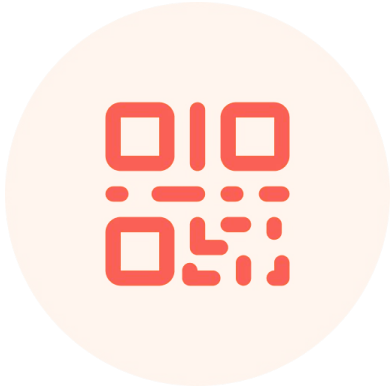
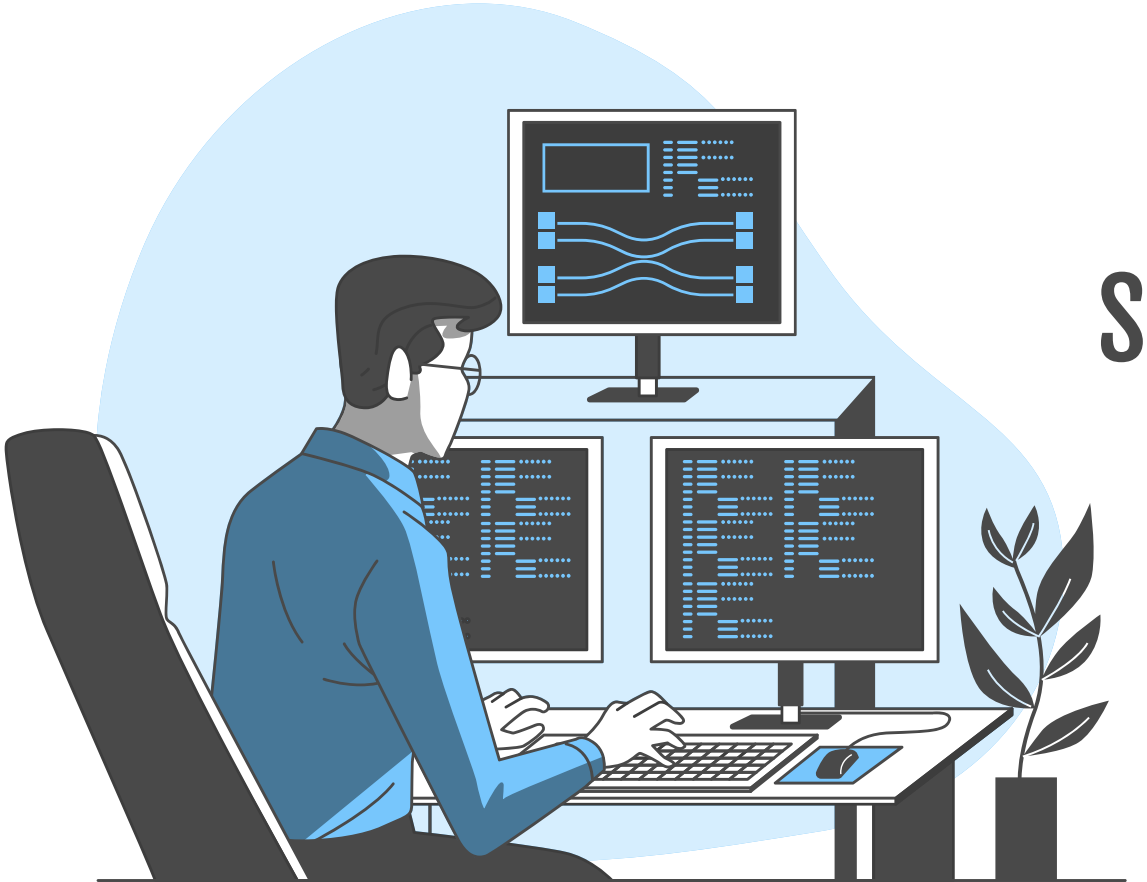
slido



Join at slido.com
#1540124

ⓘ Start presenting to display the joining instructions on this slide.



