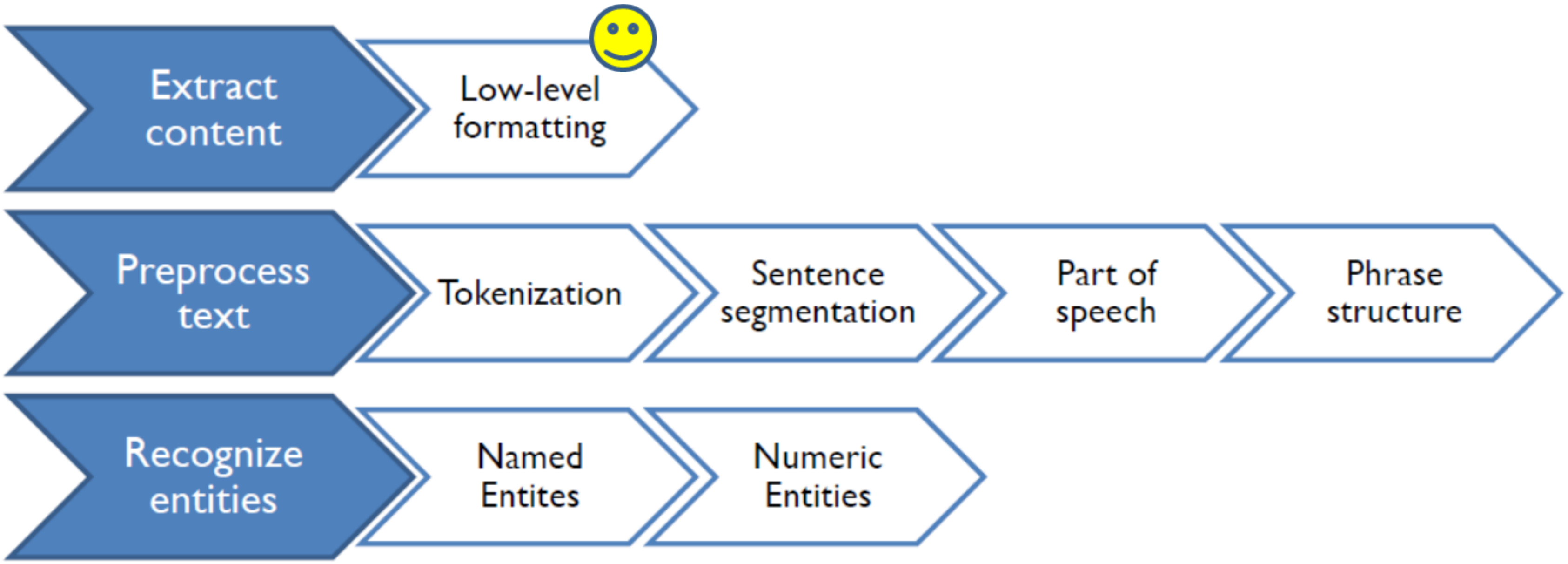
NER-Pipeline



Low Level Formatting

- **Input:** Raw text in electronical form
  - Documents (pdf, doc, rtf)
  - Websites (html)
- **Task:** Remove irrelevant content (Junk content)
  - Pictures
  - Tables
  - Diagrams
  - Advertisements

NER-Pipline

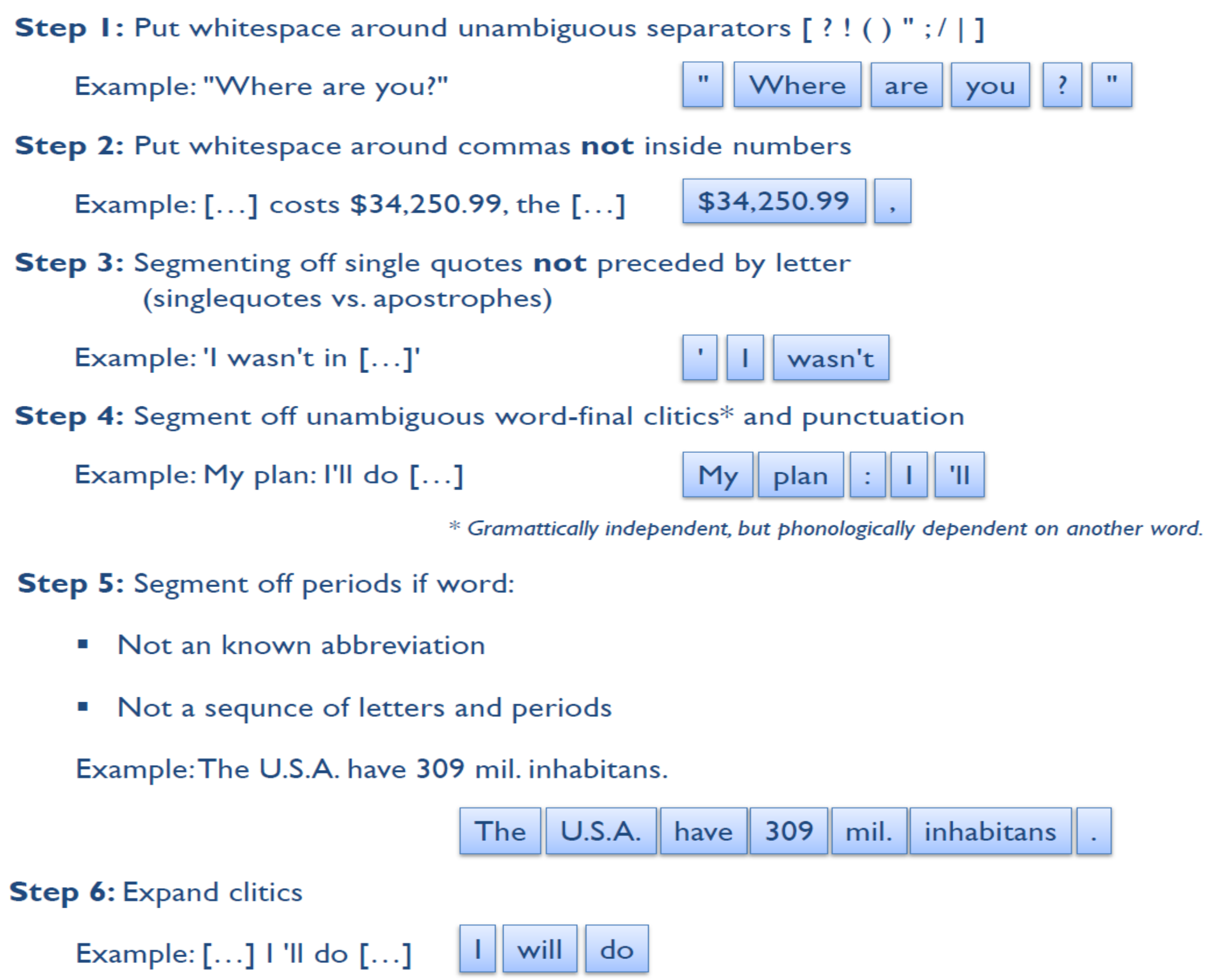

Extract content
Low-level formatting
Preprocess text
Tokenization
Sentence segmentation
Part of speech
Phrase structure
Recognize entities
Named Entites
Numeric Entities

Tokenization

- **Input:** (Junk-free) connected text
- **Task:** Divide connected text into (smaller) units / tokens
- What is a token?
  - Sequence of characters grouped together as useful semantic unit
  - How to recognize separate tokens?

    English: Words separated by whitespaces / linebreaks
- Problems:
  - Punctuation marks
    - Einstein was born in 1879.
    - The car costs $34,250.99, the bike…
  - Hyphenation
    - I am in the uni-versity
    - the New York - New Haven railroad
  - Lots more: Apostrophes, direct speech

Tokenization Problems – Heuristics



Step 1: Put whitespace around unambiguous separators [? ! ( ) " ; / |]

Example: "Where are you?"

"	Where	are	you	?	"

Step 2: Put whitespace around commas **not** inside numbers

Example: [...] costs $34,250.99, the [...]

$34,250.99	,

Step 3: Segmenting off single quotes **not** preceded by letter (singlequotes vs. apostrophes)

Example: 'I wasn't in [...]'

'	I	wasn't

Step 4: Segment off unambiguous word-final clitics* and punctuation

Example: My plan: I'll do [...]

My	plan	:	I	'll

** Gramatically independent, but phonologically dependent on another word.*

Step 5: Segment off periods if word:

- Not an known abbreviation
- Not a sequnce of letters and periods

Example: The U.S.A. have 309 mil. inhabitans.

