Department of Linguistics and Translation
香港城市大學
City University of Hong Kong

Computational Linguistics LT3233

Jixing Li

Lecture 1: Tokenization

Lecture plan

- The course logistics
- Text preprocessing
    - Regular expression
    - Byte-pair encoding
- Tokenization of Chinese
- Short break (15 mins)
- Hands-on exercises

Course logistics

- Instructor: Jixing Li
- TA: Hongbin Qin
- Location: LI-G600
- Time: F 9:00-11:50 am HKT

- Check Canvas for the course syllabus, announcements, assignments, slides, etc. Slides will be uploaded after each lecture.

Course work and grading policy

- 10 x one-week group assignments: 10 x 5%
  - **HW1 is released today! Due next Friday at 9 am HKT.**
  - Submitted to Canvas using your @cityu.edu.hk email

- Final exam: 50%
  - Dec 2 at 9:00 am HKT.

- Late day policy: 3 free late days; afterwards, 1% off the overall course grade.

What we hope to teach

- The major issues and solutions in natural language processing.
- Both traditional rule-based models and modern deep learning techniques.
  - **Topics:** tokenization, part-of-speech tagging, n-gram models, context-free grammars, parsing, linear classification, feed-forward neural networks, computational graph and backpropagation, word embeddings, recurrent neural networks, attention and transformers, transfer learning.

- **Textbooks:**
- Jurafsky, D. and Martin, J.H. (2021) Speech and Language Processing (3rd Edition). https://web.stanford.edu/~jurafsky/slp3/ **[SLP]**
- Bird, S., Klein, E. and Loper, E. (2009) Natural Language Processing with Python. https://www.nltk.org/book/ **[NLTK book]**

Path to excellence

- Read relevant textbook chapters, papers

- Ask “how does it work?”

  Understanding is the goal

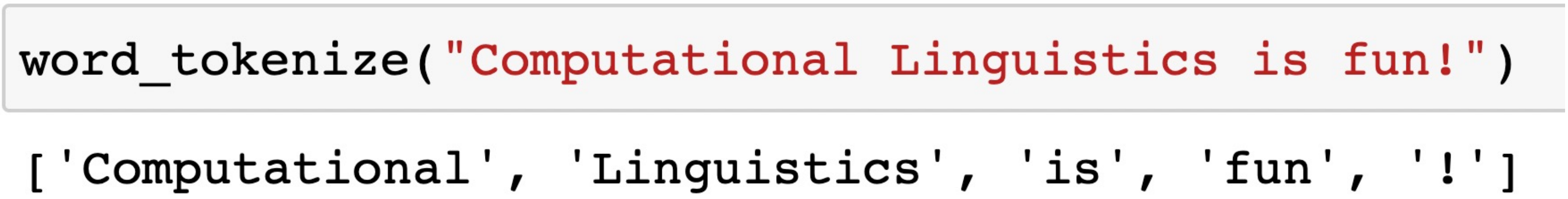
- Code up prototypes

  Hands must get dirty!

- Think about what interest you?

Every NLP task requires ...

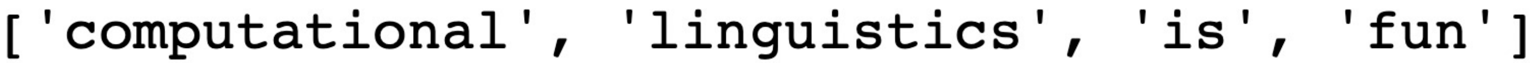
- Tokenizing (segmenting) words



```
word_tokenize("Computational Linguistics is fun!")
```

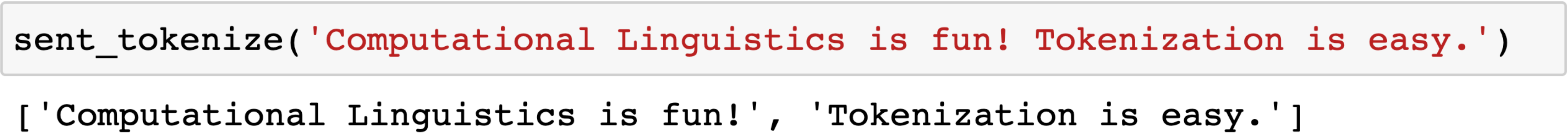
```
['Computational', 'Linguistics', 'is', 'fun', '!']
```