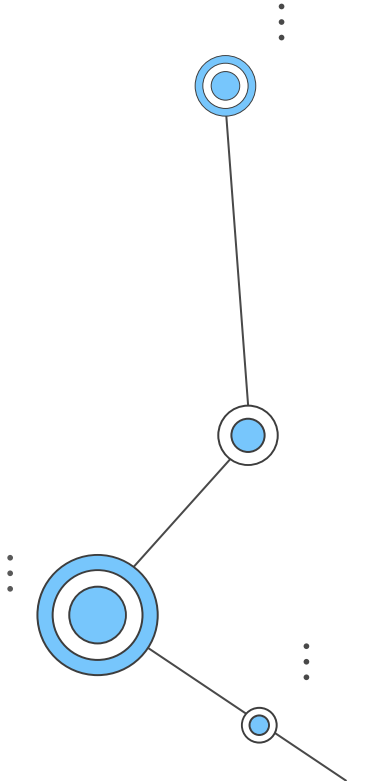
Word Segmentation

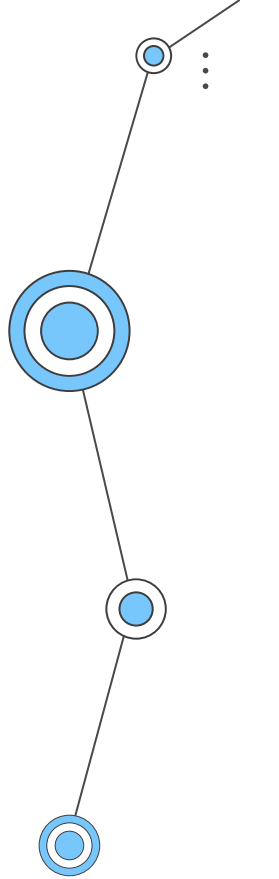
2023/03/09

Word Segmentation

Jieba & ckip

Conclusion





01 Quick Introduction

Why do we need word segmentation and POS tagging?

4.1 Word Segmentation Standard

- **Definition:** A segmentation unit is the smallest string of character(s) that has both an independent meaning and a fixed grammatical category.

Bound morpheme?
Reduplication?

- **Basic principles:**
  - A string whose meaning **cannot** be derived by the sum of its components should be treated as a segmentation unit. [Combination principle]
  - The string whose grammatical category **cannot** be derived by the sum of the grammatical categories of its components should be treated as a segmentation unit. [Combination principle]

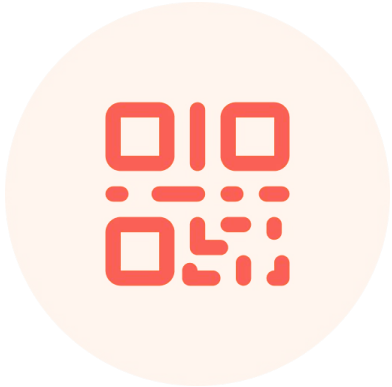
e.g. 打臉

5.1 The differences between Mainland and Taiwan Word Segmentation Standards

Segmentation units did not equal to words and the target is not the linguistic word, but a processing unit for information processing of Chinese texts.

- **The Word Segmentation Standard of Contemporary Chinese Language for Information Processing**
  - A **word** is the smallest element that may be uttered in isolation.
  - A **segmentation unit** is the smallest element that may be adopted in Chinese information processing and still have a semantic or syntax function. It includes word and phrases in the standard.

slido



Join at slido.com
#1540124

ⓘ Start presenting to display the joining instructions on this slide.

How to subcategorize these words?

Behavioral Characteristics!!!

Chinese NLP pipeline in spaCy

01 Tokenization

02 POS tagging

03 Dependency parsing

04 NER

```
下課 NOUN noun NT temporal noun
我 PRON pronoun PN pronoun
要 VERB verb VV other verb
知道 VERB verb VC 是 (copula)
寶雅 PROPN proper noun NR proper noun
屈臣氏 PROPN proper noun NR proper noun
他們 PRON pronoun PN pronoun
```