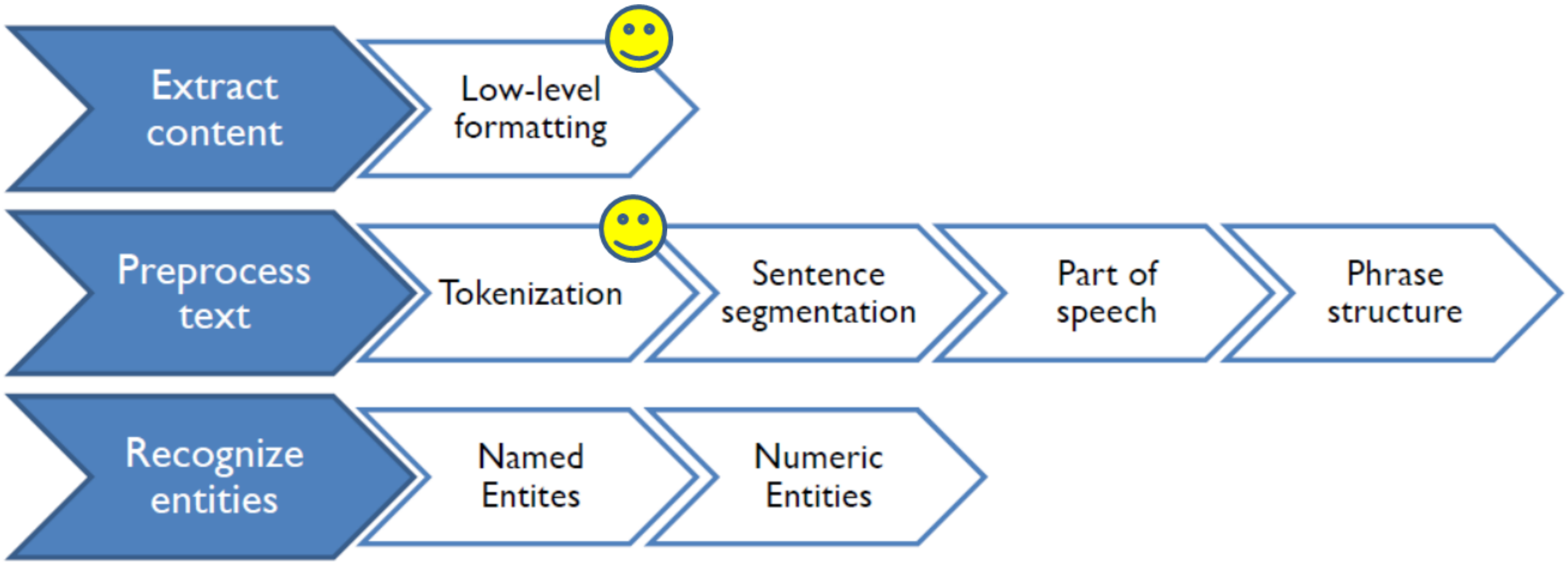
The	U.S.A.	have	309	mil.	inhabitans	.

Step 6: Expand clitics

Example: [...] I 'll do [...]

I	will	do

NER-Pipline


Extract content
Low-level formatting
Preprocess text
Tokenization
Sentence segmentation
Part of speech
Phrase structure
Recognize entities
Named Entites
Numeric Entities

Sentence Segmentation

- **Input:** Connected text
- **Task:** Divide text into sentences
- What is a sentence?
  - English language : Something ending with a . *?* or *!*
- Problems
  - Not all periods marking end of sentence
  - Other punctuation marks indicating sentence boundary [: ; -]
  - Quotes of direct speech

Sentence Segmentation Problems – Examples for Solutions

Prof. Smith Jr. said: "Google Inc. and Yahoo! etc. are search engine providers."

Step 1: Place sentence boundaries after all occurences of . ? !

Prof.| Smith Jr.| said: "Google Inc.| and Yahoo!| etc.| are search engine providers.|"

Step 2: Move boundaries after following quotation marks.

Prof.| Smith Jr.| said: "Google Inc.| and Yahoo!| etc.| are search engine providers."|

Step 3: Disqualify boundary preceded by abbr. (commonly followed by proper name)

Prof. Smith Jr.| said: "Google Inc.| and Yahoo!| etc.| are search engine providers."|

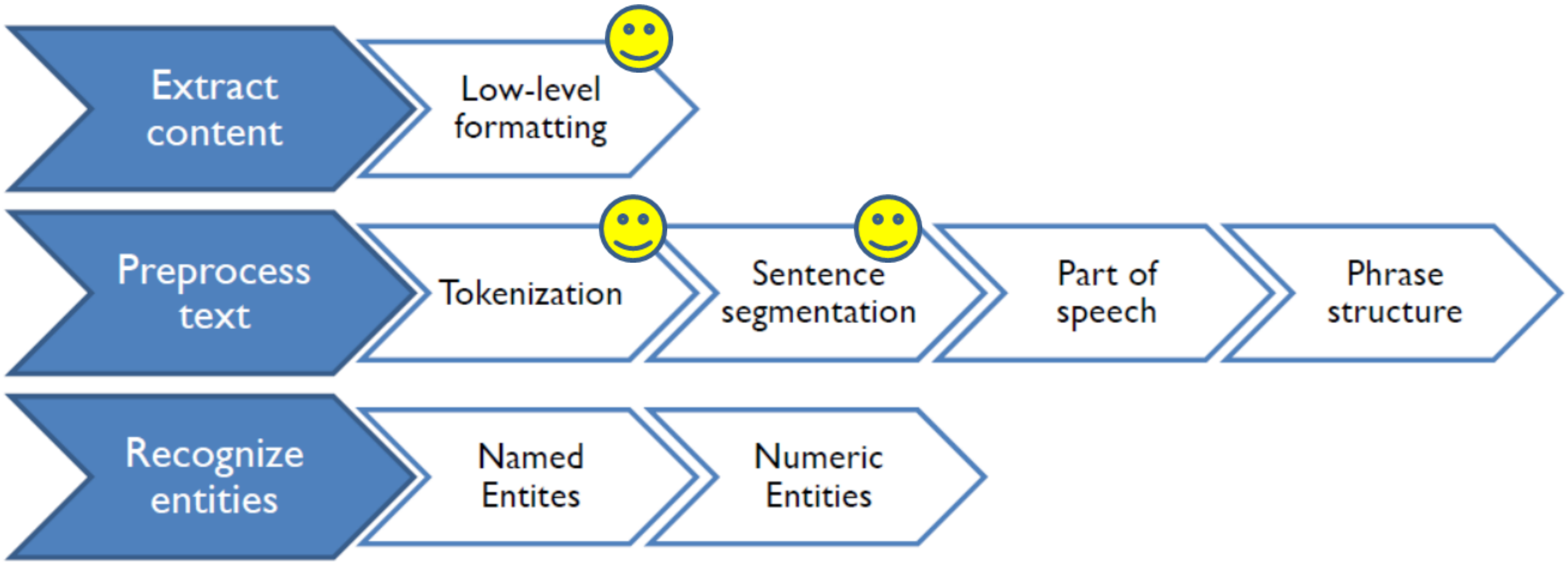
Step 4: Disqualify boundary preceded by abbr. (not followed by uppercase word)

Prof. Smith Jr. said: "Google Inc. and Yahoo!| etc. are search engine providers."|

Step 5: Disqualify boundary with a ? or ! if followed by lowercase letter

Prof. Smith Jr. said: "Google Inc. and Yahoo! etc. are search engine providers."|

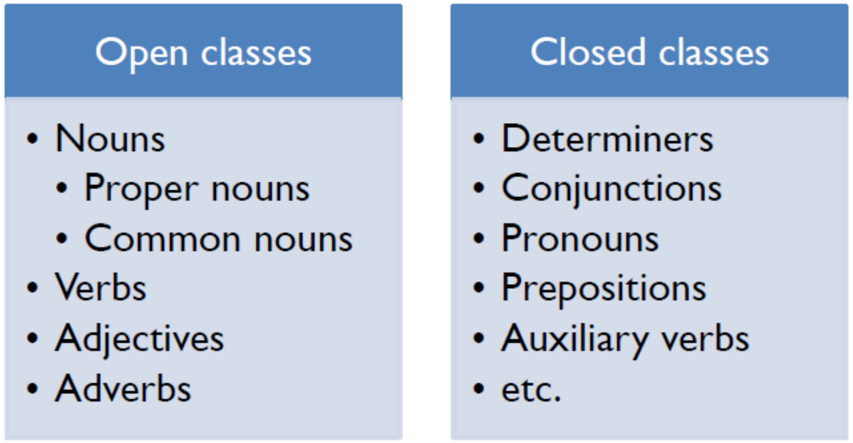
NER-Pipline


Extract content
Low-level formatting
Preprocess text
Tokenization
Sentence segmentation
Part of speech
Phrase structure
Recognize entities
Named Entites
Numeric Entities

Part of Speech Tagging (POS)

- **Input**: Tokenized text with sentence boundaries
- **Task**: Assigns parts of speech to each word such as noun, verb, adjective, etc.

- **Open class** is a class of words which does not consist of a finite number of words
- **Closed classes** consist of a finite number of words



Open classes	Closed classes
• Nouns • Proper nouns • Common nouns • Verbs • Adjectives • Adverbs	• Determiners • Conjunctions • Pronouns • Prepositions • Auxiliary verbs • etc.

On line demo pos-tagger : http://cogcomp.cs.illinois.edu/demo/pos/?id=4

Part of speech: Morphology

- Natural language is complex
- Grammatical distinction for words (context dependency)
- Inflection: Systematic modification of a root form

Nouns	Verbs	Adjectives
• Number inflection • Gender inflection • Case inflection	• Subject number • Subject person • Tense • etc.	• Positive • Comparative • Superlative