- Normalizing word format



```
['computational', 'linguistics', 'is', 'fun']
```

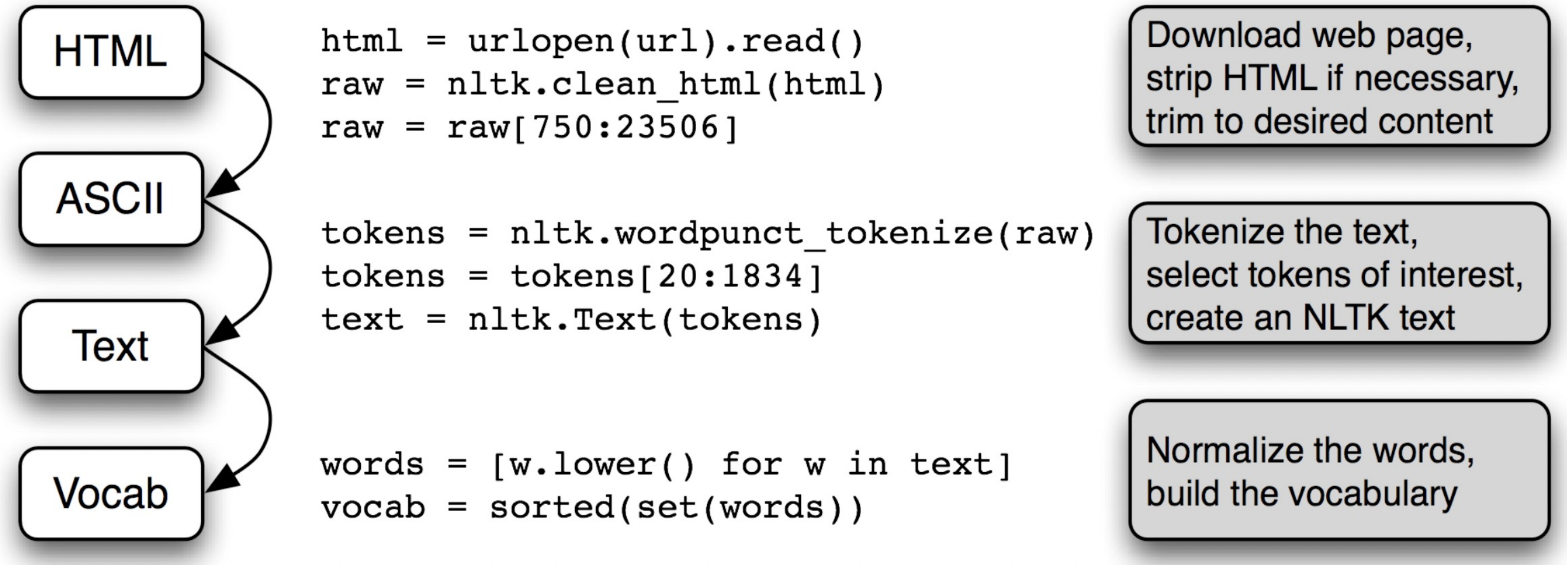
- Segmenting sentences



```
sent_tokenize('Computational Linguistics is fun! Tokenization is easy.')
```

```
['Computational Linguistics is fun!', 'Tokenization is easy.']
```

Use case: Getting web pages



```
html = urlopen(url).read()
raw = nltk.clean_html(html)
raw = raw[750:23506]
```

Download web page, strip HTML if necessary, trim to desired content

```
tokens = nltk.wordpunct_tokenize(raw)
tokens = tokens[20:1834]
text = nltk.Text(tokens)
```

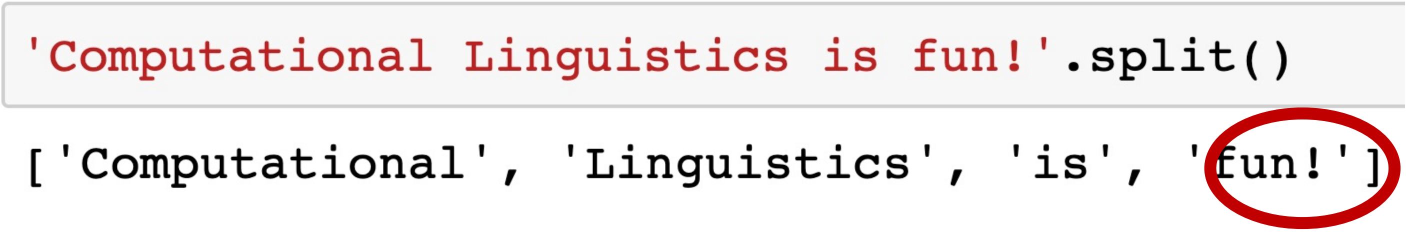
Tokenize the text, select tokens of interest, create an NLTK text

```
words = [w.lower() for w in text]
vocab = sorted(set(words))
```

