Information School
UNIVERSITY OF WISCONSIN–MADISON

Text Normalization and Parsing

The Information School, UW-Madison

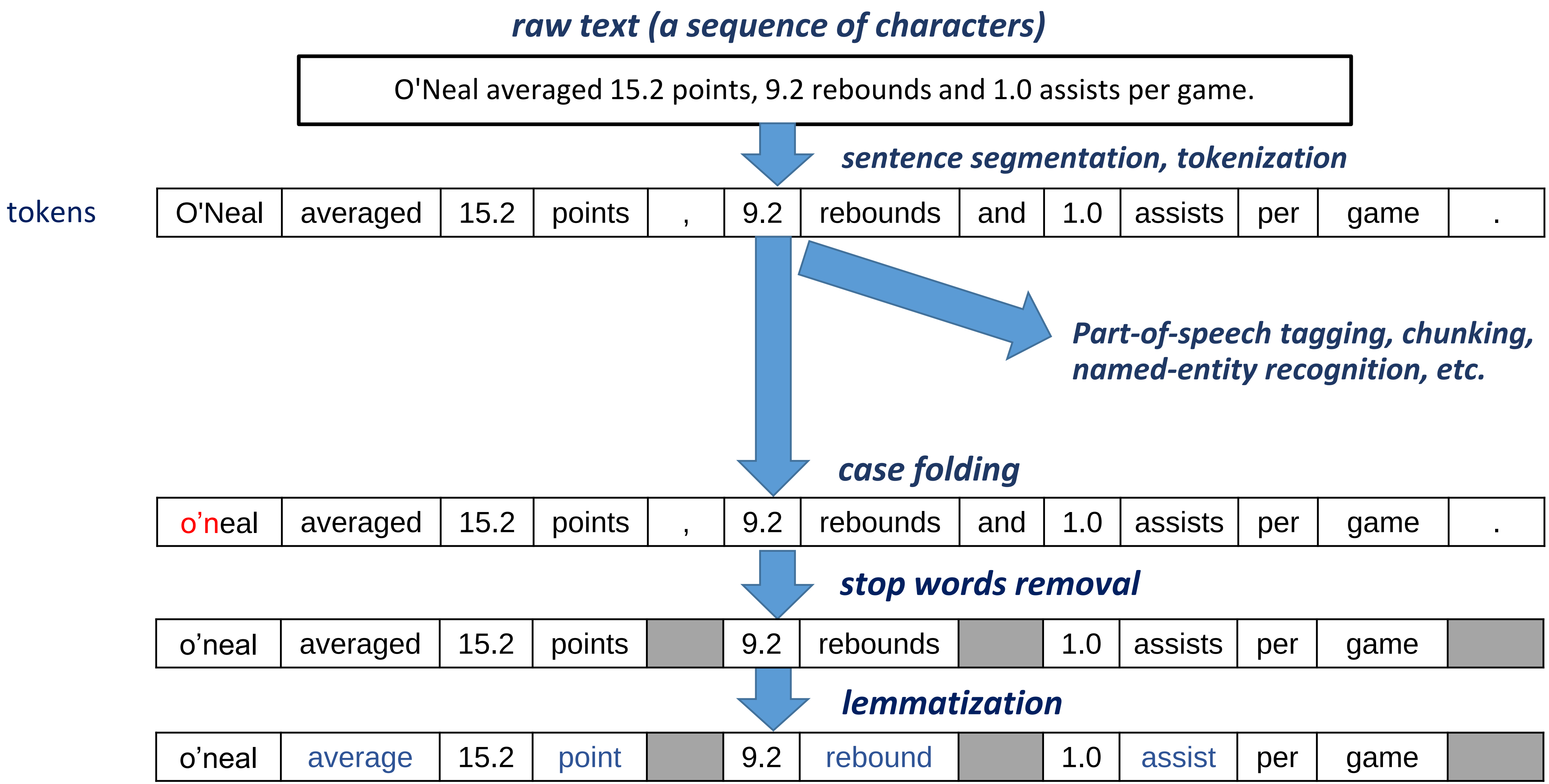
Introduction

Text normalization and parsing:

- clean and regularize texts depending on your needs
- recognize meaningful units and structures of texts

Today, many NLP tools can perform text normalization and parsing automatically with reasonably high accuracy on well-understood text data (e.g., news articles).

A Typical Text Normalization and Parsing Pipeline



raw text (a sequence of characters)

O'Neal averaged 15.2 points, 9.2 rebounds and 1.0 assists per game.

sentence segmentation, tokenization

tokens

O'Neal	averaged	15.2	points	,	9.2	rebounds	and	1.0	assists	per	game	.

Part-of-speech tagging, chunking, named-entity recognition, etc.

case folding

o’neal	averaged	15.2	points	,	9.2	rebounds	and	1.0	assists	per	game	.