English: only number inflection
German: number, gender and case inflection

TAG –Sets
Penn Treebank Tagging – English language

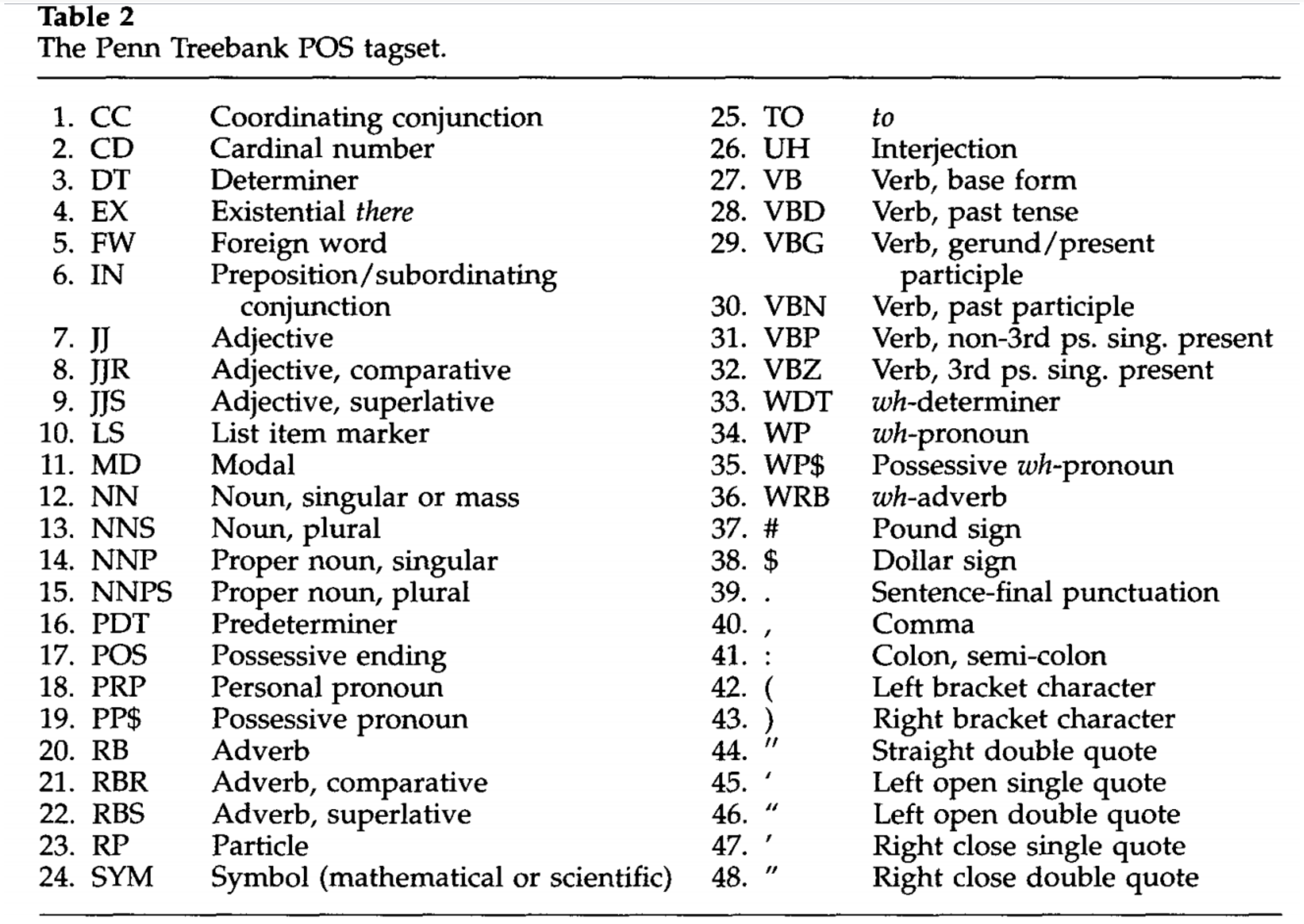


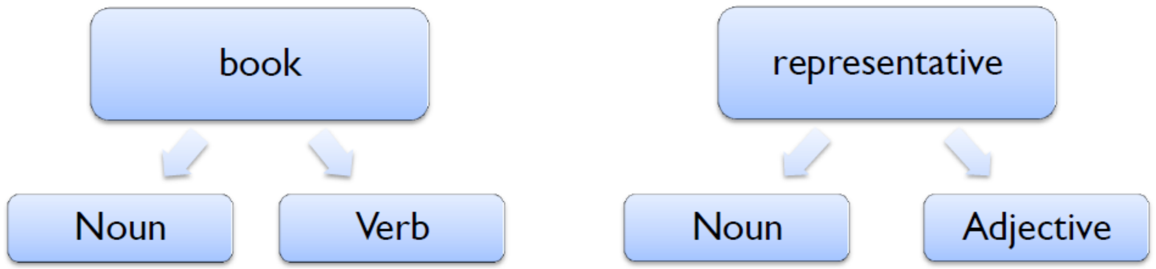
Table 2
The Penn Treebank POS tagset.

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	‘	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	”	Right close double quote

See: http://trac.sketchengine.co.uk/wiki/tagsets/penn

Part of Speech - Problems

- Ambiguity
  - often have more than one word class: book (Noun or Verb?)
  - depends on context



I	am	reading	a	book	.
PRP	VBP	VBG	DT	NN	.

He	wants	to	book	that	flight	!
PRP	VBZ	TO	VB	DT	NN	.

Part of Speech Tags: Rule Based approach

- Oldest approach
- Disambiguation is done by analyzing the linguistic features of the word, its preceding word, its following word and other aspects (context of surrounding text).

- Rules – examples:
  - *If the previous word is "to", then it's a verb.*
  - *If the previous word is "a", then it's a noun.*
  - *If the next word is …*

- Rules – examples:
  - Given input "that"
  - *If the next word is adj, adverb, or quantifier, and following that is a sentence boundary and the previous word is not a verb like "consider" which allows adjs. as object complements,*
  - *Then eliminate non-ADV tags,*
  - *Else eliminate ADV tag*

    - *I consider that odd. (that is NOT ADV)*
    - *It isn't that strange. (that is an ADV)*

Writing rules manually is hard work !

Part of Speech Tags: Probabilistic approach

Single word probability:

- *P(x|y):* Probability of word x being of type y
- Example
  - *P(NN|play)* = 0.24
  - *P(VB|play)* = 0.76

Word sequence probability:

- *P(x|y):* Probability of type x following type y
- Example
  - *P(NN|TO)* = 0.00047
  - *P(VB|TO)* = 0.83

Part of Speech Tags: Unigram Tagger

- A Unigram Tagger assigns to each token its most likely tag
- Only considers the current token, preceding tags are not considered
- Before a “UnigramTagger“ can be used to tag data, it must be trained on a “**training corpus**”.
- Training results in a table of unigram frequencies based on the corpus

Part of Speech Tags: Unigram Tagger Example

Suppose we have a corpus of simple sentences containing 1,273,000 tokens, including say 1000 uses of the word *flies*: 400 of them as an NN and 600 of them as a VB. Which POS is more likely for the word *flies*?

P(flies) = 1000/1,273,000 = 0.00078554595

P(flies ∧ NN) = 400/1,273,000 = 0.00031421838

P(flies ∧ VB) = 600/1,273,000 = 0.00047132757

Conditional Probability:
P(A|B) = P(A ∧ B)/P(B)

Our best guess will thus be that VB is the most likely part-of-speech. (And we should be right about 60% of the time.)

P(VB|flies) = P(flies ∧ VB)/P(flies) = 0.00047132757/0.00078554595 = 0.6

also simpler expressible as:

P(VB|flies) = count(flies ∧ VB) / count(flies) = 600/1000 = 0.6

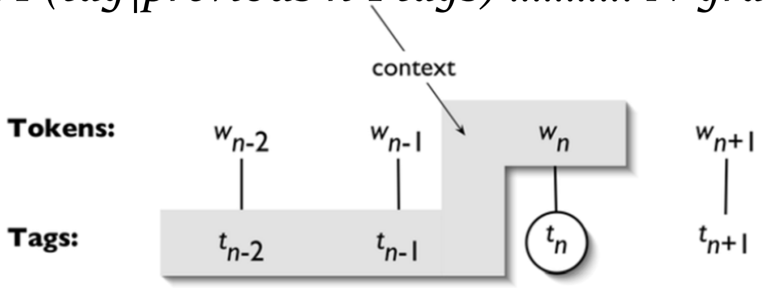
P(NN|flies) = count(flies ∧ NN) / count(flies) = 400/1000 = 0.4

Problems

- No probabilities for words not in corpus.
- Training corpus may be different from test corpus.

Part of Speech Tags: Stochastic (Probabilistic) Tagging N-gram / HMM based tagging

- **N-gram Tagger**
  - generalization of a unigram tagger
  - considers the current word together with the part-of-speech tags of the n-1 preceding words
  - Sequence classification task: given a sequence of N words, return a sequence of N tags
- **Basic idea:** consider all possible sequences of tags and choose the most probable given the sequence of words: choose tag sequence that maximizes this formula:
  - *P(word|tag) × P(tag|previous n-1 tags) .... N-gram*



Part of Speech Tags: Hidden Markov Model based tagging

- Probability calculations imply Markov models: we assume that *P(w|t)* is dependent only on the (or, a sequence of) previous word(s)
- **(Informally) Markov models are the class of probabilistic models that assume we can predict the future without taking too much account of the past**
- Markov chains can be modeled by finite state automata: the next state in a Markov chain is always dependent on some finite history of previous states
- Called "hidden" because one cannot tell what state the model is in for a given sequence of words.

- Example:
    - Bigram Taggers assign tags on the basis of sequences of two words (usually assigning tag to $word_n$ on the basis of $word_{n-1}$)

- An *n-th*-order tagger assigns tags on the basis of sequences of *n* words
- As the value of *n* increases, the complexity of the statistical calculation involved in comparing probability combinations increases too.

Part of Speech Tags: Bi-gram Tagging with HMM Example

Given Toy-Corpus: Corpus consists of 300 sentences having words in only four categories: N(oun), V(erb), ART(icle), and P(reposition): 833 nouns, 300 verbs, 558 articles, and 307 prepositions for a total of 1998 words.