Normalize the words, build the vocabulary

Tokenization using whitespaces

Using Python’s built-in split() function:



```
'Computational Linguistics is fun!'.split()
```

```
['Computational', 'Linguistics', 'is', 'fun!']
```

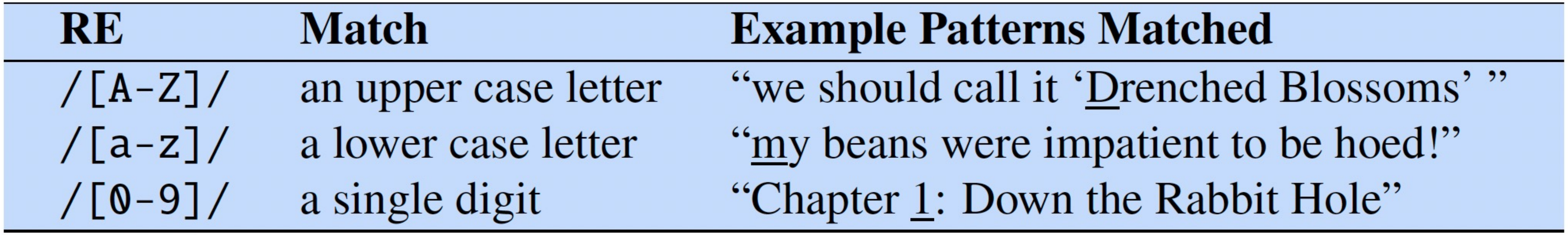
Problem: end-of-sentence words contains punctuation.

Remove punctuation?

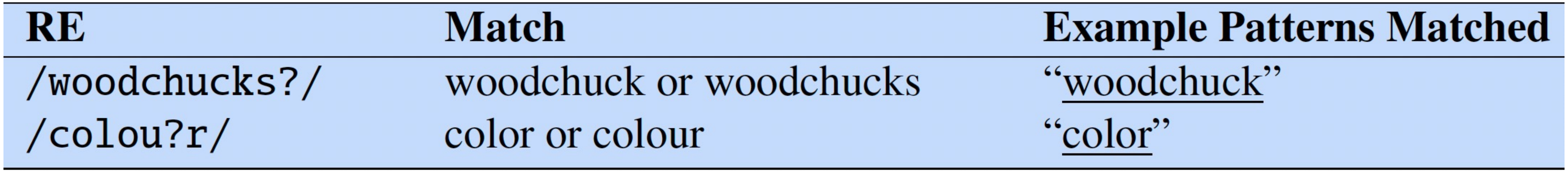
- Ph.D, AT&T; Prices: $45.55; Dates: 01/02/06; URLs: http://www.city.edu.hk
- Hashtags: #nlproc; email addresses: someone@cityu.edu.hk
- Clitics (words that don't stand on their own): ‘are’ in ‘we’re.
- Multiword expressions (MWE): rock ’n’ roll

Tokenization using regular expressions?

Regular Expressions (RE): an algebraic notation for characterizing a set of strings. **c.f. SLP 2.1:**

