Projektgruppe

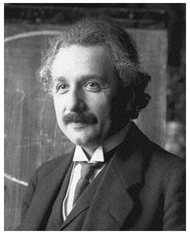
IDSE Information-Driven Software Engineering

Michael Meier

Named-Entity-Recognition Pipeline

What is Named-Entitiy-Recognition?

- Named-Entity
  - Nameable objects in the world, e.g.:
    - Person: Albert Einstein
    - Organization: Deutsche Bank
    - Location: Paderborn
- Named-Entity-Recognition (NER)
  - Find named entities in a text
  - Assign predefined categories
  - Most popular IE task

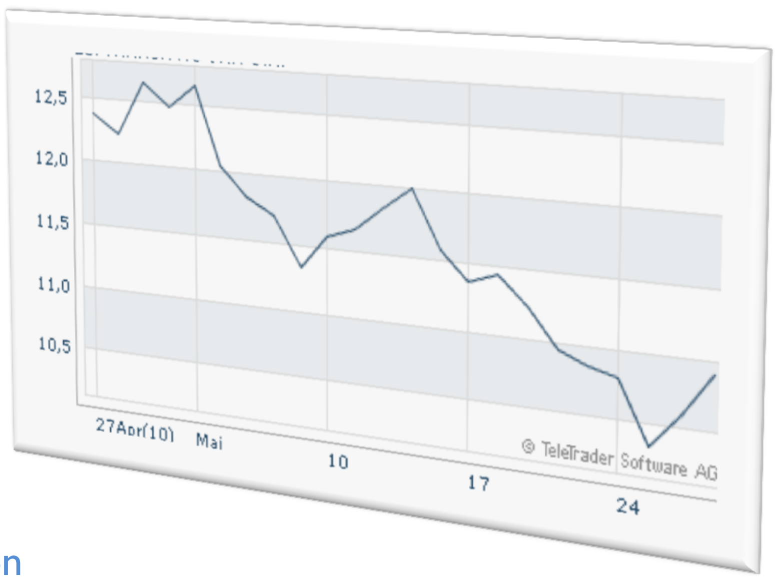


Deutsche Bank

Motivation

08.05.2010
Köln – Die Deutsche Lufthansa AG verbesserte ihren Umsatz im abgelaufenen Quartal auf 5,76 Mrd. Euro, nachdem man im Vorjahresquartal einen Umsatz von 5,02 Mrd. Euro generiert hatte. (finanzen.net)

25.05.2010
Bei der Lufthansa werden die Flugscheine teurer. Weltweit würden die Ticketpreise im Passagiergeschäft zum 01. Juni im Schnitt um 4,8 Prozent anziehen. (Focus Online)



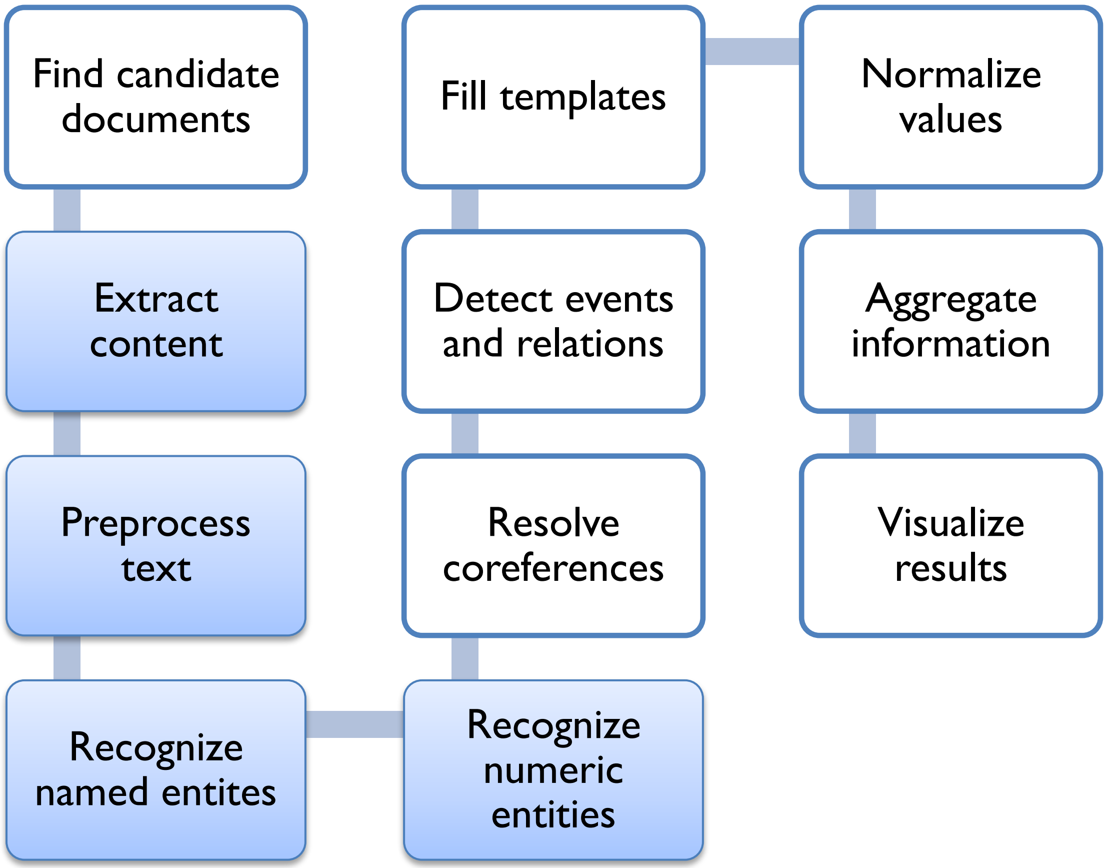
Motivation

08.05.2010
Köln – Die Deutsche Lufthansa AG verbesserte ihren Umsatz im abgelaufenen Quartal auf 5,76 Mrd. Euro, nachdem man im Vorjahresquartal einen Umsatz von 5,02 Mrd. Euro generiert hatte. (finanzen.net)