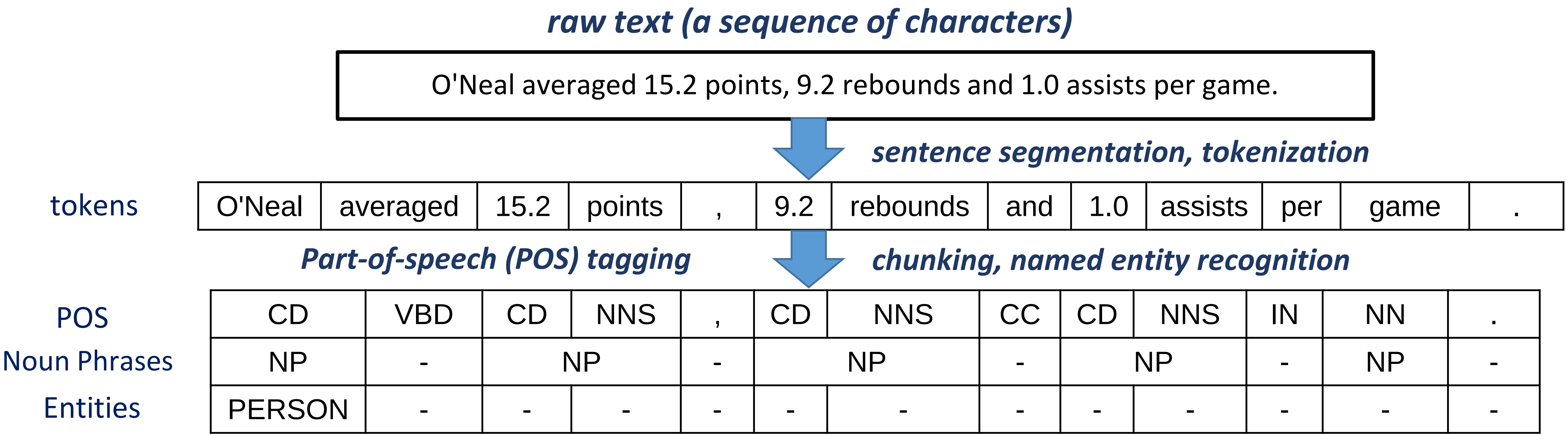
stop words removal

o’neal	averaged	15.2	points	▨	9.2	rebounds	▨	1.0	assists	per	game	▨

lemmatization

o’neal	average	15.2	point	▨	9.2	rebound	▨	1.0	assist	per	game	▨

A Typical Text Normalization and Parsing Pipeline



raw text (a sequence of characters)

O'Neal averaged 15.2 points, 9.2 rebounds and 1.0 assists per game.

sentence segmentation, tokenization

tokens	O'Neal	averaged	15.2	points	,	9.2	rebounds	and	1.0	assists	per	game	.

Part-of-speech (POS) tagging

chunking, named entity recognition

POS	CD	VBD	CD	NNS	,	CD	NNS	CC	CD	NNS	IN	NN	.
Noun Phrases	NP	-	NP		-	NP		-	NP		-	NP	-
Entities	PERSON	-	-	-	-	-	-	-	-	-	-	-	-

Text Normalization

1. **Tokenizing (segmenting words)**

2. **Normalizing word formats**

3. **Segmenting sentences**

The US is a big nation. Americans love the U.S.A. a lot. They like to drive their cars around the country. They measure speed in m.p.h and not km.p.h.

Text Normalization

1. **Tokenizing (segmenting words)**

2. Normalizing word formats

3. Segmenting sentences

‘The’, ‘US’, ‘is’, ‘a’, ‘big’, ‘nation’, ‘.’, ‘Americans’, ‘love’, ‘the’, ‘U’, ‘.’, ‘S’, ‘.’, ‘A’, ‘.’, ‘a’, ‘lot’, ‘.’, ‘They’, ‘like’, ‘to’, ‘drive’, ‘their’, ‘cars’, ‘around’, ‘the’, ‘country’, ‘.’, ‘They’, ‘measure’, ‘speed’, ‘in’, ‘m’, ‘.’, ‘p’, ‘.’, ‘h’, ‘and’, ‘not’, ‘km’, ‘.’, ‘p’, ‘.’, ‘h’, ‘.’

Text Normalization

1. Tokenizing (segmenting words)

2. **Normalizing word formats**

3. Segmenting sentences

‘The’, **‘US’**, ‘is’, ‘a’, ‘big’, ‘nation’, ‘.’, ‘Americans’, ‘love’, ‘the’, **‘U’, ‘.’, ‘S’, ‘.’, ‘A’**, ‘.’, ‘a’, ‘lot’, ‘.’, ‘They’, ‘like’, ‘to’, ‘drive’, ‘their’, ‘cars’, ‘around’, ‘the’, ‘country’, ‘.’, ‘They’, ‘measure’, ‘speed’, ‘in’, ‘m’, ‘.’, ‘p’, ‘.’, ‘h’, ‘and’, ‘not’, ‘km’, ‘.’, ‘p’, ‘.’, ‘h’, ‘.’

Text Normalization

1. Tokenizing (segmenting words)

2. **Normalizing word formats**

3. Segmenting sentences

‘The’, **‘US’**, ‘is’, ‘a’, ‘big’, ‘nation’, ‘.’, ‘Americans’, ‘love’, ‘the’, **‘US’**, ‘.’, ‘a’, ‘lot’, ‘.’, ‘They’, ‘like’, ‘to’, ‘drive’, ‘their’, ‘cars’, ‘around’, ‘the’, ‘country’, ‘.’, ‘They’, ‘measure’, ‘speed’, ‘in’, ‘m’, ‘.’, ‘p’, ‘.’, ‘h’, ‘and’, ‘not’, ‘km’, ‘.’, ‘p’, ‘.’, ‘h’, ‘.’

Text Normalization

1. Tokenizing (segmenting words)
2. Normalizing word formats
3. **Segmenting sentences**

‘The’, ‘US’, ‘is’, ‘a’, ‘big’, ‘nation’, ‘.’, ‘Americans’, ‘love’, ‘the’, ‘US’, ‘.’, ‘a’, ‘lot’, ‘.’, ‘They’, ‘like’, ‘to’, ‘drive’, ‘their’, ‘cars’, ‘around’, ‘the’, ‘country’, ‘.’, ‘They’, ‘measure’, ‘speed’, ‘in’, ‘m’, ‘.’, ‘p’, ‘.’, ‘h’, ‘and’, ‘not’, ‘km’, ‘.’, ‘p’, ‘.’, ‘h’, ‘.’