Category	Count at i	Pair	Count at i, i+1	Bigram	Estimate
<start>	300	<start>,ART	213	P(Art|<start>)	.71
<start>	300	<start>,N	87	P(N|<start>)	.29
ART	558	ART,N	558	P(N|ART)	1
N	833	N,V	358	P(V|N)	.43
N	833	N,N	108	P(N|N)	.13
N	833	N,P	366	P(P|N)	.44
V	300	V,N	75	P(N|V)	.35
V	300	V,ART	194	P(ART|V)	.65
P	307	P,ART	226	P(ART|P)	.74
P	307	P,N	81	P(N|P)	.26

Bigram probabilities from the generated corpus

Part of Speech Tags: Bi-gram Tagging with HMM Example

	N	V	ART	PRE	TOTAL
flies	21	23	0	0	44
fruit	50	5	0	0	55
like	10	30	0	31	71
a	1	0	201	0	202
the	0	0	303	0	303
flower	53	15	0	0	68
flowers	42	16	0	0	58
birds	64	1	0	0	65
others	582	210	56	284	1132
TOTAL	833	300	560	315	1998

Some corpus word counts

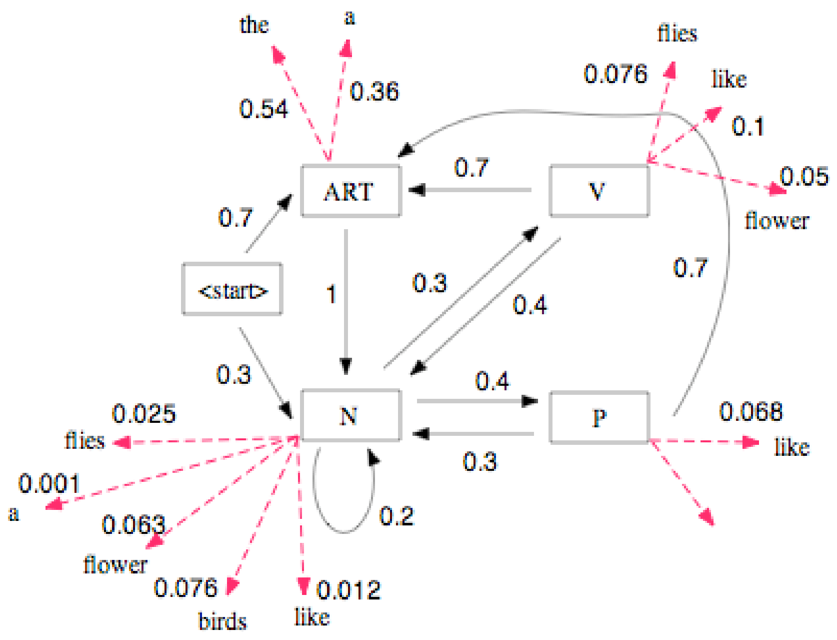
Some of lexical generation probabilities

P(the | ART)	0.54
P(flies | N)	0.025
P(flies | V)	0.076
P(like | V)	0.1
P(like | P)	0.068
P(like | N)	0.012
P(a | ART)	0.360
P(a | N)	0.001
P(flower | N)	0.063
P(flower | V)	0.05
P(birds | N)	0.076

Given some tag T_i
the probability of a particular word w_i is:

$$P(w_i \mid T_i) = \frac{\#(occurrences\ of\ word\ w_i\ with\ tag\ T_i)}{\#(words\ with\ tag\ T_i)}$$

Part of Speech Tags:
Bi-gram Tagging with HMM Example



Formally defined by:

States Q = $q_1, q_2 \ldots q_{N;}$
Observations O = $o_1, o_2 \ldots o_{M;}$
Transition probabilities (prior)

Transition probability matrix $A = \{a_{ij}\}$

Observation likelihoods (likelihood)

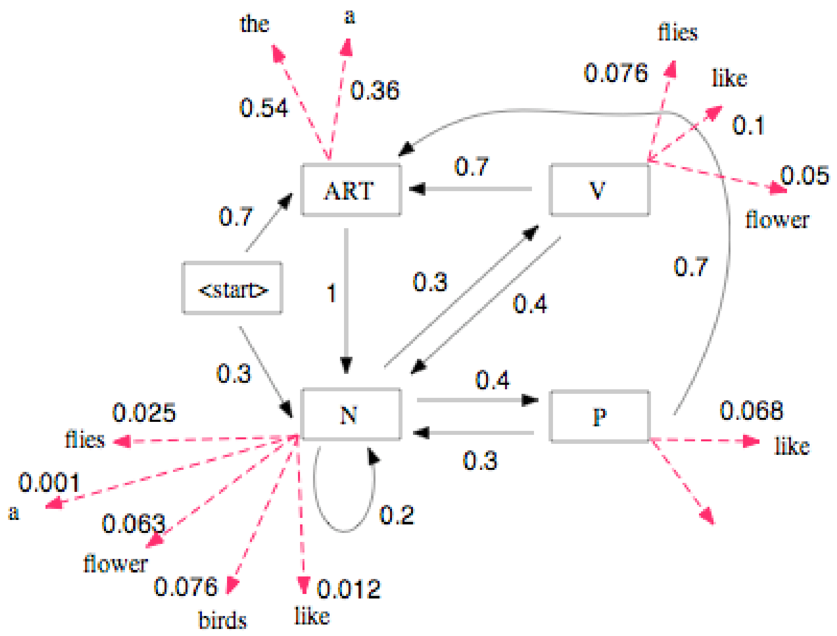
Output probability matrix $B=\{b_i(o_t)\}$

a set of observation likelihoods, each expressing the probability of an observation o_t being generated from a state i, (emission probabilities)

Special initial probability vector the probability that the HMM will start in state *i*, each expresses the probability

$p(q_i|START)$

Part of Speech Tags: Bi-gram Tagging with HMM Example



States Q = $q_1, q_2 \ldots q_{N;}$

Q={ <start>, ART,N,V,P}

Observations O = $o_1, o_2 \ldots o_{M;}$

O= {flies, fruit, like, a, the, flower, flowers, birds, others }

Transition probabilities (prior)

Transition probability matrix A = $\{a_{ij}\}$

	ART	V	N	P
ART	0	0	1	0
V	0.7	0	0.4	0
N	0	0.3	0.2	0.4
P	0.7	0	0.3	0

Observation likelihoods (likelihood) - only shown for state N

P(flies | N)	0.025
P(like | N)	0.012
P(a | N)	0.001
P(flower | N)	0.063
P(birds | N)	0.076

Special initial probability vector

ART	0.7
V	0
N	0.3
P	0

Part of Speech Tagging: HMM-Finding the Most Likely Tag Sequence

- We want to generate all possible tag sequences to determine which one is most likely.
- To find the most likely sequence, sweep forward through the words one at a time finding the most likely sequence for each ending category.
- In other words, find the four best sequences for the two words
  - Flies like:
    - the best ending with
      - like as a V, like as a N, like as a P, like as an ART.
- Then use this information to find the four best sequences for the words flies like a, each one ending in a different category.
- This process is repeated until all the words are accounted for. (flies like a flower)

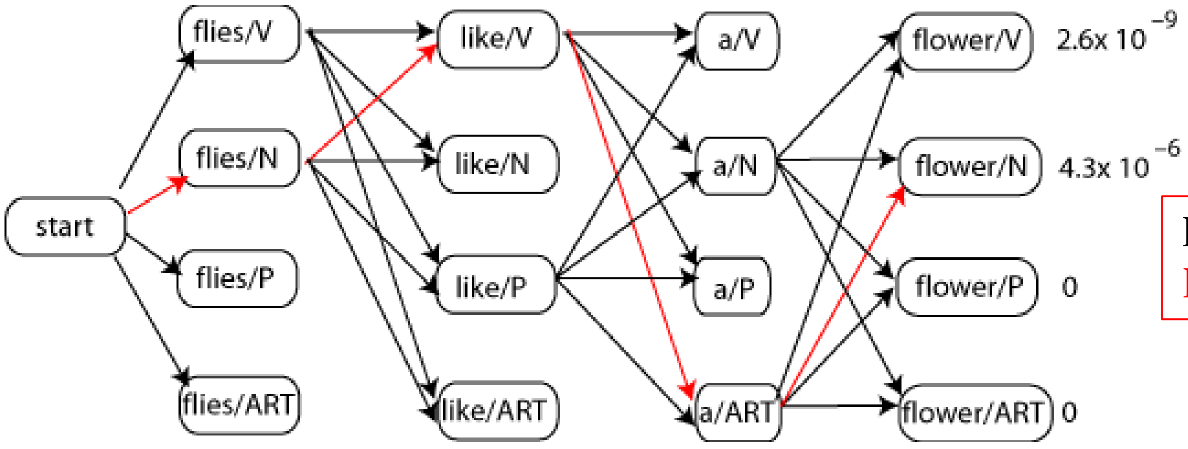
$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \overbrace{P(w_1^n|t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

- But there can be an exponential number of possible tag sequences!

→ Use Viterbi Algorithm

Part of Speech Tagging: HMM Viterbi Algorithm

Word sequence: **Flies like a flower**



Most likely tag sequence
Flies/N like/V a/ART flower/N

V	7.6×10^{-6}	$\boxed{0.00031}$	0	2.6×10^{-9}
N	$\boxed{0.00725}$	1.3×10^{-5}	1.2×10^{-7}	$\boxed{4.3\times10^{-6}}$
P	0	0.00022	0	0
ART	0	0	$\boxed{7.2\times10^{-5}}$	0