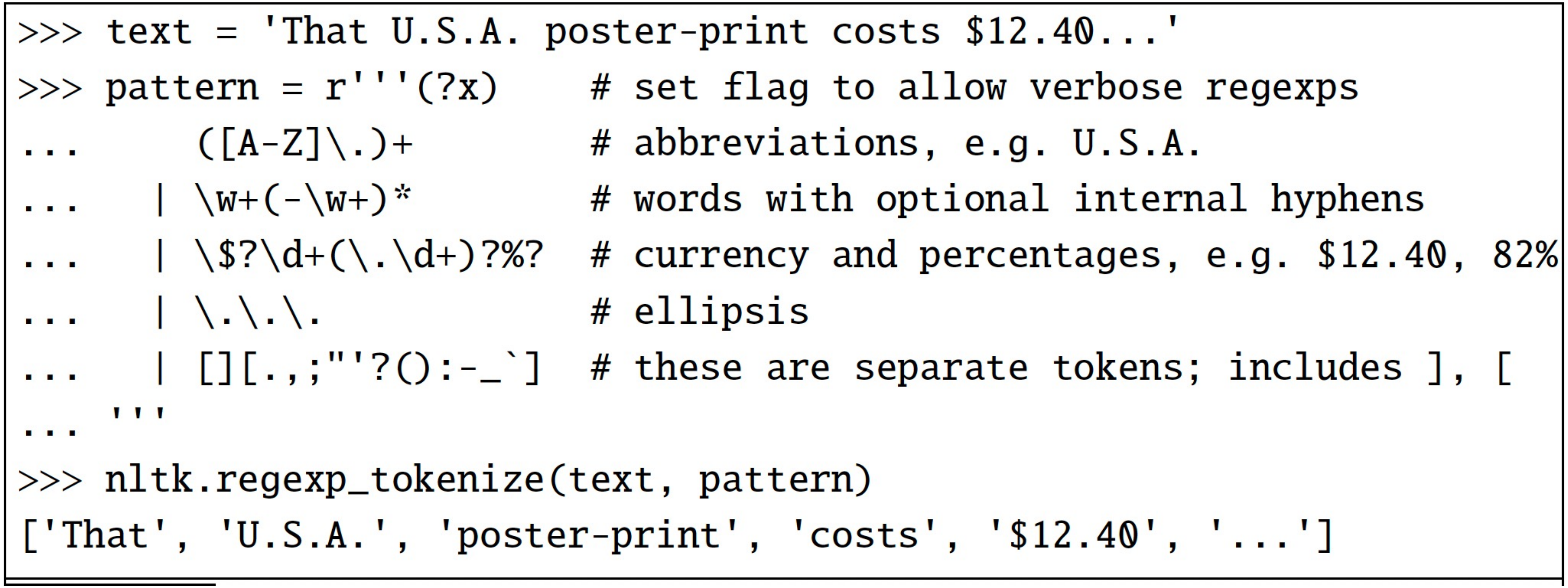


RE	Match	Example Patterns Matched
`/[A-Z]/`	an upper case letter	"we should call it 'Drenched Blossoms' "
`/[a-z]/`	a lower case letter	"my beans were impatient to be hoed!"
`/[0-9]/`	a single digit	"Chapter 1: Down the Rabbit Hole"



RE	Match	Example Patterns Matched
`/woodchucks?/`	woodchuck or woodchucks	"woodchuck"
`/colou?r/`	color or colour	"color"

Tokenization using regular expressions?



```
>>> 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', '...']
```

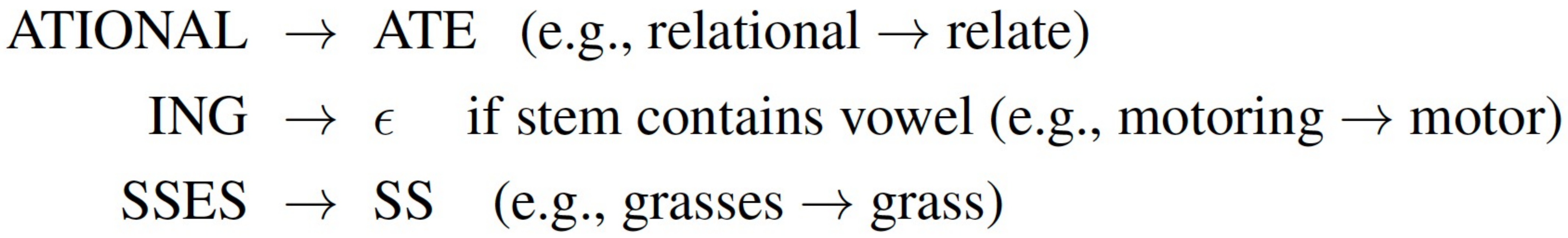
From NLTK book, Ch3

Lemmatization

- **Lemmatization:** replace words with its roots:
  - am, are, is → be, computers → computer
  - ‘He is reading NLP books.’ → ‘He be read NLP book.’
- Sophisticated method: Morphological parser to parse words into **morphemes**.
  - Stems: the central morpheme of the word, supplying the main meaning
  - Affixes: adding ‘additional’ meanings of various kinds.