Tokenization & Sentence Segmentation

Computers store text data just a sequence of characters …

Tokenization: chunk a text into “tokens” (the smallest unit of analysis in most cases)

- A token can be a word, a number, a punctuation, a chemical compound, a gene, etc.
- Not as simple as chunking texts by whitespace and other non-alphabetical symbols…
- Apostrophe? e.g., Shaquille O’Neal
- Comma? 1,600 feet high
- Hyphen? C-3PO, R2-D2
- devansh.saxena@wisc.edu, 123-456-7000, devansh_saxena
- No rules are smart enough to cover all cases …

Sentence Segmentation: segment a text into sentences

- rules + exceptions
- Is it a sentence separator or a part of a meaningful token? Ms. Dr. Yahoo!

Word Tokenization (segmenting text into words)

Whitespace split

- Split sentences by whitespaces
- Replace punctuations with spaces
- Replace special characters with space

Pros

- Simple to implement
- Effective for many basic NLP tasks

Cons:

- Removes important punctuations
- Removes special characters

Hello, how are you today?

Word Tokenization (segmenting text into words)

Whitespace split

- Split sentences by whitespaces
- Replace punctuations with spaces
- Replace special characters with space

[Hello, how , are , you , today]

Pros

- Simple to implement
- Effective for many basic NLP tasks

Cons:

- Removes important punctuations
- Removes special characters

Word Tokenization (segmenting text into words)

Whitespace split

- Split sentences by whitespaces
- Replace punctuations with spaces
- Replace special characters with space

Pros

- Simple to implement
- Effective for many basic NLP tasks

Cons:

- Removes important punctuations
- Removes special characters

Parking is $4.50/hour at UW-Madison

Word Tokenization (segmenting text into words)

Whitespace split

- Split sentences by whitespaces
- Replace punctuations with spaces
- Replace special characters with space

[Parking , is , 4 , 50 , hour , at , UW , Madison]

Pros

- Simple to implement
- Effective for many basic NLP tasks

Cons:

- Removes important punctuations
- Removes special characters

Word Tokenization (using Penn Treebank)

Penn Treebank tokenization

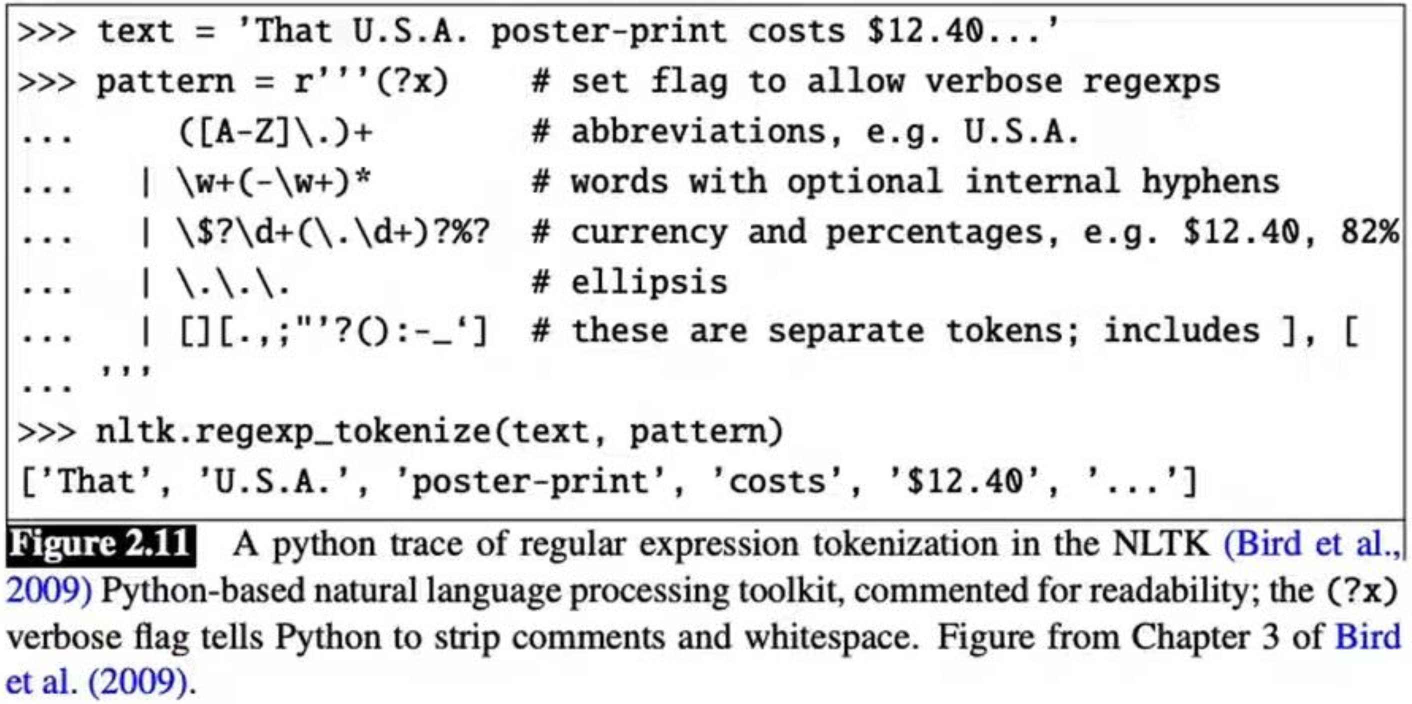
- Commonly used tokenization standard
- Created by Linguistic Data Consortium

Pros

- Understands important punctuations
- Keeps hyphenated words
- Separates unnecessary punctuations

Cons:

- Requires additional post-processing
- Separates out special characters



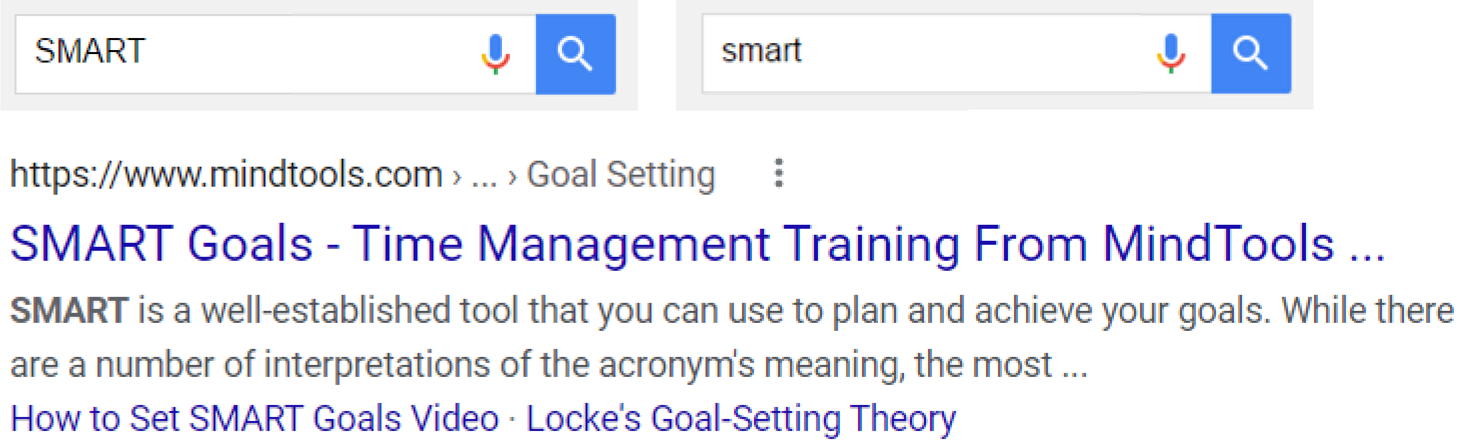
```
>>> text = 'That U.S.A. poster-print costs $12.40...'
>>> pattern = r'''(?x)    # set flag to allow verbose regexps
...     ([A-Z]\.)+        # abbreviations, e.g. U.S.A.
...   | \w+(-\w+)*        # words with optional internal hyphens
...   | \$?\d+(\.\d+)?%?  # currency and percentages, e.g. $12.40, 82%
...   | \.\.\.           # ellipsis
...   | [][.,;"'?():-_`]  # these are separate tokens; includes ], [
... '''
>>> nltk.regexp_tokenize(text, pattern)
['That', 'U.S.A.', 'poster-print', 'costs', '$12.40', '...']
```

Figure 2.11 A python trace of regular expression tokenization in the NLTK (Bird et al., 2009) Python-based natural language processing toolkit, commented for readability; the (?x) verbose flag tells Python to strip comments and whitespace. Figure from Chapter 3 of Bird et al. (2009).

Case-folding: lowercasing everything

Case-folding is widely applied to many text information systems …

- e.g., Web search engines returns the same results for “SMART” and “smart”
- It helps regularize words in text (e.g., words at the beginning of a sentence)



SMART

smart

https://www.mindtools.com › ... › Goal Setting

SMART Goals - Time Management Training From MindTools ...

SMART is a well-established tool that you can use to plan and achieve your goals. While there are a number of interpretations of the acronym's meaning, the most ...

How to Set SMART Goals Video · Locke's Goal-Setting Theory

Sometimes letter case may be informative, e.g.,

- **W**ill **S**mith
- the **US** health care system
- He is **ABSOLUTELY** a genius (especially common and important on social media)

Stop words removal

Stop words

- Words that can be ignored in text analysis, e.g., counting words frequencies
- Usually not very informative for representing the topics of texts
- *(but usually helpful for understanding the structures of texts)*
- Usually have very high frequencies (removal can reduce data size significantly…)
- Remove them or not? Depends on needs and text analytics methods…

An example list of stop words

- a, an, and, are, as, at, be, but, by, for, if, in, into, is, it, no, not, of, on, or, such, that, the, their, then, there, these, they, this, to, was, will, with

Lemmatization & Stemming

Purpose

- To categorize words with the same root or lemma
- Plural ⍰ singular, verb (different tenses), adj & adv etc.
- Example: “cats” and “cat”; “search”, “searches”, “searching”

Methods

- Rule-based: defines and performs a set of rules (e.g., suffix stripping)
- Dictionary-based: e.g., can handle exceptions

Porter Stemming

Rule-based, a list of suffix-stripping rules

- Just some examples
  - -sses ⍰ -ss, e.g., caresses ⍰ caress
  - -ies ⍰ -i, e.g., ponies ⍰ poni
  - remove -s, e.g., cats ⍰ cat
  - eed ⍰ ee, e.g., agreed ⍰ agree
  - remove -ed, e.g., plastered ⍰ plaster
  - remove -ing, e.g., motoring ⍰ motor
  - -ational ⍰ -ate, e.g., relational ⍰ relate
  - -tional ⍰ -tion, e.g., conditional ⍰ condition
- Iterative: organization ⍰ organize ⍰ organ ☹
- Cannot handle exceptions
- Sometimes hard to interpret (as the outputs are stems, which may not be words)

Porter Stemming

Rule-based, a list of suffix-stripping rules

- Just some examples
  - -sses -ss, e.g., caresses caress
  - -ies -i, e.g., ponies poni
  - remove -s, e.g., cats cat
  - eed ee, e.g., agreed agree
  - remove -ed, e.g., plastered plaster
  - remove -ing, e.g., motoring motor
  - -ational -ate, e.g., relational relate
  - -tional -tion, e.g., conditional condition
- Iterative: organization organize organ ☹
- Cannot handle exceptions
- Sometimes hard to interpret (as the outputs are stems, which may not be words)

Krovetz Stemming: Rule + Dictionary

by Robert Krovetz

- R. Krovertz. Viewing morphology as an inference process. SIGIR 1993.