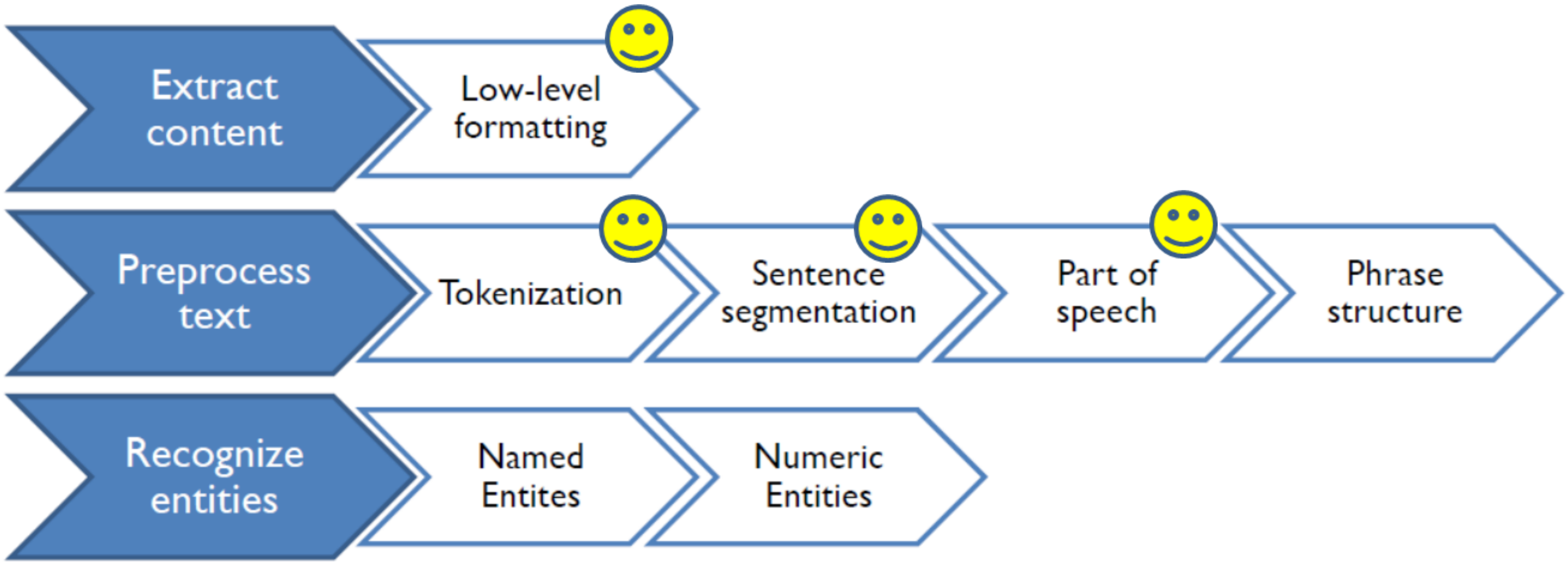
For each point in time and status a local path probability is calculated.

Calculation examples for max values:

P(N|<start>) x P(flies|N)=0.00725
P(N|<start>) x P(flies|N) x P(V|N) x P(like|V)
0,00725 x 0 .43 x 0,1 = 0,00031
P(N|<start>) x P(flies|N) x P(V|N) x P(like|V) x P(A|V)
x P(a|V) = 0,00031 x 0,65 x 0,36 ~ 0,000072

P(N|<start>) x P(flies|N) x P(V|N) x P(like|V) x P(A|V)
x P(a|V) x P(N|A) x P(flower|N) =
= 0,0000729495 x 1 x 0,063 ~ 0,0000043

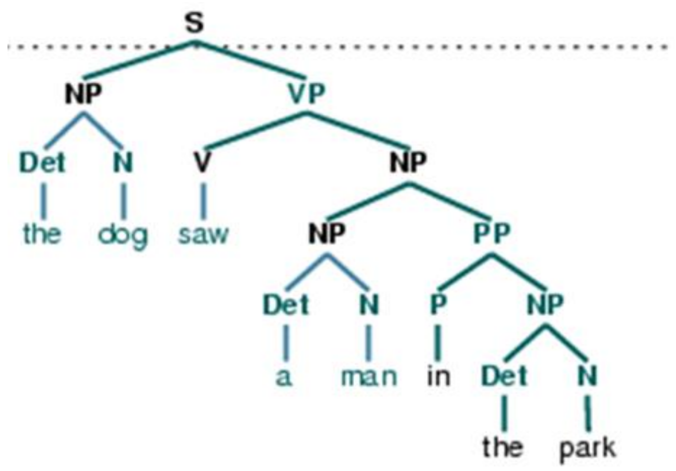
NER-Pipeline



Phrase Structure: Full Parsing

- **Input:** Tokenized text with POS tags and sentence boundaries
- **Task:** Assign (full) syntactic structure to sentences

- Determine the structure of the input text
- Identify syntactic groups within a sentence
  - Noun phrases (NP)
  - Prepositional phrases (PP)
  - Verb phrases (VP)
  - Adjective phrases (AP)
- Represented by context-free grammar
  - Build a complete parse tree for a sentence
  - Sentence consists of a hierarchy of phrases consisting of basic syntactic groups
  - Example: *The dog saw a man in the park*



Phrase Structure: Full Parsing

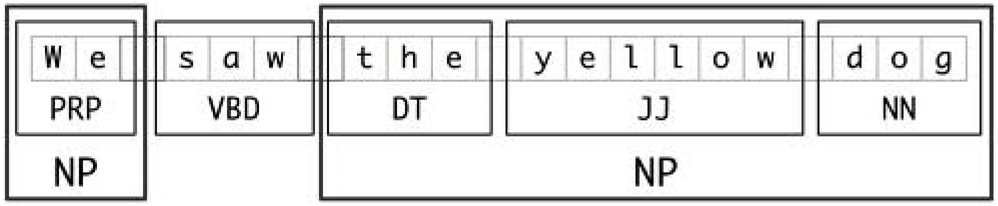
- Problems using context-free grammars
  - Exponential solution space
  - Dependence on semantic context
  - Long-range dependencies
  - Ambiguity
  - Errors
  - Too slow to use with very large sources of text (e.g., the web).

- Question: „**Do we really need Full Syntactic Parsing?**
  - Goal revisited
    - Find named entities
    - No need for full parse tree
  - Solution: Partial parsing

Phrase Structure: Chunking (Partial Parsing)

- **Idea:** Identify and classify basic phrases of a sentence
- Approach is called **chunking**
- Example:
  [NP We] [VP saw] [NP the yellow dog].
- Chunking: assign a ***partial*** structure to a sentence.
  - Simpler solution space
  - Local context
  - Non-recursive
- Chunking divides a sentence into a sequence of chunks
  - Chunks are non-overlapping regions of a text
    [We] saw [the yellow dog].
  - Chunks are non-recursive
  - A chunk can not contain other chunks
  - Chunks are non-exhaustive
  - Not all words are included in chunks
  - Noun-phrase (NP) chunking
    [We] saw [the yellow dog].
  - Verb-phrase (VP) chunking:
    We [saw] the yellow dog.

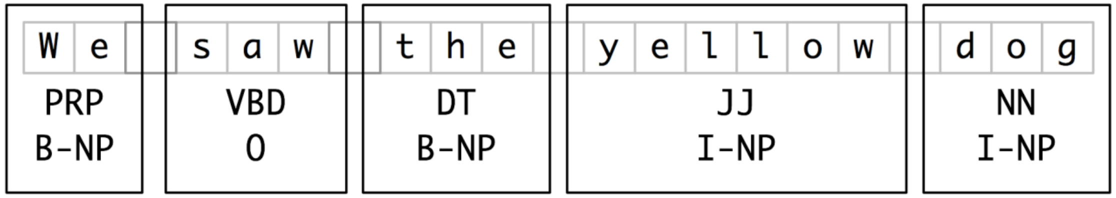
Possible approaches:
- chunking with regular expressions and tag patterns
- train a chunk parser



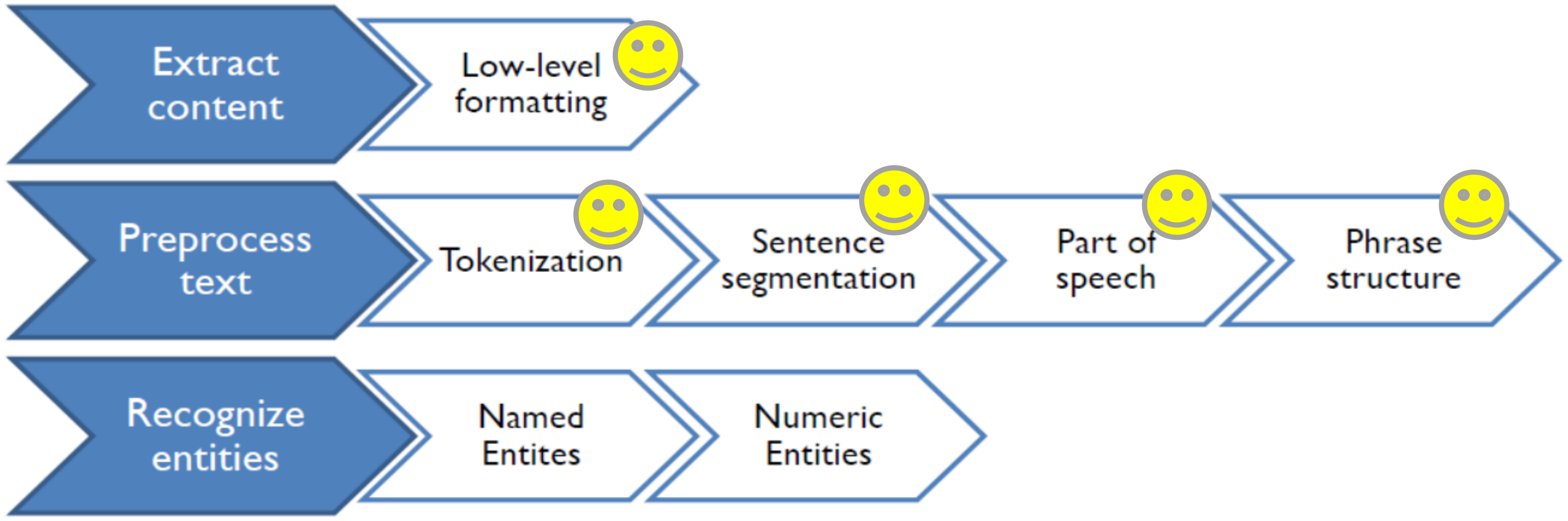
Noun-phrase (NP) chunking

Phrase Structure: Representation of Chunks

- Most widespread representation for chunks uses so-called IOB (BIO) tags
- IOB tags have become the standard way to represent chunk structures
  - Each token is tagged with one of three special chunk tags,
    - I (inside)
    - O (outside)
    - B (begin).
  - A token is tagged as B if it marks the beginning of a chunk.
  - Subsequent tokens within the chunk are tagged with I.
  - All other tokens are tagged O.
  - B and I tags are suffixed with the chunk type, e.g. B-NP, I-NP.
  - Not necessary to specify a chunk type for tokens that appear outside a chunk, so these are just labeled O