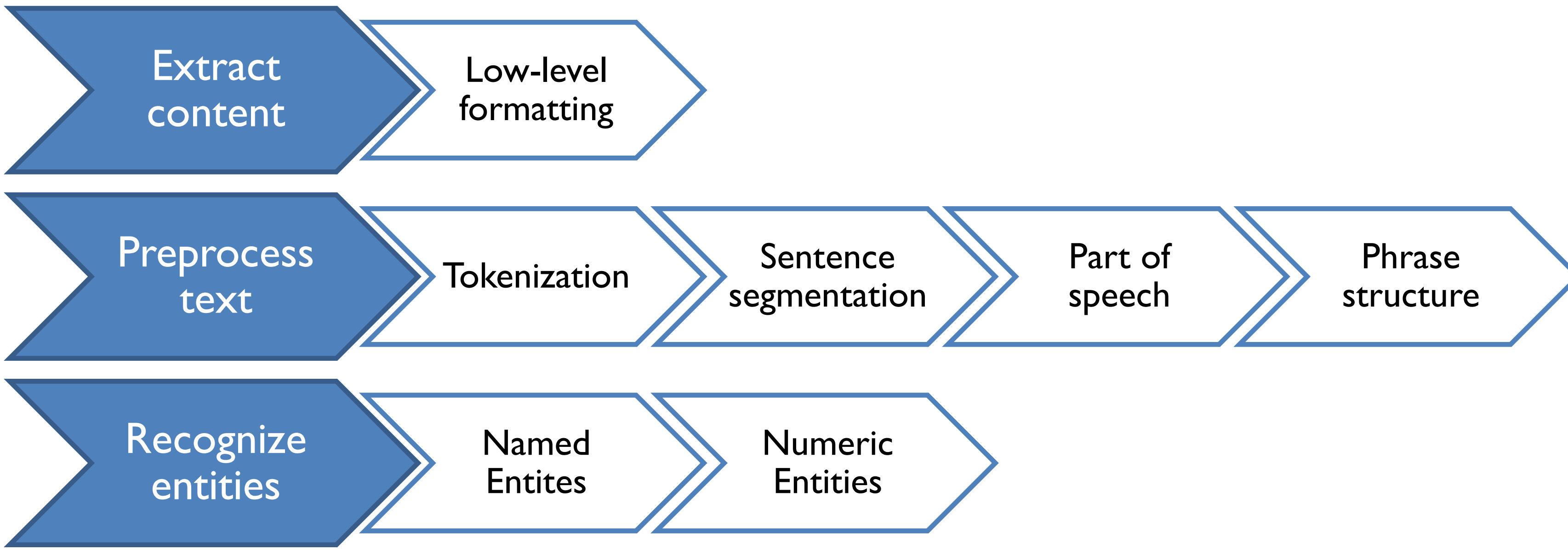
25.05.2010
Bei der Lufthansa werden die Flugscheine teurer. Weltweit würden die Ticketpreise im Passagiergeschäft zum 01. Juni im Schnitt um 4,8 Prozent anziehen. (Focus Online)

Information Extraction (IE) Pipeline

- Named-Entity-Recognition (NER) Pipeline in IE context



Outline NER Pipeline



- Agenda
  - What is done?
  - Problems
  - Algorithm/Approach

IDSE Information-Driven Software Engineering

Low-level formatting

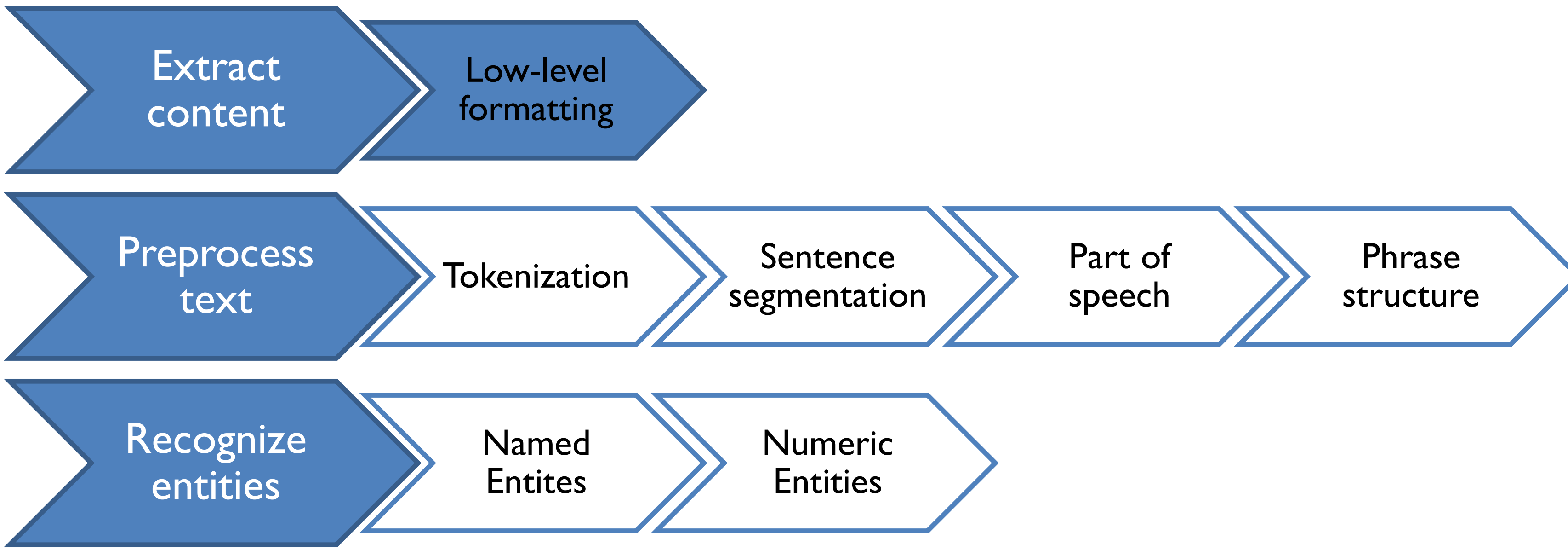
- **Input:** Raw text in electronical form
  - Documents (pdf, doc, rtf)
  - Websites (html)