Stemming

- Simpler but cruder method: chopping off word-final affixes based on rules.
- The Porter stemmer (1980)



ATIONAL → ATE (e.g., relational → relate)

ING → ϵ if stem contains vowel (e.g., motoring → motor)

SSES → SS (e.g., grasses → grass)

Sub-word tokenization

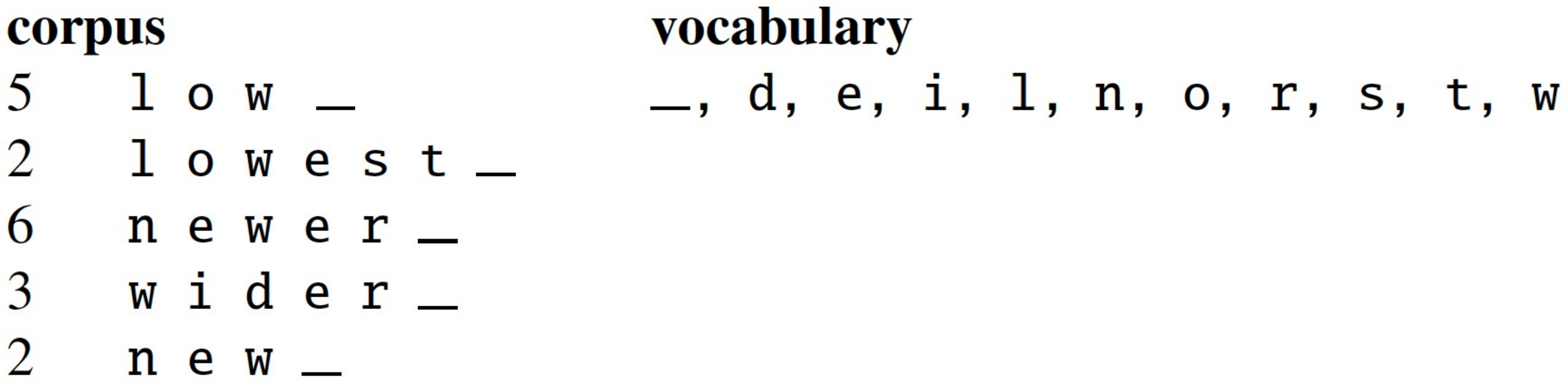
- NLP algorithms often learn from a **training corpus** and tests on a separate **test corpus**. The test corpus may contain words that are not in the training corpus.
  - e.g. low, new, newer, but not lower

- How to deal with these **out-of-vocabulary (OOV)** words?

- **Byte-pair encoding (BPE)** (Sennrich et al., 2016)

Byte-pair encoding (BPE)

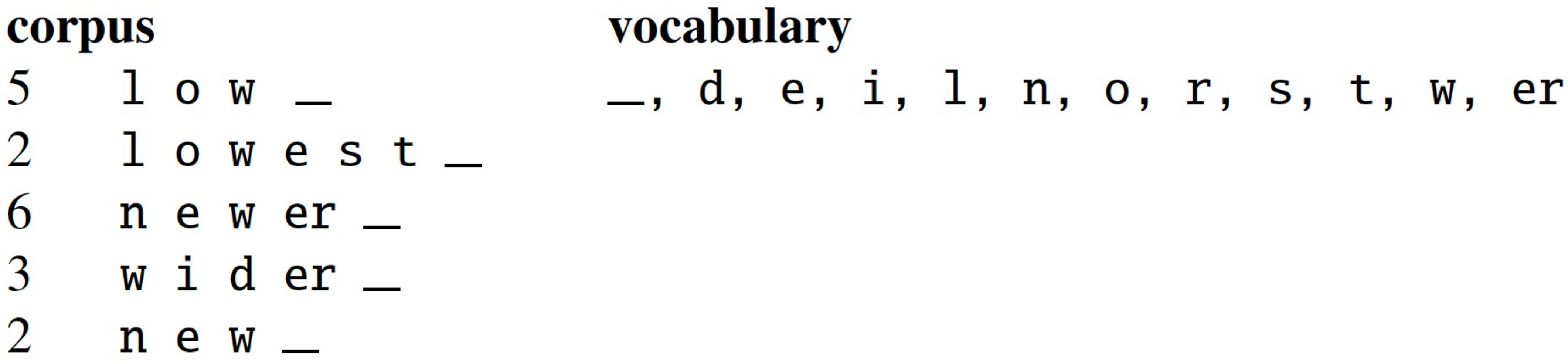
- Begins with a vocabulary that is just the set of all individual characters in the training corpus, with counts for each word.