token.pos_:
coarse-grained pos

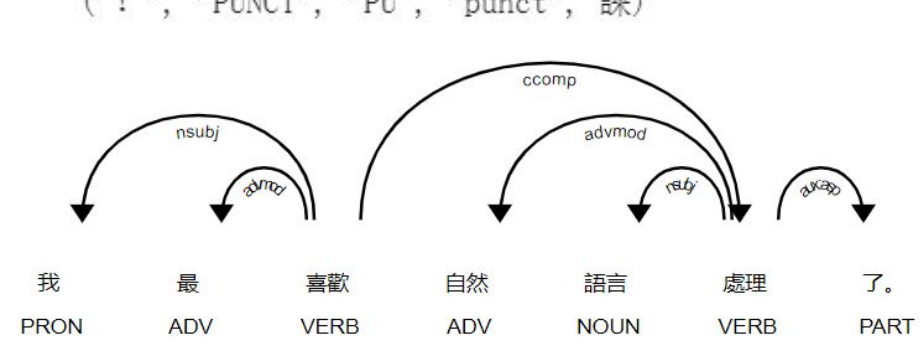
token.tag_:
fine-grained pos

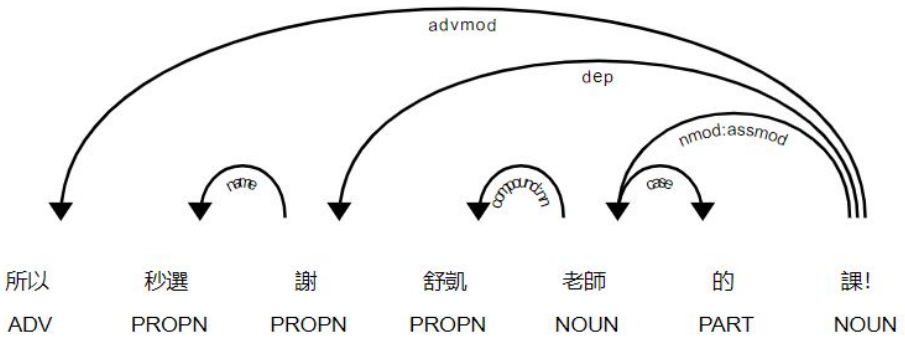
Chinese NLP pipeline in spaCy

03 Dependency parsing

```
('我', 'PRON', 'PN', 'nsubj', 喜歡)
('最', 'ADV', 'AD', 'advmod', 喜歡)
('喜歡', 'VERB', 'VV', 'ROOT', 喜歡)
('自然', 'ADV', 'AD', 'advmod', 處理)
('語言', 'NOUN', 'NN', 'nsubj', 處理)
('處理', 'VERB', 'VV', 'ccomp', 喜歡)
('了', 'PART', 'SP', 'aux:asp', 處理)
('。', 'PUNCT', 'PU', 'punct', 喜歡)
('所以', 'ADV', 'AD', 'advmod', 課)
('秒選', 'PROPN', 'NR', 'name', 謝)
('謝', 'PROPN', 'NR', 'dep', 課)
('舒凱', 'PROPN', 'NR', 'compound:nn', 老師)
('老師', 'NOUN', 'NN', 'nmod:assmod', 課)
('的', 'PART', 'DEG', 'case', 老師)
('課', 'NOUN', 'NN', 'ROOT', 課)
('！', 'PUNCT', 'PU', 'punct', 課)
```

token 對這個 head 有這樣的依存關係





slido

To what extent, can we create new verbs?

ⓘ Start presenting to display the poll results on this slide.

slido

To what extent, can we create new prepositions?

ⓘ Start presenting to display the poll results on this slide.