NER - Pipeline



Conclusion of preprocessing

- Several steps based on each other
- Error-prone
- Inaccuracies affect all subsequent results

Recognize Entities

- **Input:**
  - Preprocessed text with various Tags
  - Not all steps described before have to be applied
  - Depends on implementation
- **Task:**
  - Find named entities (NE)
  - Broken into two subproblems
    - identifying the boundaries of the NE (segmentation)
    - identifying its category (classification)

Jack Welch will retire as CEO of General Electric tomorrow. The top role at the Connecticut company will be filled by Jeffrey Immelt.

segmentation

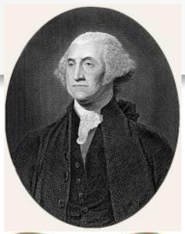
Jack Welch will retire as CEO of General Electric tomorrow. The top role at the Connecticut company will be filled by Jeffrey Immelt.

classification

Recognize Entities: NER Categories

- Generic Named Entities
  - Person (PER)
  - Organization (ORG)
  - Location (LOC)
- Custom Named Entities (domain specific)
  - Medical Context: Drugs, Deseases
  - Biological Context: Proteins, Genes
  - Sports Context: Soccer, Skiing
- Numeric Entities
  - Temporal Expressions (TIME)
  - Numerical Expressions (NUM)

NER- Ambiguity Problem



- **Washington** was born in 1732.
  - Category: Person (PER)
- **Washington** has won the last game against Denver.
  - Category: Organization (ORG)

- Merkel arrived in **Washington** for a state visit.
  - Category: Location (LOC)
- **Washington** just passed a primary seatbelt law.
  - Category: Geo-Polictal Entity (GPE)

- The **Washington** is stationed in Yokosuka, Japan.
  - Category: Vehicle (VEH)
- Tourists prefer to stay at the **Washington**.
  - Category: Facility (FAC)

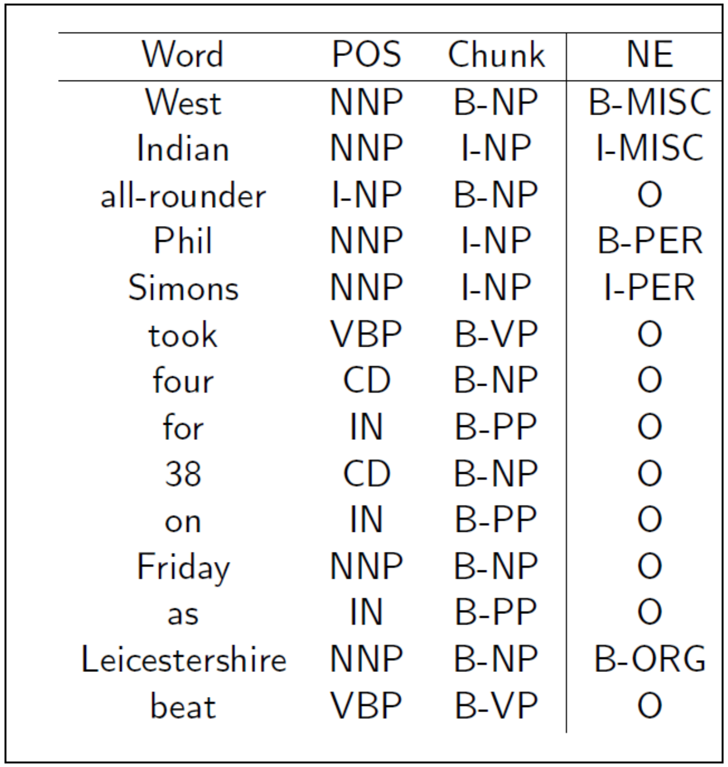
NER: Representation of NE

- Use IOB Tagging for NE:
  - B-PER I-PER
  - B-ORG I-ORG
  - B-LOC I-LOC
  - O
  - ……
- Source text:
  … the racer of Yamaha Valentino Rossi …..
- Annotated text (IOB version):

……..	
the	O
racer	O
of	O
Yamaha	B-ORG
Valentino	B-PER
Rossi	I-PER

Sequence Labeling

- Many Problems in IE can be viewed as sequence labeling.
- Each token in a sequence is assigned a label or tag
- attempt to assign a tag to each token in a sequence
- Tags in NER can be
  - POS Tags
  - Chunking Tags
  - NE Tags
- Sequence labeling can be treated as a set of independent **classification tasks**, one per token of the sequence.
- Accuracy is generally improved by making the optimal label for a given token dependent on the choices of nearby tokens.



Word	POS	Chunk	NE
West	NNP	B-NP	B-MISC
Indian	NNP	I-NP	I-MISC
all-rounder	I-NP	B-NP	O
Phil	NNP	I-NP	B-PER
Simons	NNP	I-NP	I-PER
took	VBP	B-VP	O
four	CD	B-NP	O
for	IN	B-PP	O
38	CD	B-NP	O
on	IN	B-PP	O
Friday	NNP	B-NP	O
as	IN	B-PP	O
Leicestershire	NNP	B-NP	B-ORG
beat	VBP	B-VP	O

Sequence Labels

Sequence Labeling – a classification problem

Sequence labeling setting:

Input sequence: $x = x_1, x_2,,,,,,,,,x_N$ (Tokens of Input Sentence)

Output sequence: $z = z_1, z_2,,,,,,,z_N$ (Labels: Tags from specific Tag set)

Local Classifier

- For each position i , classify each x_i independently from neighbors
- Criterion: $\mathbf{z_i} = \mathbf{argmax}\ \boldsymbol{P(z_i|x_i)}$
- Each example consists of a single position in the sequence (a token/label pair)
- Any classification method can be used (mostly from the field of supervised learning):
  - Hidden Markov Models (HMM)
  - Decision Trees (DT)
  - Maximum Entropy Models (ME)
  - Support Vector Machines (SVM)
  - Conditional Random Fields (CRF)
  - Perceptron (NN)

Sequence Labeling – a classification problem

Local Classifier with previous labels:

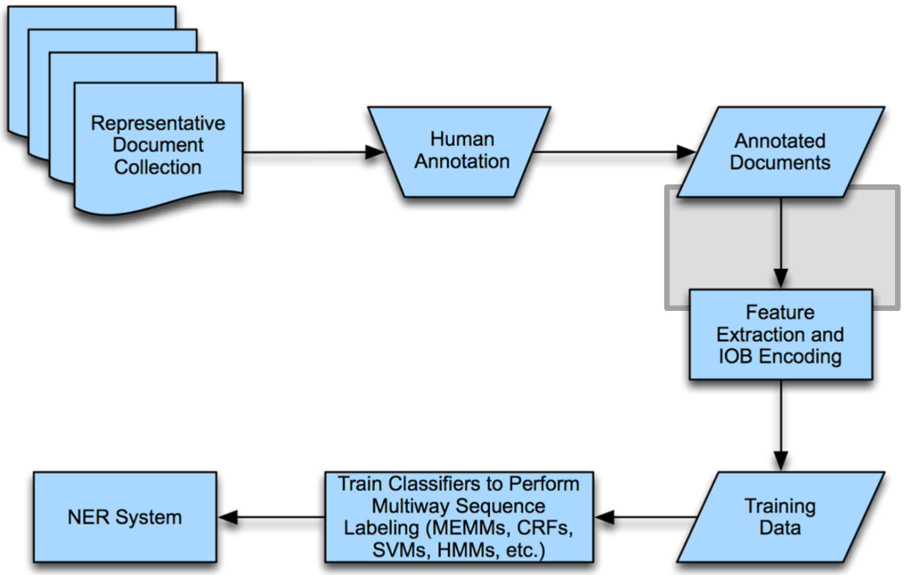
- For each position *i* , classify each x_i sequentially (e.g. from left to right), making use of the previously assigned label
- Criterion: $\mathbf{z_i = argmax}\ \boldsymbol{P(z_i|x_i, z_{i-1})}$

Global Classification:

- label by maximizing on the entire sentence
- Criterion: $\boldsymbol{z = argmax\ P(z|x)}$
- Well established method:
  - CRF (Conditional Random Field)

Training the classifier

- Task Definition: **Which are the entities in the domain?**
- The **expert/annotator** selects **representative examples** where each example is classified.
- Examples **(training set)** are given as input to a learning method, which produces a classifier.
- **Performance** of the classifier is **evaluated** on a different set of annotated data (test set).
- If performance is considered good for the task, then the classifier is applied on a large-scale.