corpus		vocabulary
5	l o w _	_, d, e, i, l, n, o, r, s, t, w
2	l o w e s t _	
6	n e w e r _	
3	w i d e r _	
2	n e w _	

Byte-pair encoding (BPE)

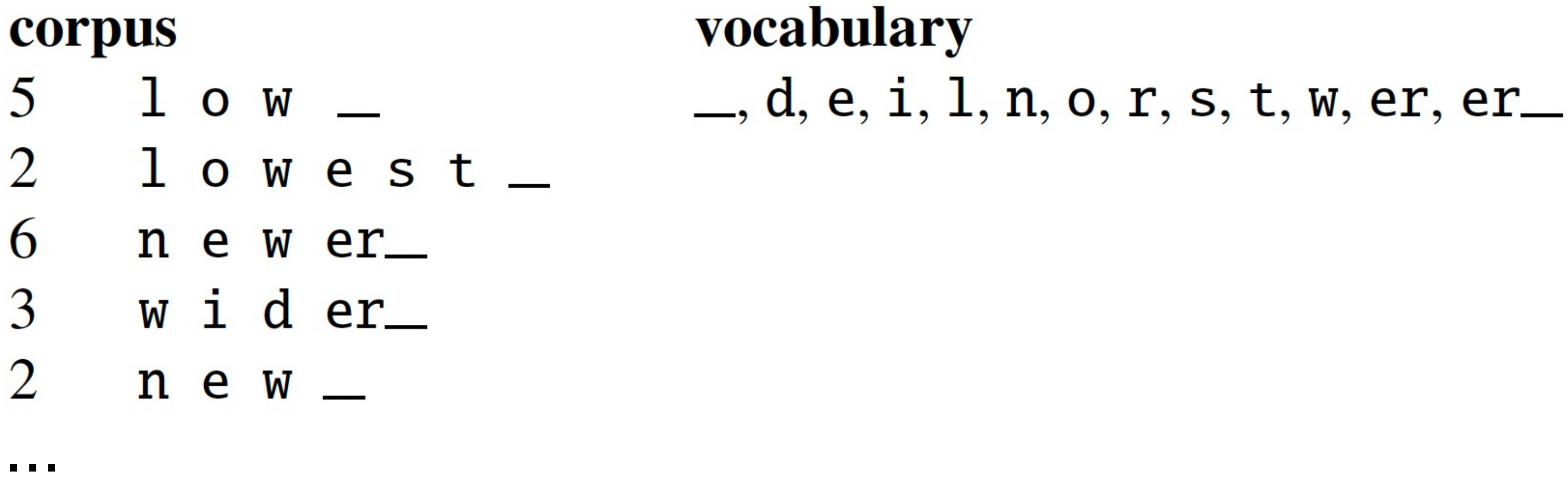
- Examines the training corpus, chooses the two symbols that are most frequently adjacent (say ‘A’, ‘B’), adds a new merged symbol ‘AB’ to the vocabulary, and replaces every adjacent ’A’ ’B’ in the corpus with the new ‘AB’.