Use of dictionary to handle exceptions

- Large dictionary of “head words” in a dictionary, e.g., lists of country names and nationalities, proper nouns, etc.
- If a term is a head word, do not stem it
  - *policy ≠ police* and *gravity ≠ grave* and *marbled ≠ marble*
- If it appears as an entry, convert to the headword
- Otherwise, fall back to Porter-like rule-based approach

Stems generated by Krovetz stemming are always actual words

Porter and Krovetz Stemming

Original	Porter (rule-based)	Krovetz (dictionary-based)
communities	commun	community
generated	gener	generate
significantly	significantli	significant
successfully	successfulli	successful
additionally	addition	additional
relatives	rel	relative
internationally	internation	international
importantly	importantli	important
laos	lao	laos
computers	comput	computer
proceeds	proce	proceeds
contents	content	contents
safer	safer	safe

Examples of stemming “errors”

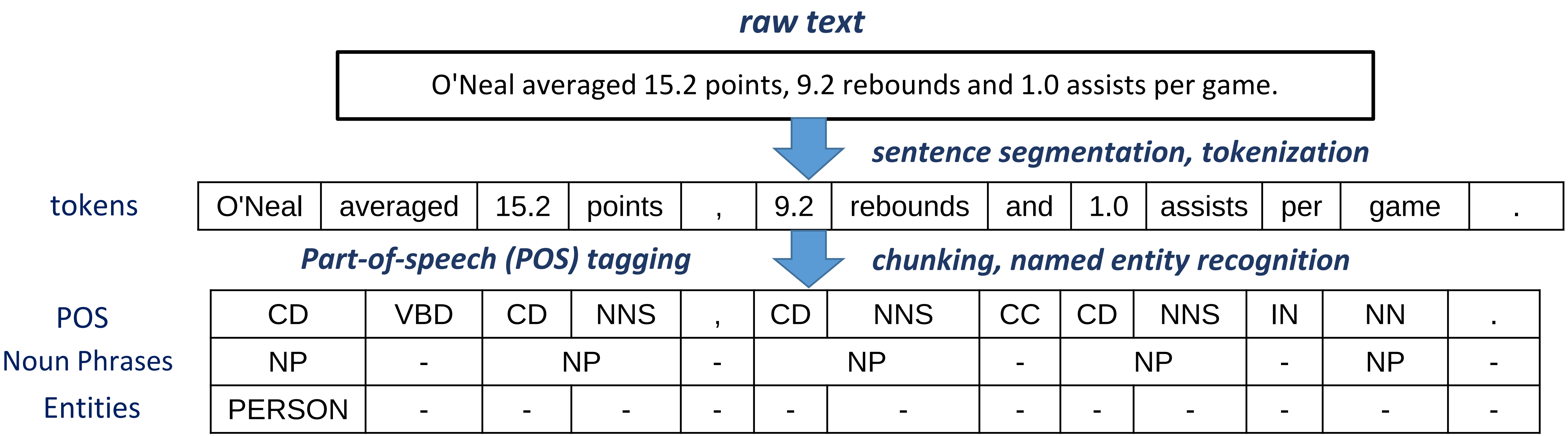
Overstemming

Original	Porter (rule-based)	Krovetz (dictionary-based)
organization	organ	organization
organ	organ	organ
heading	head	heading
head	head	head

Understemming

Original	Porter (rule-based)	Krovetz (dictionary-based)
european	european	europe
europe	europ	europe
urgency	urgenc	urgent
urgent	urgent	urgent

A Typical Text preprocessing and parsing pipeline



raw text

O'Neal averaged 15.2 points, 9.2 rebounds and 1.0 assists per game.

sentence segmentation, tokenization

tokens	O'Neal	averaged	15.2	points	,	9.2	rebounds	and	1.0	assists	per	game	.

Part-of-speech (POS) tagging

chunking, named entity recognition

POS	CD	VBD	CD	NNS	,	CD	NNS	CC	CD	NNS	IN	NN	.
Noun Phrases	NP	-	NP		-	NP		-	NP		-	NP	-
Entities	PERSON	-	-	-	-	-	-	-	-	-	-	-	-