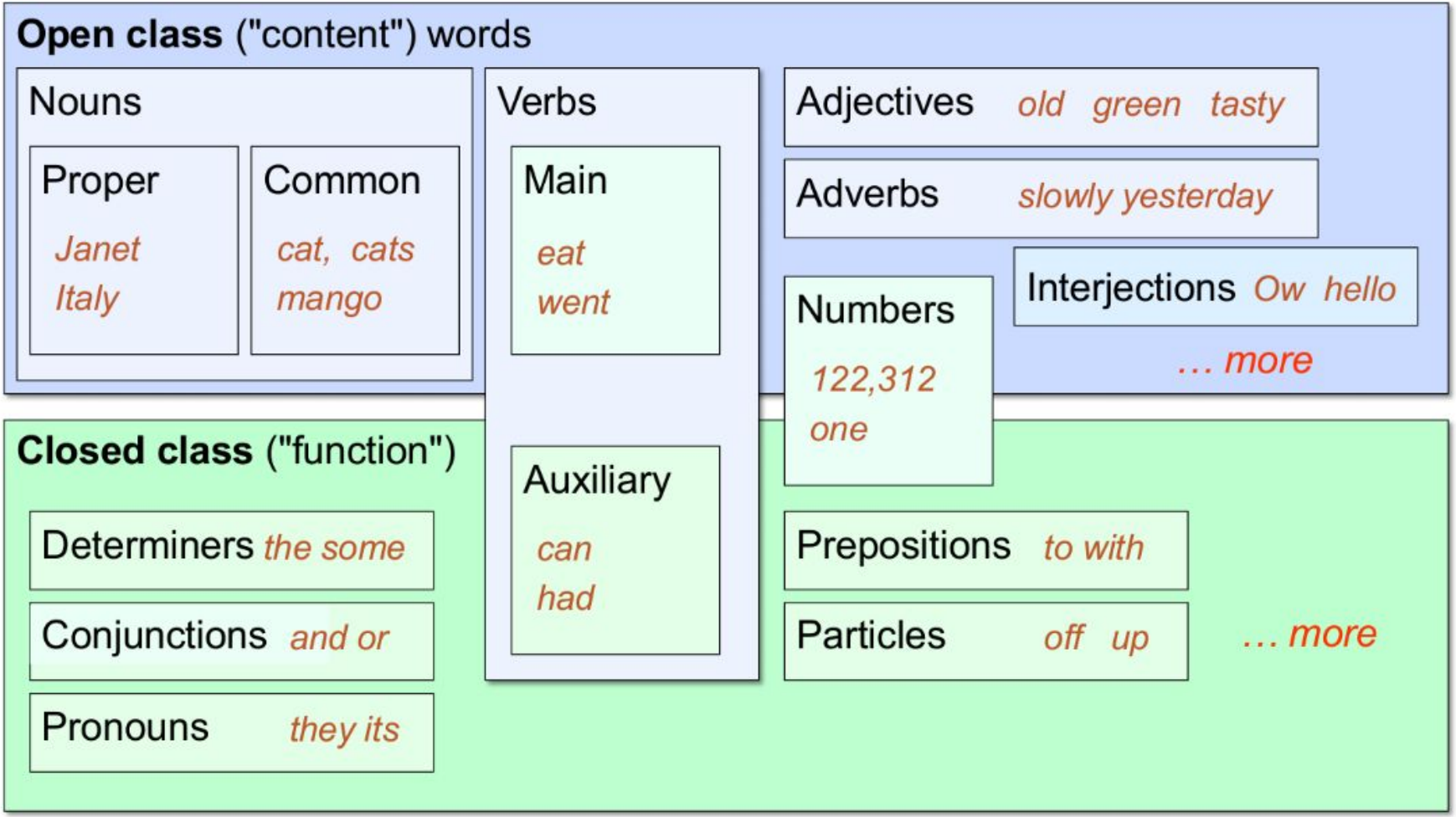
slido

To what extent, can we create new nouns?

ⓘ Start presenting to display the poll results on this slide.

Open class vs. Closed class

- Open class words
  - Usually **content** words: nouns, verbs, adjectives, adverbs …
  - New words like *iPhone* or *to fax*
- Closed class words
  - Usually **function** words with grammatical function: determiners, pronouns, prepositions …
  - Relatively fixed membership


Open class ("content") words
Nouns
Proper
Janet
Italy
Common
cat, cats
mango
Verbs
Main
eat
went
Auxiliary
can
had
Adjectives old green tasty
Adverbs slowly yesterday
Numbers
122,312
one
Interjections Ow hello
… more
Closed class ("function")
Determiners the some
Conjunctions and or
Pronouns they its
Prepositions to with
Particles off up
… more