➔ Analyze connected text

- Irrelevant content (Junk content)
  - Pictures
  - Tables
  - Diagrams
  - Advertisements

➔ Remove before any further processing

Outline NER Pipeline



Tokenization

- **Input:** (Junk-free) connected text
- **Step:** 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?
  - Main clue in English: Words separated by whitespaces / linebreaks

Einstein was born in 1879. ➔

Einstein	was	born	in	1879	.

Tokenization: Problems

- Punctuation marks
  - Einstein was born in 1879.
    - 1879.
    - 1879 | .
  - The car costs $34,250.99, the bike…
    - $ | 34 | , | 250 | . | 99 | ,
    - $34,250.99 | ,
- Hyphenation
  - I am in the uni-
    versity.
    - uni | - | versity
    - university
  - e-mail
    - email
    - e-mail
  - the New York-New Haven railroad
    - New | York-New | Haven
    - New York | New Haven
- Lots more: Apostrophes, direct speech, etc.

Tokenization: Example Script (1)

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 |

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

Tokenization: Example Script (2)

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.

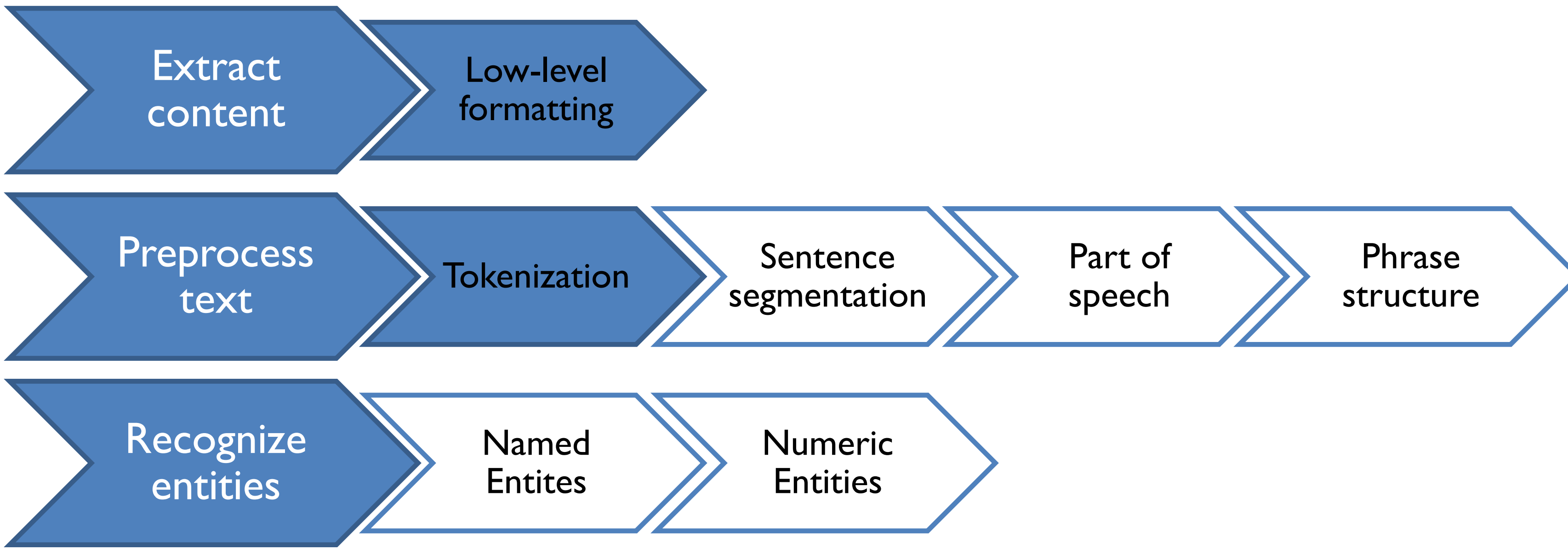
The	U.S.A.	have	309	mil.	inhabitans	.