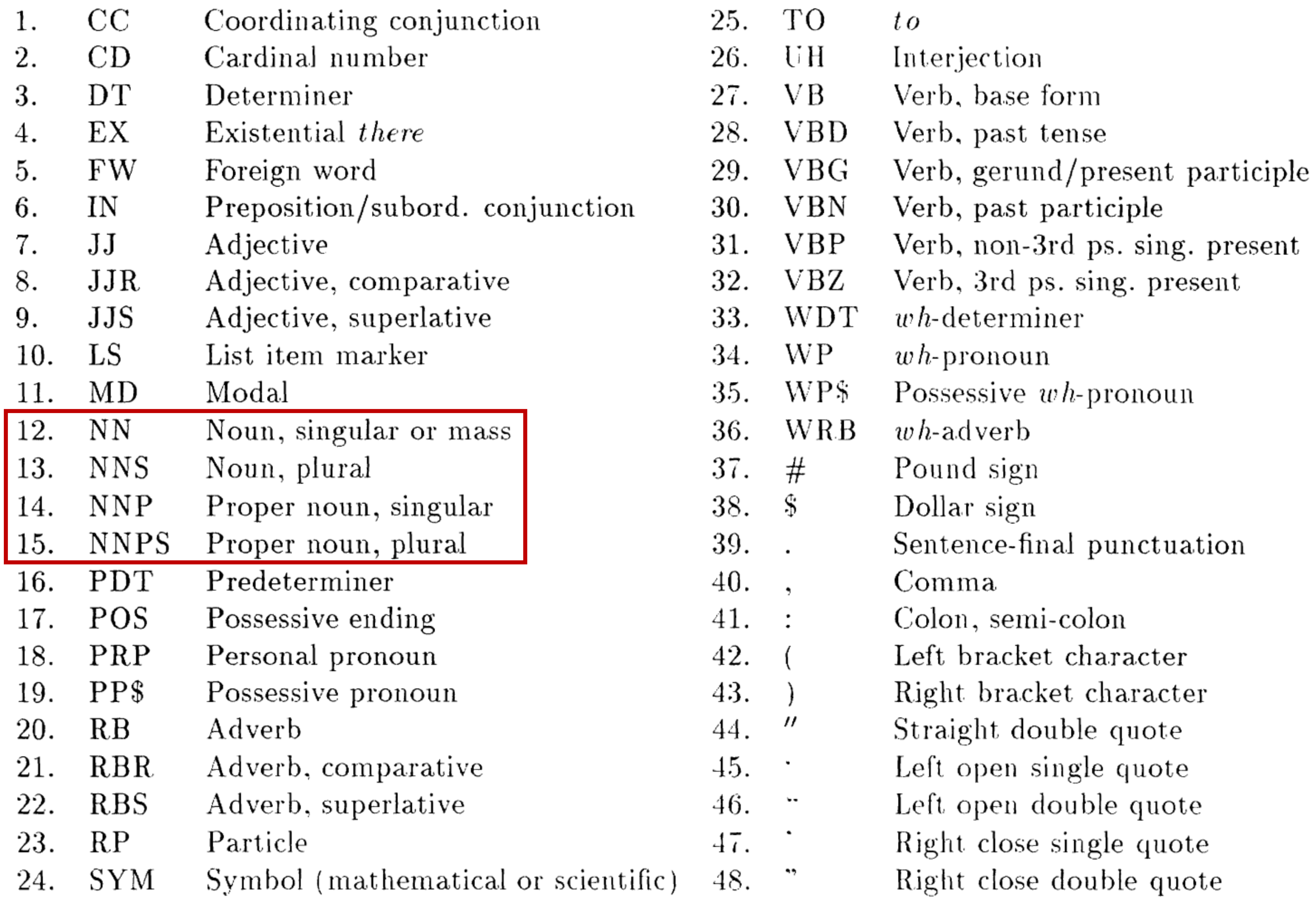
Part of Speech (POS) Tagging

- A part of speech is a category of words that have similar grammatical properties.
  - e.g., noun, pronoun, verb, adjective, etc.
- POS tagging annotates each word in a sentence with a part-of-speech marker.
- Most common POS tags used today is the Penn Treebank POS tagset
  - Fine-grained categories (40+ categories in total)
- The lowest level of syntactic analysis
- Useful for subsequent parsing such as chunking and named entity recognition.

Word token	John	saw	the	saw	and	decided	to	take	it	to	the	table.
POS tag	NNP	VBD	DT	NN	CC	VBD	TO	VB	PRP	IN	DT	NN

The Penn Treebank POS tagset

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. Comput. Linguist. 19, 2 (June 1993), 313–330.



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/subord. 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

POS tagging: Methods and Accuracies

Methods (we'll cover details in the sequential labeling module of this course)

- Train a machine learning model to recognize POS tags of words based on:
- The word itself, e.g., if it is a particular word, the possible part-of-speech in a dictionary
- The word's context (helps resolve ambiguity), e.g., the words before or after it
    - I **like** candy. (verb)
    - Time flies **like** an arrow. (preposition)

Accuracy (% tokens assigned the correct POS tags)

- On the Penn Treebank WSJ dataset (news articles)
    - Accuracy: 96.46% (2000) ⍰ 97.85% (2018)
    - An almost "solved" problem ☺
    - But out-of-the-box tools are trained using news dataset …

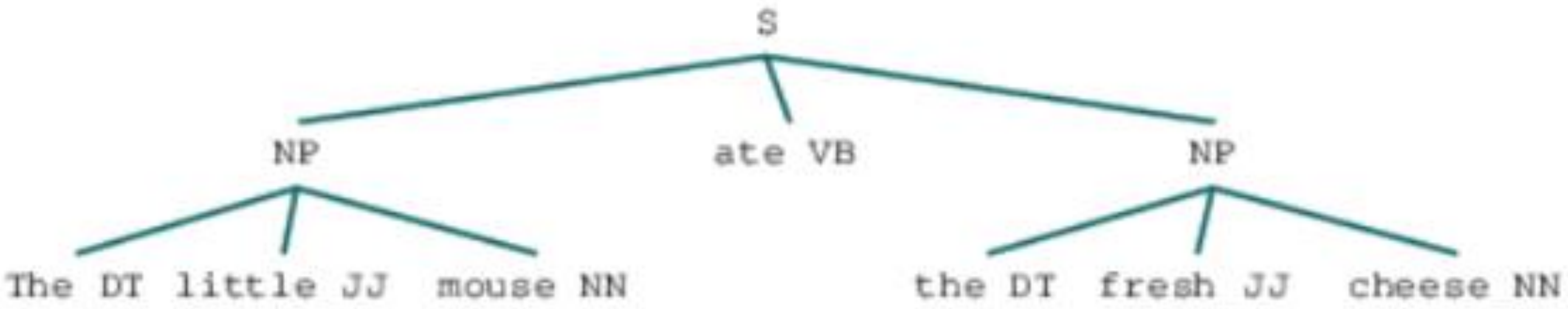
https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

- On a twitter dataset (Gimpel et al., 2011)
    - 24 POS tags + other tags (e.g., hashtag)
    - Accuracy: 89.37% (2011)

Gimpel et al. (2011). Part-of-speech tagging for Twitter: annotation, features, and experiments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.