Representative Document Collection
Human Annotation
Annotated Documents
Feature Extraction and IOB Encoding
Training Data
Train Classifiers to Perform Multiway Sequence Labeling (MEMMs, CRFs, SVMs, HMMs, etc.)
NER System

Need to have features for training

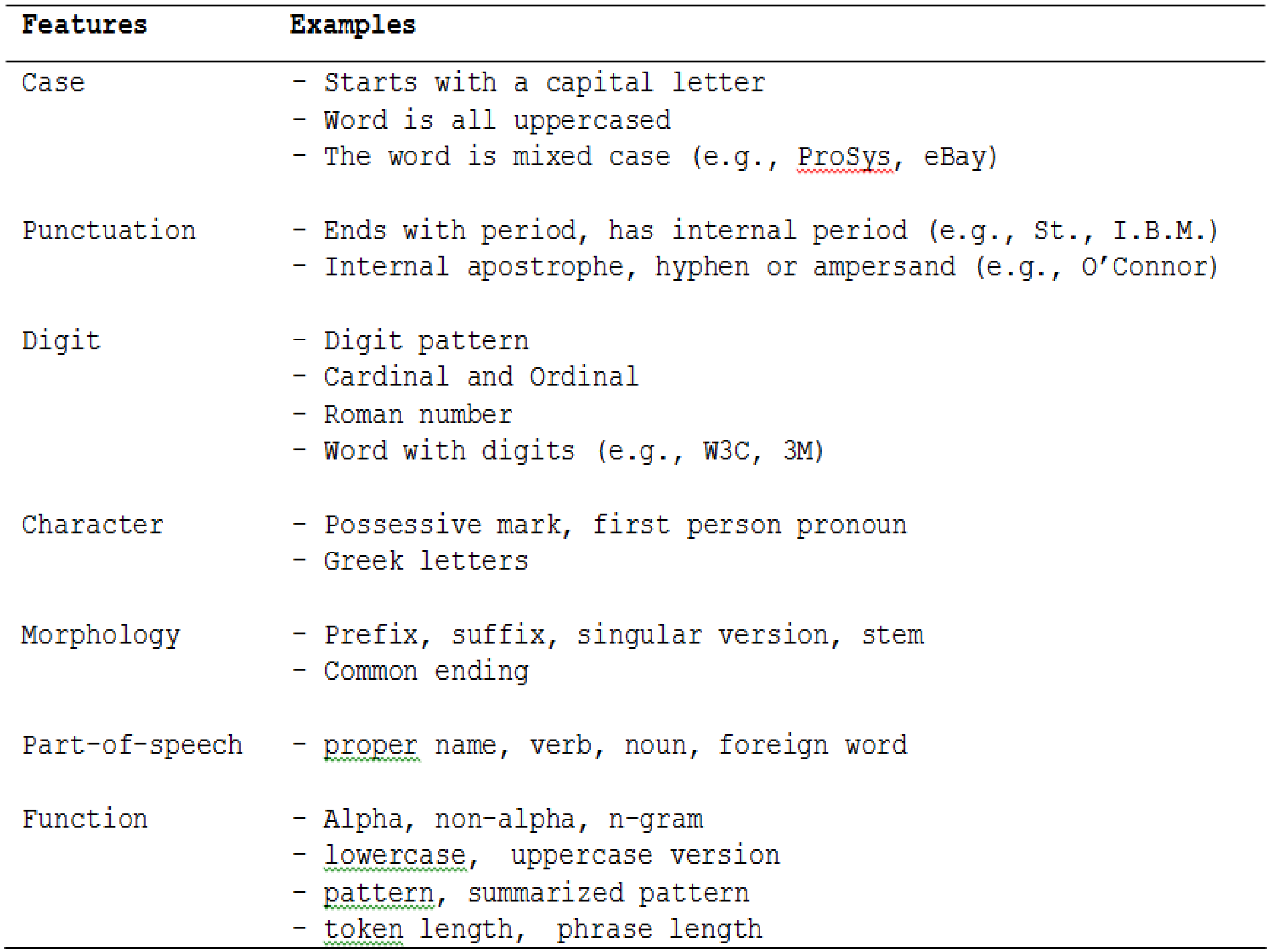
Features for Training of Classifier Example NE

- **Identification of lexical and phrasal characteristics** in text which give clues to named entities
- **Choice of the best discriminative features** to represent NEs affects
  - accuracy,
  - learning time (depends on the optimal training data).
- In many NER applications, it is not unusual to find **thousands** of features.
- It has been observed that beyond a certain point, the inclusion of additional features leads to worse rather than better performance

Features

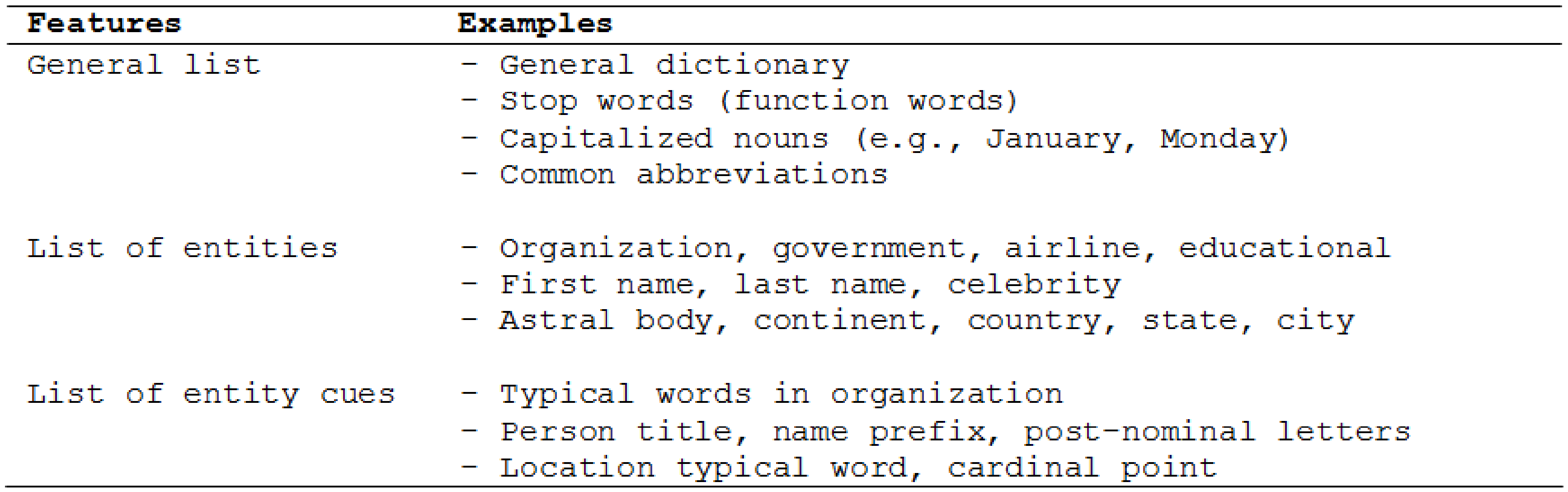
- **Word-level features:**
  - Word-level features are related to the character makeup of words.
  - specifically describe word case, punctuation, numerical value and special characters
- **List lookup features:**
  - Lists are the privileged features in NER
  - “gazetteer”, “lexicon” and “dictionary” are often used interchangeably with the term “list”.
  - List inclusion is a way to express the **relation “is a”** (e.g., Paris is a city). It may appear obvious that if a word (Paris) is an element of a list of cities, then the probability of this word to be city, in a given text, is high.
- **Document & corpus features:**
  - go beyond the single word and multi-word expression and include meta-information about documents and corpus statistics

Word-level features - Examples



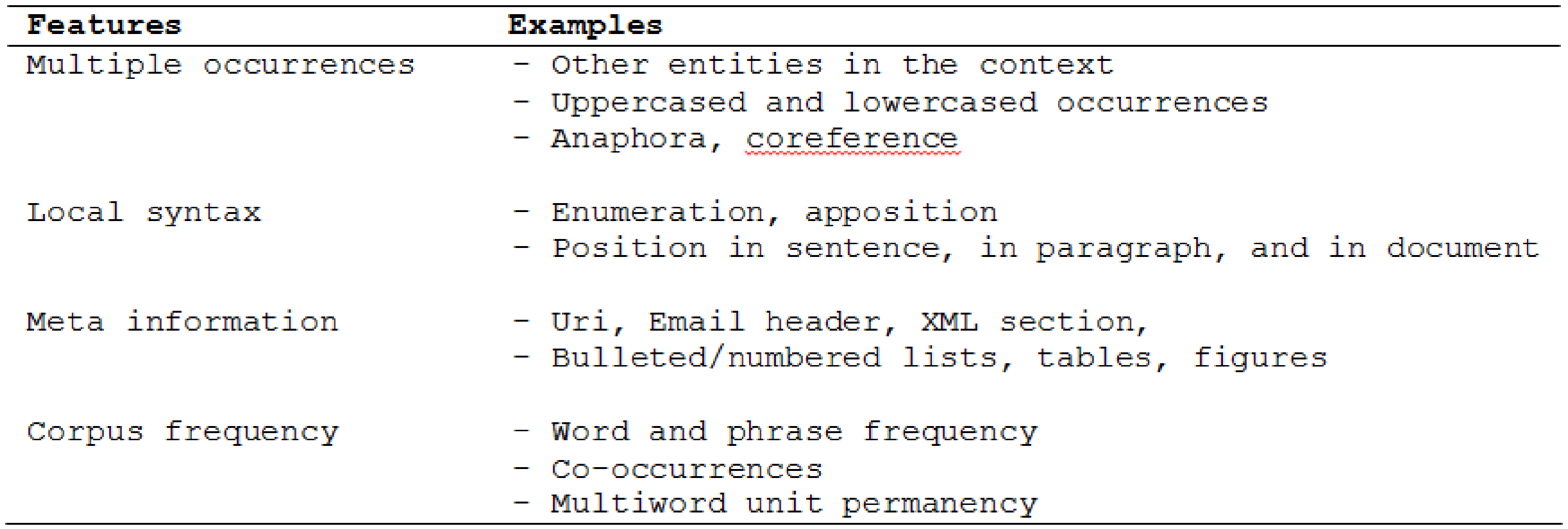
Features	Examples
Case	- Starts with a capital letter - Word is all uppercased - The word is mixed case (e.g., ProSys, eBay)
Punctuation	- Ends with period, has internal period (e.g., St., I.B.M.) - Internal apostrophe, hyphen or ampersand (e.g., O'Connor)
Digit	- Digit pattern - Cardinal and Ordinal - Roman number - Word with digits (e.g., W3C, 3M)
Character	- Possessive mark, first person pronoun - Greek letters
Morphology	- Prefix, suffix, singular version, stem - Common ending
Part-of-speech	- proper name, verb, noun, foreign word
Function	- Alpha, non-alpha, n-gram - lowercase, uppercase version - pattern, summarized pattern - token length, phrase length