Step 6: Expand clitics

Example: […] I 'll do […]

I	will	do

Outline NER Pipeline



Sentence segmentation

- **Input:** Connected text
- **Step:** Divide text into sentences
- What is a sentence?
  - Main clue in English: 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: Heuristic Sentence Boundary Detection Algorithm (1)

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."|

Sentence segmentation: Heuristic Sentence Boundary Detection Algorithm (2)

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)

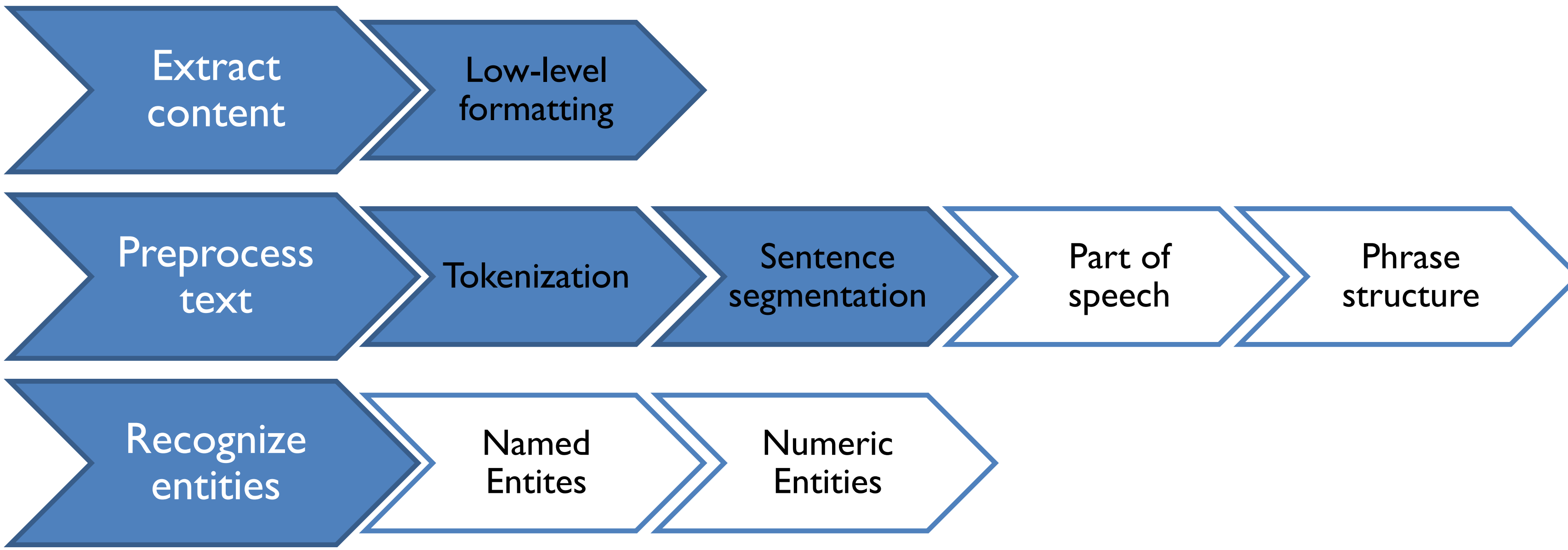
Prof. Smith Jr. said: "Google Inc. and Yahoo!| etc. are search engine providers."|

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."|

Result: Regard temporary sentence boundaries as final sentence boundaries.

Outline NER Pipeline



Part of speech

- **Input**: Tokenized text with sentence boundaries
- **Step**: Group tokens into classes with similar syntactic behavior

➔ Part of speech (POS)

Open classes	Closed classes

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

Part of speech: Morphology (1)

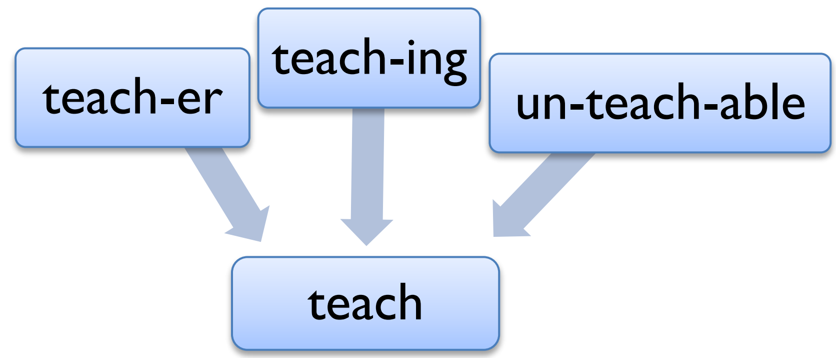
- Natural language is complex
- Grammatical distinction for words (context dependency)

→ Inflection: Systematic modification of a root form

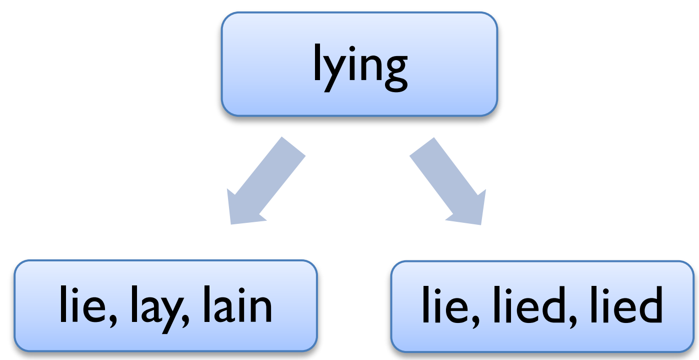
Nouns	Verbs	Adjectives
• Number inflection • Gender inflection • Case inflection	• Subject number • Subject person • Tense • etc.	• Positive • Comparative • Superlative

Part of speech: Morphology (2)