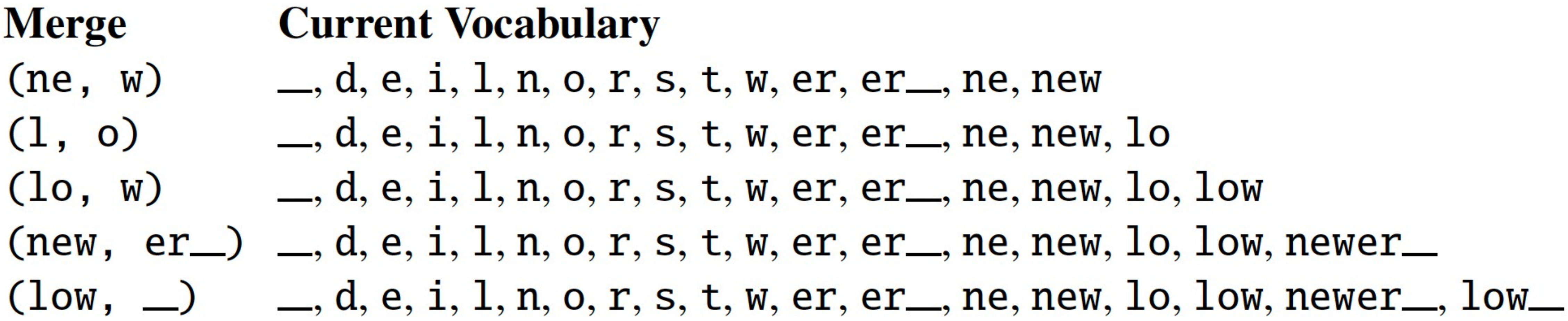
corpus		vocabulary
5	l o w _	_, d, e, i, l, n, o, r, s, t, w, er
2	l o w e s t _	
6	n e w er _	
3	w i d er _	
2	n e w _	

Byte-pair encoding (BPE)

- Repeat this process, until k merges have been done



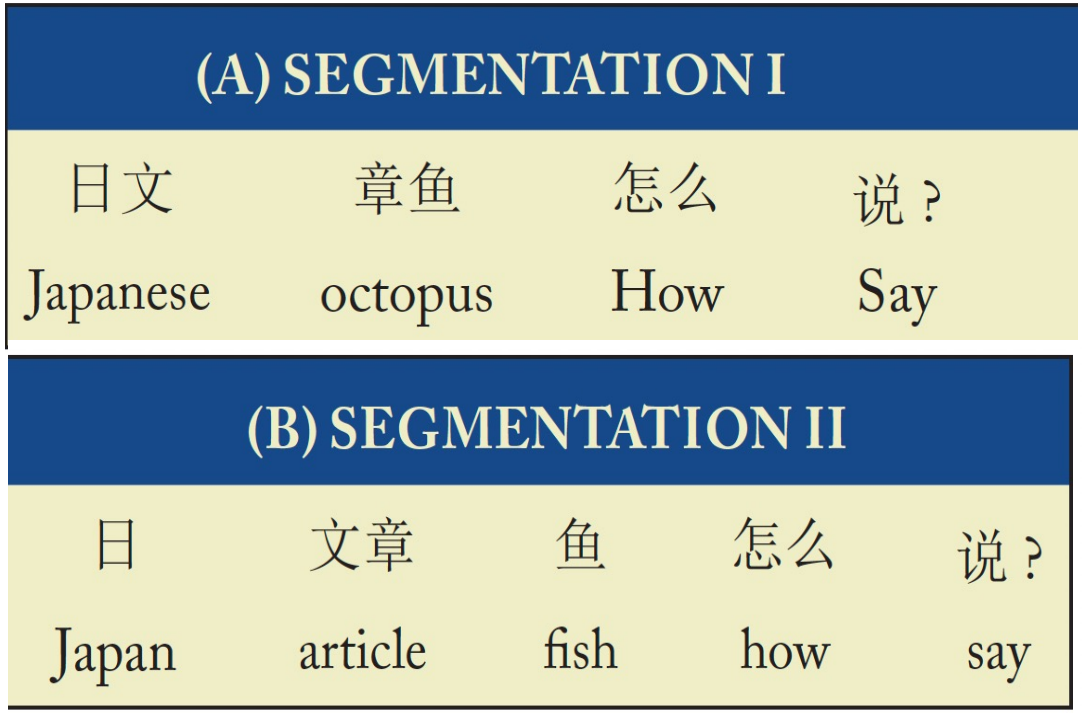
corpus		vocabulary
5	l o w _	_, d, e, i, l, n, o, r, s, t, w, er, er_
2	l o w e s t _	
6	n e w er_	
3	w i d er_	
2	n e w _	
…		



Merge	Current Vocabulary
(ne, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new
(l, o)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo
(lo, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low
(new, er_)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_
(low, _)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_, low_

Tokenization of Chinese

- What about other languages like Chinese, Japanese, Thai, etc do not use spaces to mark word boundaries?