Chunking

- Purpose: to extract phrases, e.g., noun phrases (NP), verb groups
- Most chunking methods require POS tagging first
- Example: "The little mouse ate the fresh cheese."
  - Two noun phrases: the/DT little/JJ mouse/NN, the/DT fresh/JJ cheese/NN



Rule-based Method

- Defines a POS tag pattern for a type of phrase (e.g., NP)
  - For example: DT? JJ* NN (zero or one determiner, zero or multiple adjective, and a noun)

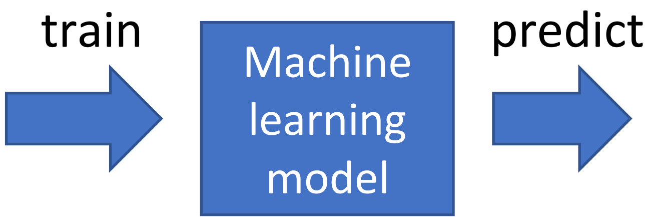
Chunking

Machine Learning-based Method (we’ll cover details in week 8-10)

- Problem formulation: to classify if a particular token is the beginning or inside token of a phrase or not a part of a phrase.

Token	IOB Tags
Mr.	B-NP (beginning of an NP)
Meador	I-NP (inside an NP)
had	B-VP (beginning of a VP)
been	I-VP (inside a VP)
executive	B-NP (beginning of an NP)
vice	I-NP (inside an NP)
president	I-NP (inside an NP)
of	O (not a part of a phrase)
Balcor	B-NP (beginning of an NP)
.	O (not a part of a phrase)

Human Annotation (training data)



Token	IOB Tags
The	?
little	?
mouse	?
ate	?
the	?
fresh	?
cheese	?
.	?

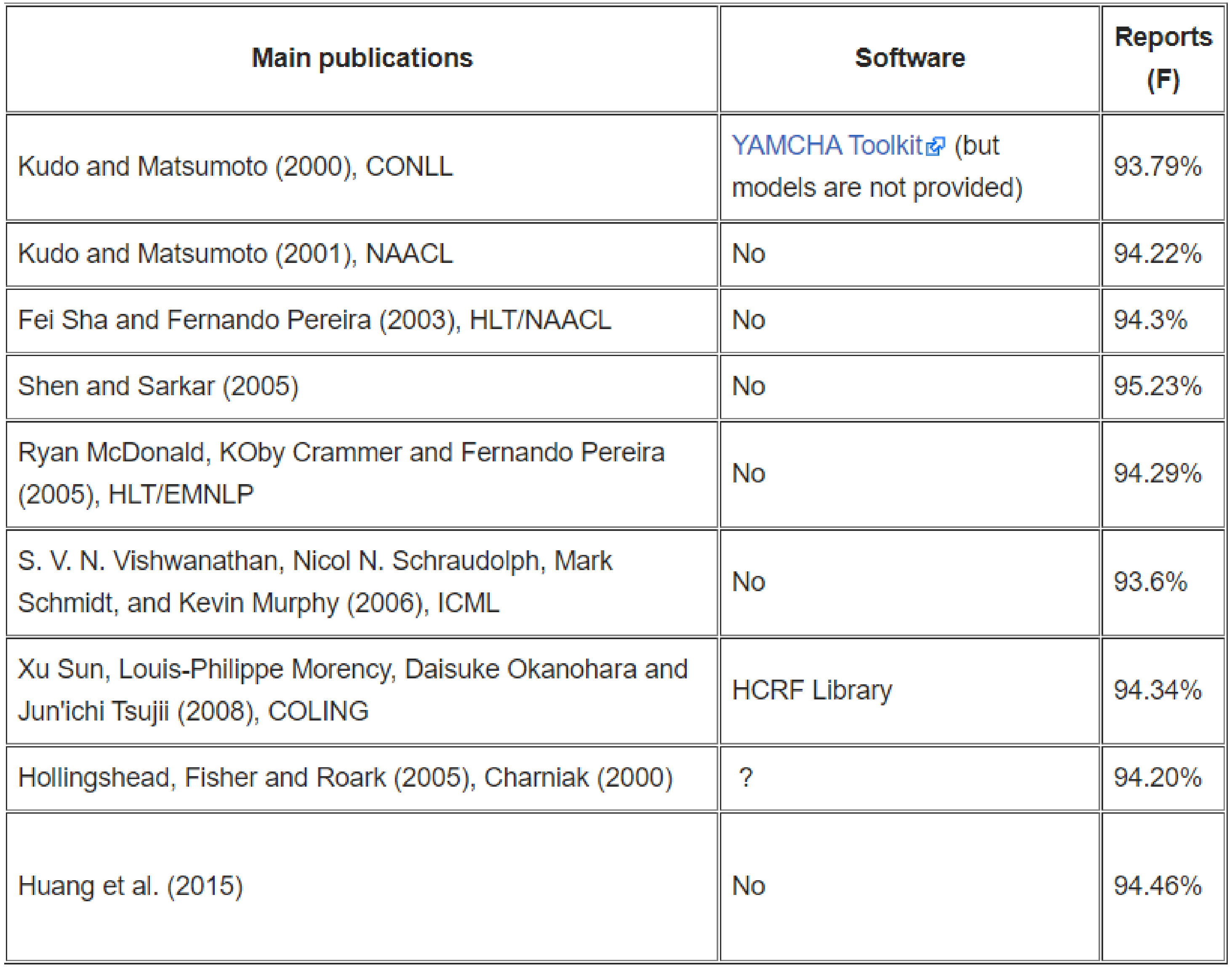
Your problem data

NP Chunking: Accuracies

- Over 90% F-measure (the value ranges between 0-1, where 1 means the best accuracy) on news article dataset.
- About 86% accuracy on a twitter dataset.

Ritter, A., Clark, S., & Etzioni, O. Named entity recognition in tweets: an experimental study. In Proceedings of the 2011 conference on empirical methods in natural language processing (pp. 1524-1534).