- Stemming
  - Strip off affixes from words
  - ➔ Stem

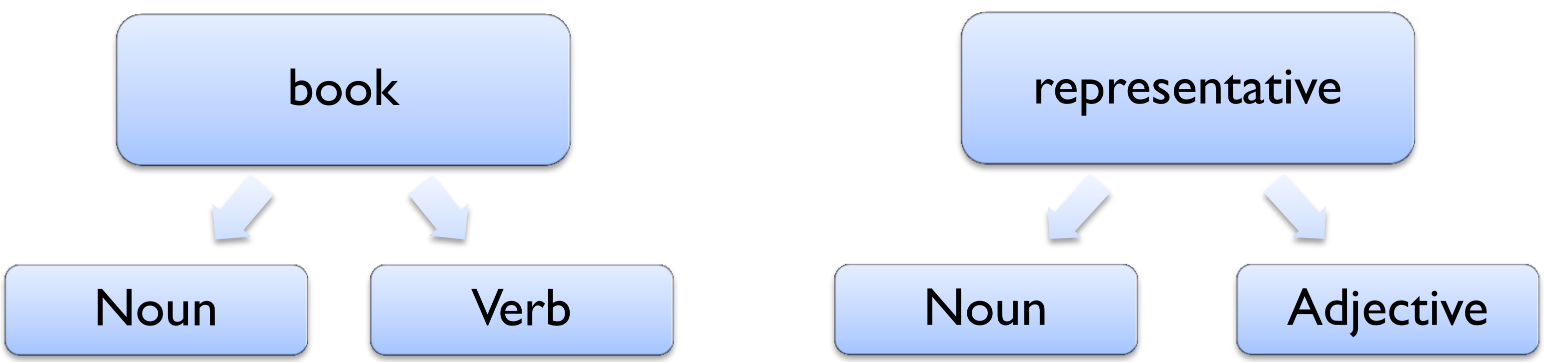


- Lemmatization
  - Find lemma of inflected form



Part of speech: Problems

- Ambiguity
  - Words have more than one syntactic category
  - Depends on context



Part of speech: Tagging

- **Step**
  - Group tokens into classes with similar syntactic behavior

  → Assign part of speech tag to each token

- Examples

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

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

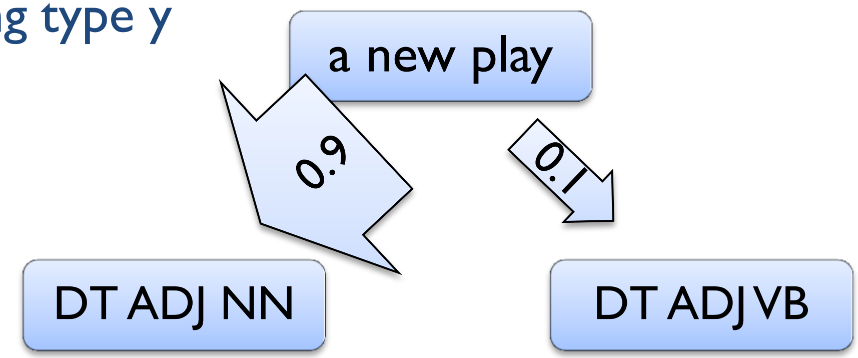
Part of speech: Tagset example

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	“to”	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential ‘there’	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	$
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one’s*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

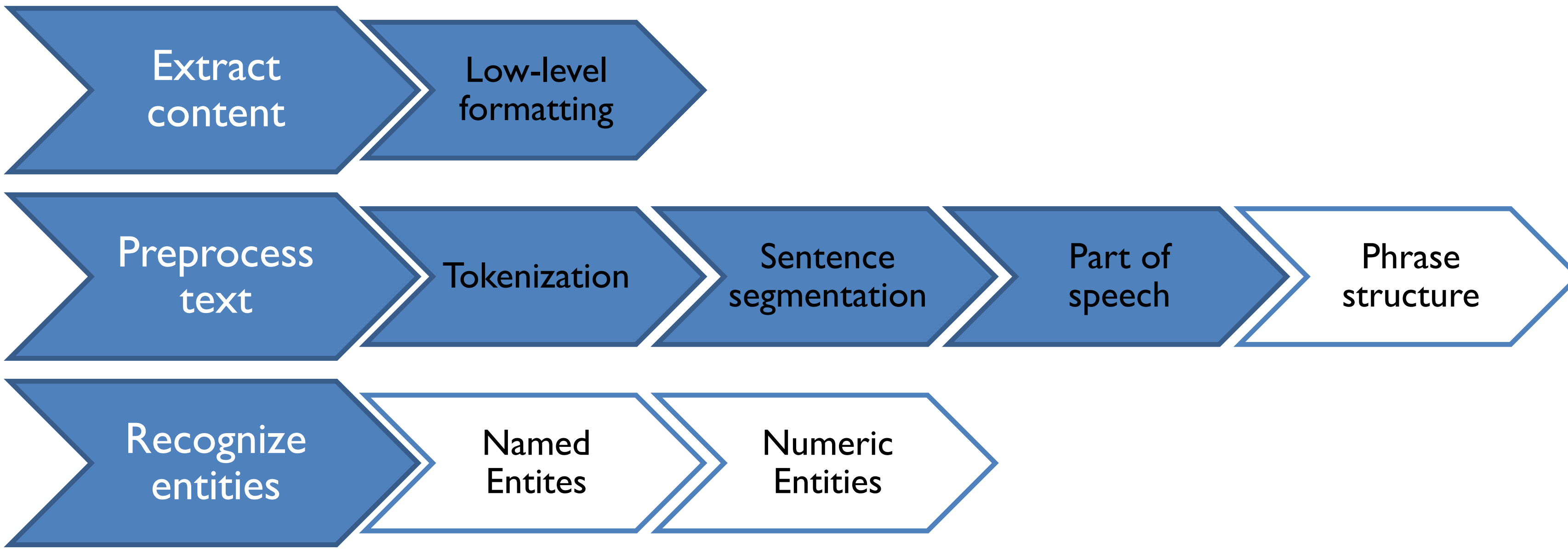
Figure 5.6 Penn Treebank part-of-speech tags (including punctuation).