List lookup features - Examples



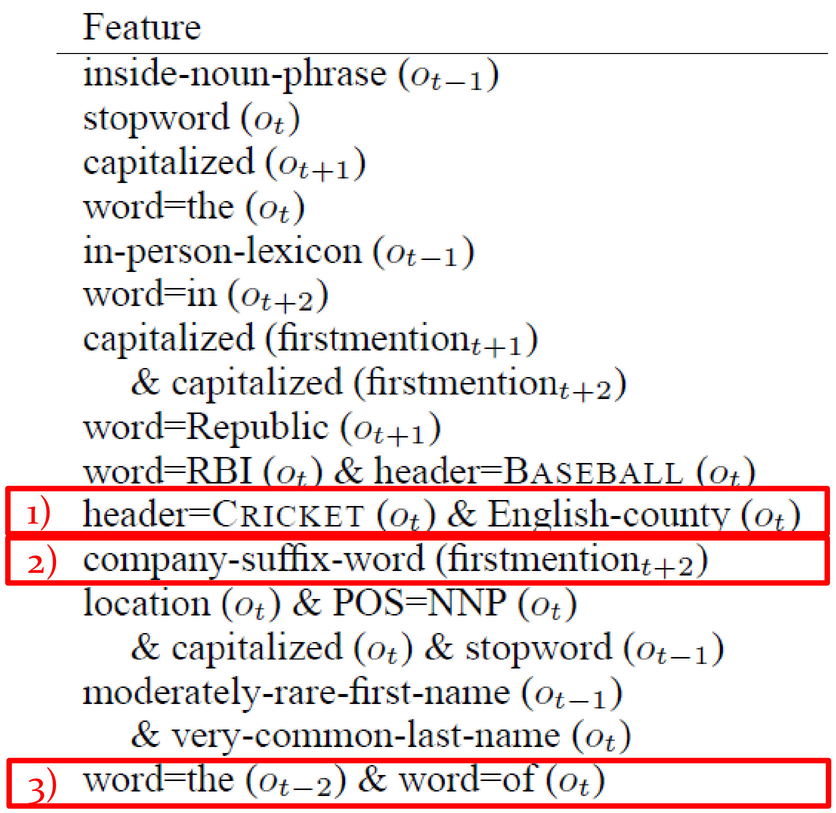
Features	**Examples**
General list	– General dictionary – Stop words (function words) – Capitalized nouns (e.g., January, Monday) – Common abbreviations
List of entities	– Organization, government, airline, educational – First name, last name, celebrity – Astral body, continent, country, state, city
List of entity cues	– Typical words in organization – Person title, name prefix, post-nominal letters – Location typical word, cardinal point

Document & corpus features- Examples



Features	Examples
Multiple occurrences	- Other entities in the context - Uppercased and lowercased occurrences - Anaphora, coreference
Local syntax	- Enumeration, apposition - Position in sentence, in paragraph, and in document
Meta information	- Uri, Email header, XML section, - Bulleted/numbered lists, tables, figures
Corpus frequency	- Word and phrase frequency - Co-occurrences - Multiword unit permanency

Features Functions



	Feature
	inside-noun-phrase (o_{t-1})
	stopword (o_t)
	capitalized (o_{t+1})
	word=the (o_t)
	in-person-lexicon (o_{t-1})
	word=in (o_{t+2})
	capitalized (firstmention$_{t+1}$) & capitalized (firstmention$_{t+2}$)
	word=Republic (o_{t+1})
	word=RBI (o_t) & header=BASEBALL (o_t)
1)	header=CRICKET (o_t) & English-county (o_t)
2)	company-suffix-word (firstmention$_{t+2}$)
	location (o_t) & POS=NNP (o_t) & capitalized (o_t) & stopword (o_{t-1})
	moderately-rare-first-name (o_{t-1}) & very-common-last-name (o_t)
3)	word=the (o_{t-2}) & word=of (o_t)

In this example o_t means current input token;
1) ORGANIZATION: „ Cricket … Essex County …"
2) ORGANIZATION: „… Apple Inc. …"
3) ORGANIZATION: „… the CEO of UAS …"

- Each feature is encapsulated into a feature function
- Feature functions evaluate to true or false
- Output is a bit vector which length corresponds to number of features evaluated
- Example: [0,1,0,0,0,1,0,1,0,0,1,1]

Training

- Each feature function carries a weight that gives the strength of that feature function for the proposed label
  - **High positive** weights indicate a **good association** between the feature and the proposed label
  - **High negative** weights indicate a **negative association** between the feature and the proposed label
  - Weights **close to zero** indicate the feature has **little or no impact** on the identity of the label
- For given *x* find

$$z = arg\ max_z\ P(z/x) = arg\ max_z\ \Sigma w_i f_i(x,z)$$

$$w_i \ldots\ Weigths\ \text{ and }\ f_i(x,z) \ldots.\ Feature\ Functions$$

- Use gradient or other optimization procedure to set weights to maximize conditional likelihood of training data

Conditional Random Fields (CRF)

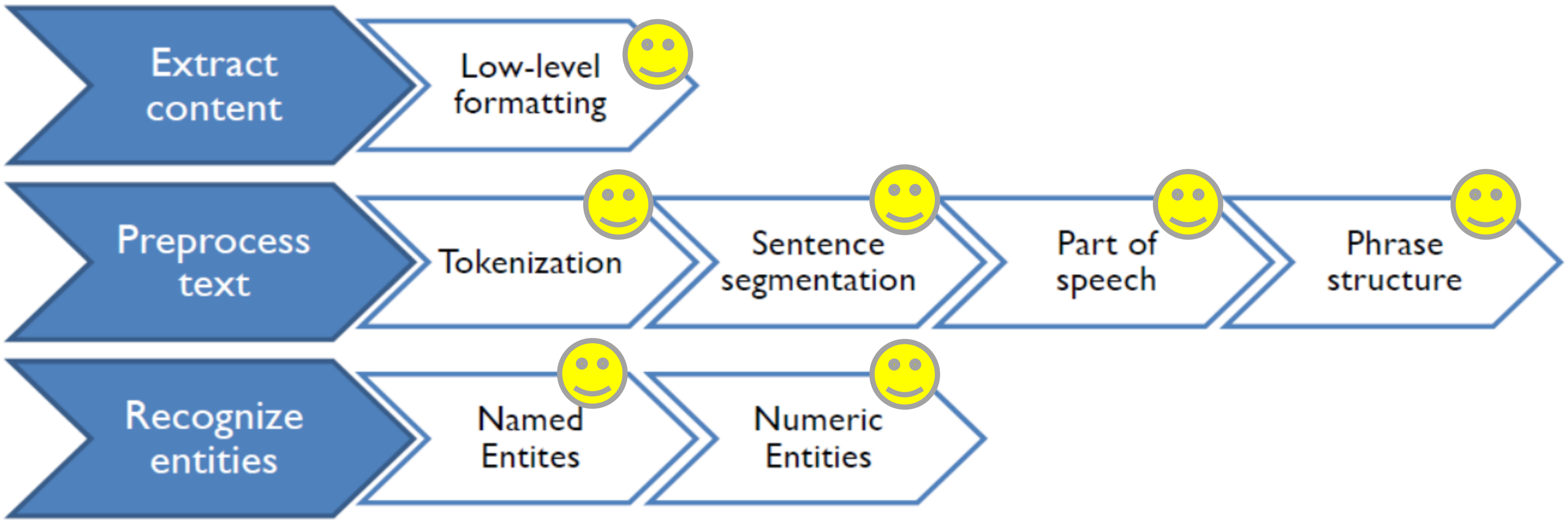
- It is one of the state-of-the-art sequence labeling techniques
- A conditional random field (CRF) is a type of discriminative probabilistic model most often used for the labeling of sequential data
- Based on feature functions
- CRF is more powerful than HMM.

- **Training:** Some sequences are specified, from which **both the input and the desired output is known**. The learning process then attempts **to adjust the parameters** in the CRF so that for as many sequences in the training data, the correct output sequence is predicted.

- Detailed definitions, descriptions, and proofs can be found from the following book:

  Sutton, C., McCallum, A.: *An Introduction to Conditional Random Fields for Relational Learning*. In "Introduction to Statistical Relational Learning". Edited by Lise Getoor and Ben Taskar. MIT Press. (2006).

NER - Pipeline



Evalution

How can the performance of a system be evaluated?

Standard Methodology from Information Retrieval:

- **Precision:** Probability that a retrieved document is relevant.

$$\text{Precision} = \frac{tp}{tp + fp}$$

- **Recall:** Probability that a relevant document is retrieved in a search.

$$\text{Recall} = \frac{tp}{tp + fn}$$

- **F-measure** (combination of Precision/Recall)

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Review: State of the Art

- Tokenization: ~ 99%
- Sentence segmentation: ~ 99%
- POS tagging: ~ 97% correct
  - Caution: 20+ words / sentence
  - still 1 tag error / sentence
- Full parsing (F =~ 90%)
- Chunking (F =~ 95%)
- NER
  - English: F =~ 93% (vs. humans: F =~ 97%)
  - German: F =~ 70% (bad recall)

For further reading

- *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, by Dan Jurafsky, James H. Martin, Prentice Hall, 2009
- *Introduction to Information Extraction Technology: A Tutorial*, Prepared for IJCAI-99 by Douglas E. Appelt and David J. Israel, http://www.grupolys.org/docencia/ln/biblioteca/ie.pdf
- *Information Extraction: A survey*, by Sunita Sarawagi http://osm.cs.byu.edu/CS652s09/papers/Sarawagi.ieSurvey.pdf
- *Introduction to Information Retrieval*, by Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Cambridge University Press 2008