NP chunking accuracies on the WSJ dataset



Main publications	Software	Reports (F)
Kudo and Matsumoto (2000), CONLL	YAMCHA Toolkit (but models are not provided)	93.79%
Kudo and Matsumoto (2001), NAACL	No	94.22%
Fei Sha and Fernando Pereira (2003), HLT/NAACL	No	94.3%
Shen and Sarkar (2005)	No	95.23%
Ryan McDonald, KOby Crammer and Fernando Pereira (2005), HLT/EMNLP	No	94.29%
S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark Schmidt, and Kevin Murphy (2006), ICML	No	93.6%
Xu Sun, Louis-Philippe Morency, Daisuke Okanohara and Jun'ichi Tsujii (2008), COLING	HCRF Library	94.34%
Hollingshead, Fisher and Roark (2005), Charniak (2000)	?	94.20%
Huang et al. (2015)	No	94.46%

https://aclweb.org/aclwiki/NP_Chunking_(State_of_the_art)

Named Entity Recognition (NER)

Named Entities

- Recognizes occurrences of Person, Organization, Location, etc. from texts

Michael Dell is the CEO of **Dell Computer Corporation** and lives in **Austin, Texas**.

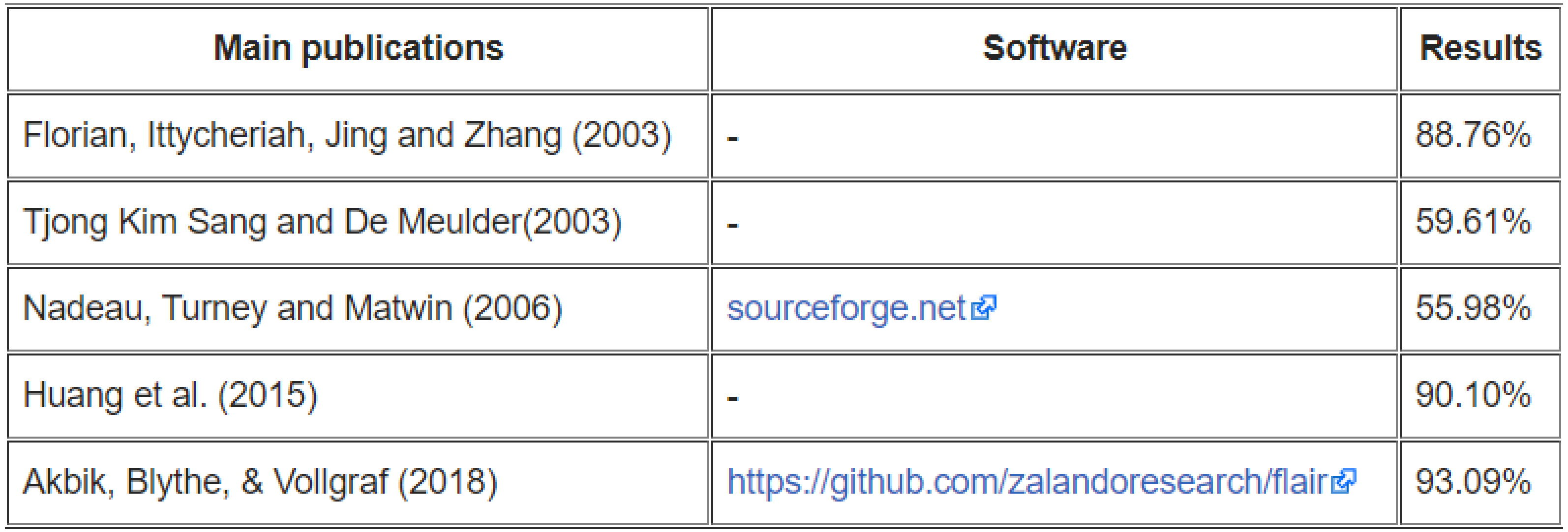
B-Per **I-Per** O O O O **B-Org** **I-Org** **I-Org** O O O **B-Loc** **I-Loc** .

- Machine learning-based solutions for NER are like those for chunking …
  - Use IOB annotations of entity occurrences to train models to predict IOB tags on new text
  - Prediction features can include:
    - Word content
    - POS tags
    - Word shape, e.g., Xxxxx, XXXX (so do not apply case-folding before NER)
    - The above features of context words (left and right n words)

Named Entity Recognition: Accuracies

- Over 90% F-measure (the value ranges between 0-1, where 1 means the best accuracy) on news article dataset (e.g., CONLL-2003).
- About 67% accuracy on a twitter dataset. (Ritter et al., 2011)

NP chunking accuracies on the CONLL-2003 dataset



Main publications	Software	Results
Florian, Ittycheriah, Jing and Zhang (2003)	-	88.76%
Tjong Kim Sang and De Meulder(2003)	-	59.61%
Nadeau, Turney and Matwin (2006)	sourceforge.net	55.98%
Huang et al. (2015)	-	90.10%
Akbik, Blythe, & Vollgraf (2018)	https://github.com/zalandoresearch/flair	93.09%

https://aclweb.org/aclwiki/CONLL-2003_(State_of_the_art)

Summary

Text normalization & parsing

- Tokenization, case-folding, lemmatization & stemming, stop words removal
- Part-of-speech tagging, NP chunking, Named Entity Recognition (NER)

Automatic NLP methods can make mistakes …

- Some problems (e.g., NER) are naturally harder than others (e.g., POS tagging)
- Some text data (e.g., Tweets, text messages) are noisier than others (e.g., news articles)
- Is the training data similar to your problem data?
  - Most out-of-the-box NLP tools are trained using news articles datasets…
  - Needs to be conservative about how well out-of-the-box tools can perform on your data …