Part of speech: Probabilistic approach

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

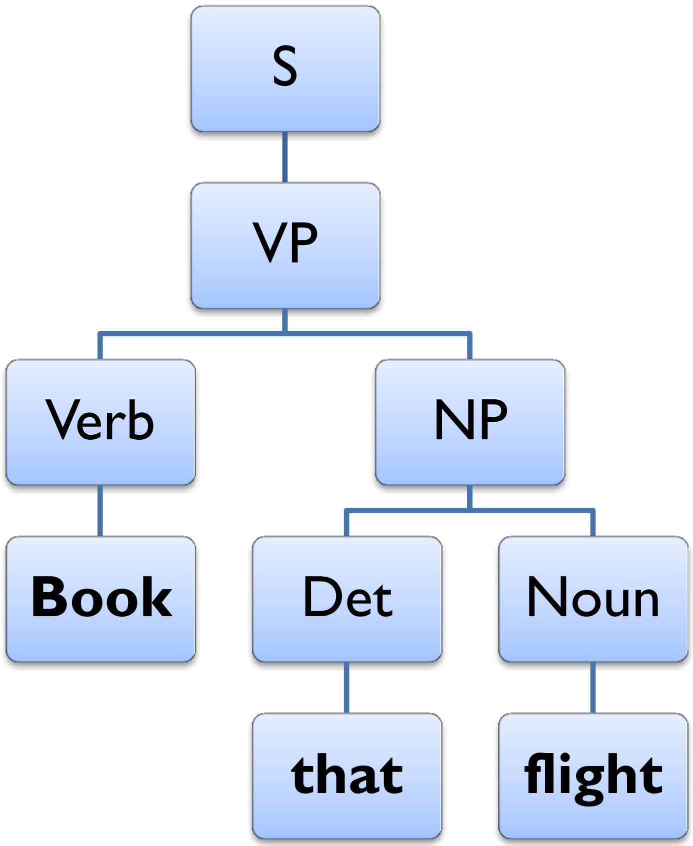


Outline NER Pipeline



Phrase structure

- **Input:** Tokenized text with POS tags and sentence boundaries
- **Step:** Assign (full) syntactic structure to sentences
- Sentence groups
  - Noun phrases (NP)
  - Prepositional phrases (PP)
  - Verb phrases (VP)
  - Adjective phrases (AP)
- Represented by context-free grammar

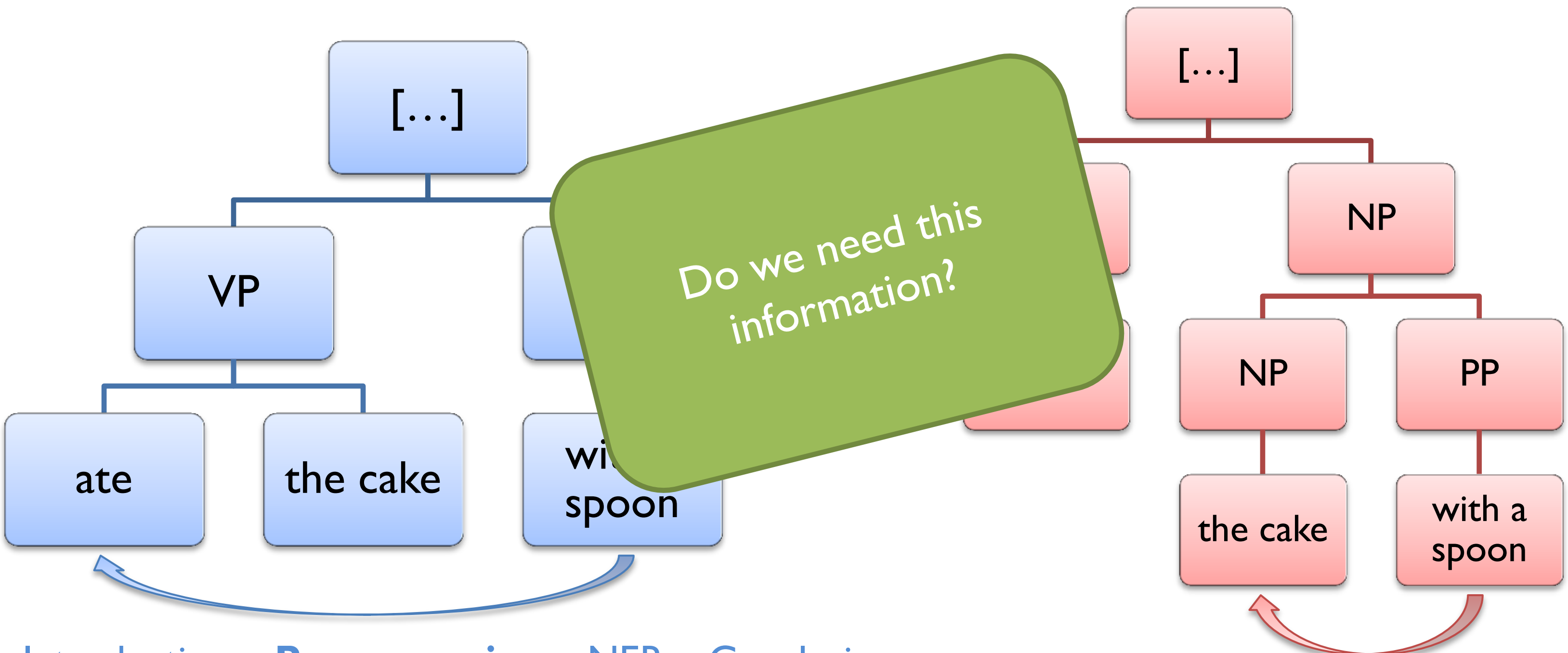


Phrase structure: Problems

- Slow computation (for longer sencences)
- Context-free grammar is ambiguous

➔ Ambiguous phrase structure

- Example: I ate the cake with a spoon.