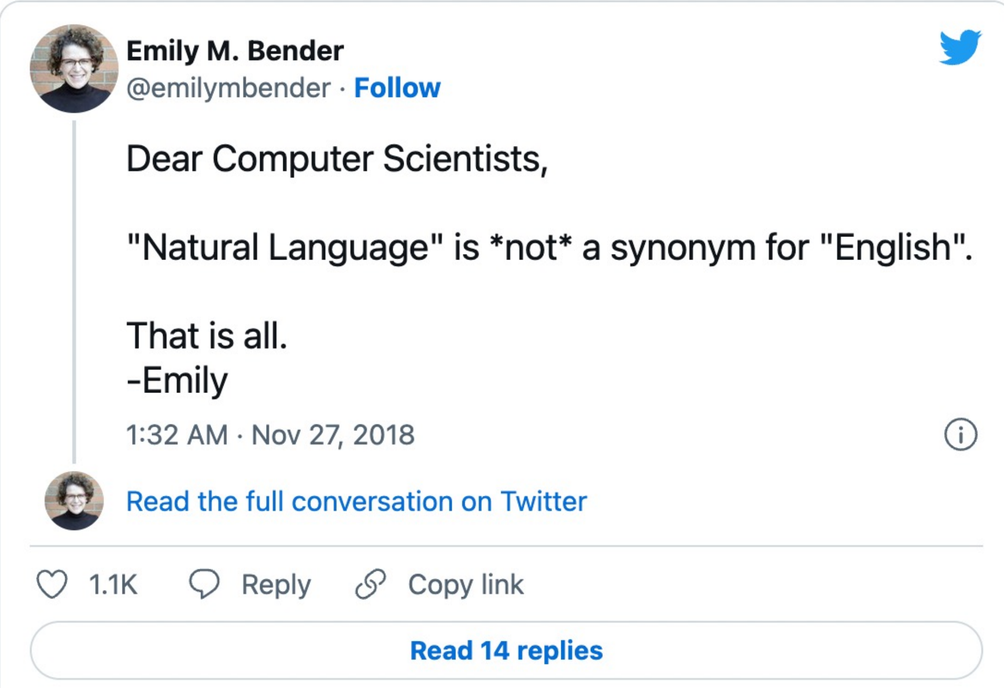


(A) SEGMENTATION I

日文	章鱼	怎么	说？
Japanese	octopus	How	Say

(B) SEGMENTATION II

日	文章	鱼	怎么	说？
Japan	article	fish	how	say



Emily M. Bender
@emilymbender · Follow

Dear Computer Scientists,

"Natural Language" is *not* a synonym for "English".

That is all.
-Emily

1:32 AM · Nov 27, 2018

The algorithm behind jieba

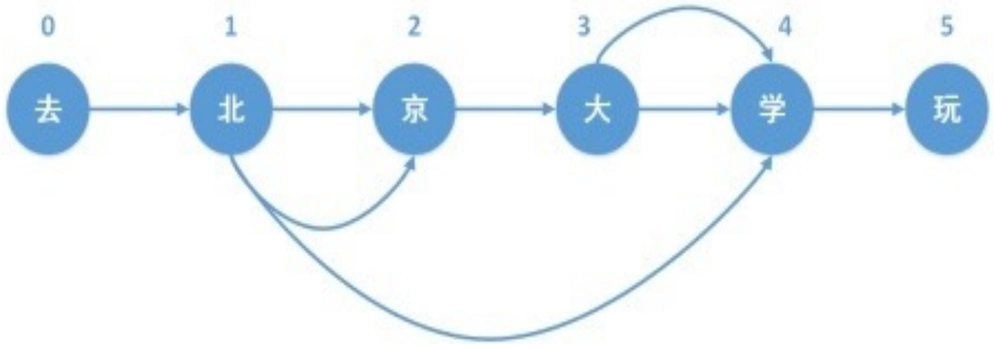
- E.g.,”去北京大学玩”,

北京大学	2053	nt
大学	20025	n
去	123402	v
玩	4207	v
北京	34488	ns
北	17860	ns
京	6583	ns
大	144099	a
学	17482	n

→

北京大学	2053
北京大	0
大学	20025
去	123402
玩	4207
北京	34488
北	17860
京	6583
大	144099
学	17482

→



→ HMM

To do

- Read SLP Ch2, NLTK book Ch1-2
- Do HW1