Chunking

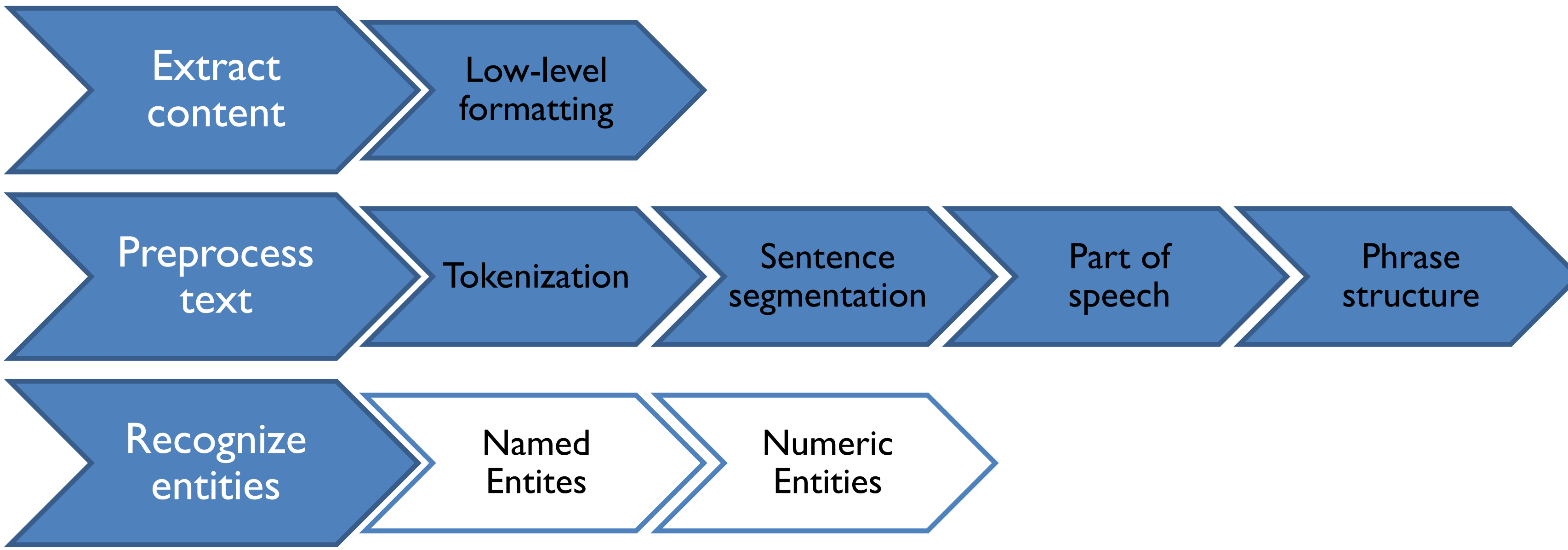
- Goal revisited
  - Find named entities
  - No need for full parse tree
  - Solution: Partial parsing
- Chunking
  - Idea: Identify and classify basic phrases of a sentence
  - Example:

    [**NP** I] [**VP** ate] [**NP** the cake] [**PP** with] [**NP** a spoon].

Chunking: Approach

- Rule-based
  - NP → (DT) NN* NN
  - NP → NNP
  - VP → VB
  - VP → Aux VB
- Supervised machine learning
  - Sequence labeling

Outline NER Pipeline



- Conclusion of preprocessing
  - Several steps based on each other
  - → Error-prone
  - → Inaccuracies affect all subsequent results

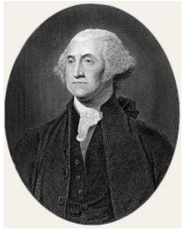
Recognize entities

- **Input:**
  - Some sort of preprocessed text
  - Not all steps described before have to be applied
  - Depends on implementation
- **Step:**
  - Find named entities
  - Assign predefined categories

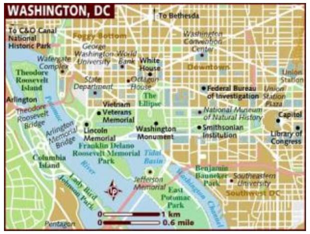
NER Categories

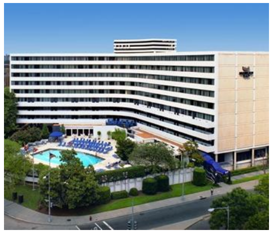
- Generic Named Entities
  - Person (PER)
  - Organization (ORG)
  - Location (LOC)
- Custom Named Entities
  - Biological context: Proteins, genes
  - IDSE context?
    - Class diagrams: Class, Attribute, Relation
    - Sequence diagrams: Boundary, Control, Entity
- Numeric Entities
  - Temporal expressions (TIME)
  - Numerical expressions (NUM)

NER Ambiguity Problem



- **Washington** was born in 1732.
  - ➔ Person (PER)
- **Washington** has won the last game against Denver.
  - ➔ Organization (ORG)
- Blair arrived in **Washington** for a state visit.
  - ➔ Location (LOC)
- **Washington** just passed a primary seatbelt law.
  - ➔ Geo-Polictal Entity (GPE)
- The **Washington** is stationed in Yokosuka, Japan.
  - ➔ Vehicle (VEH)
- Tourists prefer to stay at the **Washington**.
  - ➔ Faculty (FAC)





NER Approach

- Algorithm: Sequence labeling
- General features (pre-processing)
  - Lexical features (words and lemmas)
  - Syntactic features (part-of-speech)
  - Chunking
- NER-specific features
  - Shape feature
  - Presence in named entity list
  - Predictive words
  - Lots more…

Shape feature

- Orthographic pattern of the target word
  - Case distinction (Upper case, lower case, capitalized forms)
  - More elaborate patters
  - ➔ Regular expressions
- Examples
  - Two adjacent capitalized words in middle of text ➔ Person
  - Regular expression /[AB][0-9]{1,3}/
    - German street name
    - Car name
    - Size of a paper
    - ➔ Domain dependency

Presence in a named entity list (1)

- Extensive lists of names for a specific category
  - PER: First names and surnames
    - United States Census Bureau, e.g. 1990er Census
      - Male first names: 1,219
      - Female first names: 4,275
      - Surnames: 88,799
  - LOC: Gazetteers (= Ortslexikon)
    - Place names (countries, cities, mountains, lakes, etc.)
    - GeoNames database (www.geonames.org), > 8 Mio. placenames
  - ORG
    - Yellow Pages

Presence in a named entity list (2)

- Disadvantages
  - Difficult to create and maintain (or expensive if commercial)
  - Usefulness varies depending on category [Mikheev]
    - Quite effective: Gazetteers
    - Not nearly as beneficial: Lists of persons and organizations
  - Ambiguity
    - Remember „Washington“-Example
    - Would occur in more lists of different types (PER, LOC, FAC,…)

Predictive words

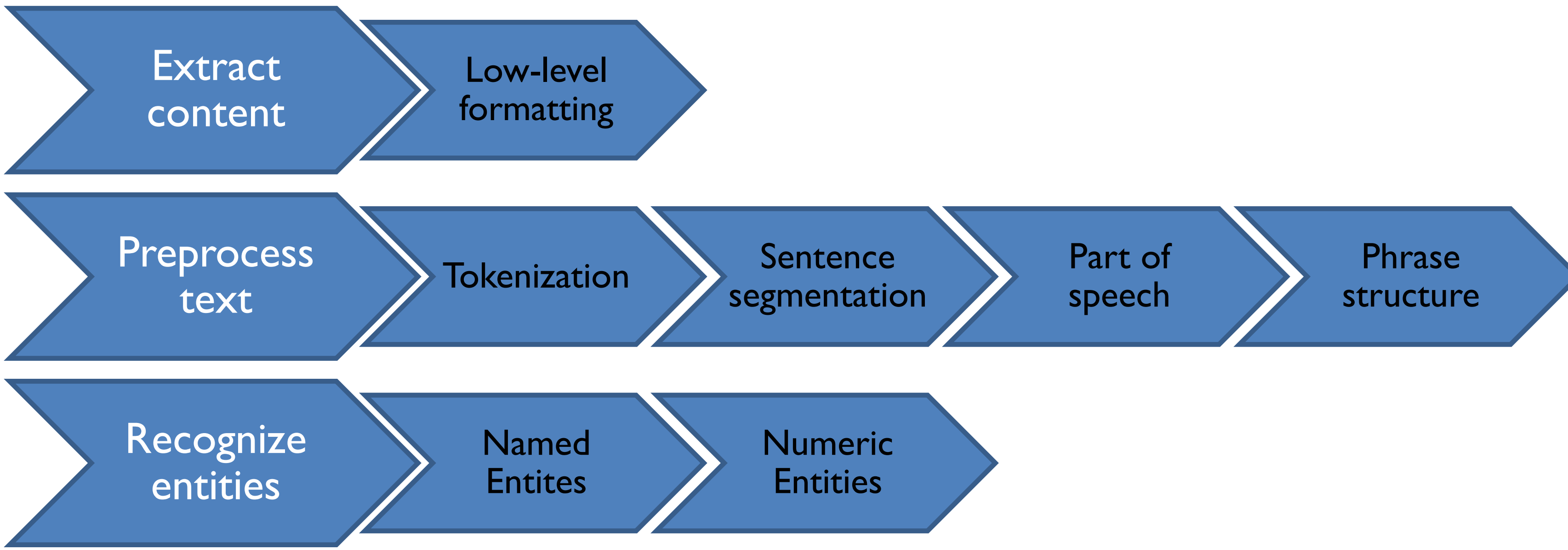
- Predictive words in surrounding text
- Can accurately indicate class of an entity
- Examples
  - John Smith M.D. announced his new health care plans.
  - Google Inc. has made a revenue of $22 billon in 2009.
  - I met John Smith last week in Ridgewood, NY.
- Advantages (in contrast to gazetteers)
  - Relatively short lists and stable over time
  - Easy to develop and maintain

Overview of other features

- Classification
  - "Washington" in sports article
  - Higher probability of category ORG & LOC
- Affixes
  - German: Mei-er, Beck-er, Müll-er ➔ PER
  - Other languages: Podol-ski, Hender-son, O'Neill ➔ PER
- And so on…

➔ "Invent" your own feature for your domain

Outline NER Pipeline



- Conclusion NER
  - Several features combined
  - Based on implementation
  - Usefulness of any feature ➔ Greatly domain- and language-dependent

Review: Language dependency

- Tokenization
  - Chinese & Japanese: Words not separated
- Part of speech
  - Nouns
    - English: only number inflection
    - German: number, gender and case inflection
  - Verbs
    - English: regular verb 4, irregular verb up to 8 distinct forms
    - Finnish: more than 10,000 forms
- NER: Shape feature
  - English: Only proper nouns capitalized
  - German: All nouns capitalized

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)

Conclusion

- Named Entity Recognition
  - Linguistic preprocessing necessary
  - Several features combined
- Usage for IDSE
  - Why recognize Named Entities?
  - → Extract semantic information (talk of Mirko)
  - → Transform Requirement-Documents (talk of Othmane)

The end…

Questions?