"Universal Dependencies" Tagset

Nivre et al. 2016

	Tag	Description	Example
Open Class	**ADJ**	Adjective: noun modifiers describing properties	*red, young, awesome*
	ADV	Adverb: verb modifiers of time, place, manner	*very, slowly, home, yesterday*
	NOUN	words for persons, places, things, etc.	*algorithm, cat, mango, beauty*
	VERB	words for actions and processes	*draw, provide, go*
	PROPN	Proper noun: name of a person, organization, place, etc..	*Regina, IBM, Colorado*
	INTJ	Interjection: exclamation, greeting, yes/no response, etc.	*oh, um, yes, hello*
Closed Class Words	**ADP**	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	*in, on, by under*
	AUX	Auxiliary: helping verb marking tense, aspect, mood, etc.,	*can, may, should, are*
	CCONJ	Coordinating Conjunction: joins two phrases/clauses	*and, or, but*
	DET	Determiner: marks noun phrase properties	*a, an, the, this*
	NUM	Numeral	*one, two, first, second*
	PART	Particle: a preposition-like form used together with a verb	*up, down, on, off, in, out, at, by*
	PRON	Pronoun: a shorthand for referring to an entity or event	*she, who, I, others*
	SCONJ	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	*that, which*
Other	**PUNCT**	Punctuation	; , ()
	SYM	Symbols like $ or emoji	$, %
	X	Other	asdf, qwfg

- Can be useful for other NLP tasks
  - Improve syntactic/dependency parsing
  - Help ML in reordering of constituents
  - Sentiment analysis may want to distinguish adjectives or other POS
  - Text-to-speech needs to contrast between homonyms/allophones etc. (e.g. *兒的生活好痛苦一點也沒有糧食多病少掙了很多錢*)

- And also useful for linguistic or language-analytic computational tasks
  - Control for POS when studying linguistic change (creation of new words, or meaning shift)
  - Control for POS in measuring meaning similarity or difference

02

Jieba & ckip

Python modules for word segmentation and POS tagging

- If you haven’t installed the modules
  - To install jieba

```
$ pip install jieba
```

  - To install ckip-transformers

```
pip install -U ckip-transformers
```

- To load module

```
# dependencies
import jieba
import jieba.posseg as pseg
import logging
jieba.setLogLevel(logging.INFO)

import ckip_transformers
from ckip_transformers.nlp import CkipWordSegmenter, CkipPosTagger
```

01

Word Segmentation

02

POS tagging

Jieba

```
text_jb = jieba.cut(text)

print(text)
print('\t'.join(text_jb))
```

```
我最喜歡自然語言處理了
我	最	喜歡	自然	語言	處理	了
```

01

Word Segmentation

Jieba

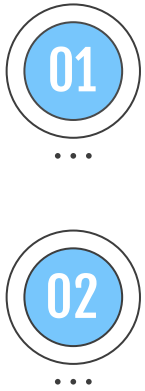
02

POS tagging

```
words = pseg.cut(text)

print(text)
print('\t'.join(text_jb))
print('\t'.join([f'{word}/{tag}' for word, tag in words]))
```

```
我最喜歡自然語言處理了
我      最      喜歡      自然      語言      處理      了
我/r    最/d    喜歡/v    自然/d    語言/n    處理/v    了/ul
```



Word Segmentation

POS tagging

ckip

```
# with GPU
ws_driver = CkipWordSegmenter(device=0)
pos_driver = CkipPosTagger(device=0)

# no GPU
# ws_driver = CkipWordSegmenter(device=-1)
# pos_driver = CkipPosTagger(device=-1)
```

```
texts = text.strip().splitlines()
ws_texts = ws_driver(texts)
pos_texts = pos_driver(ws_texts)
```

```
for sentence, sentence_ws in zip(texts, ws_texts):
  print(sentence)
  print('\t'.join(sentence_ws))
```

```
我最喜歡自然語言處理了
我	最	喜歡	自然	語言	處理	了
```

01

Word Segmentation

ckip

02

POS tagging

```
for sentence, sentence_ws, sentence_pos in zip(texts, ws_texts, pos_texts):
  print(sentence)
  print('\t'.join(sentence_ws))
  print('\t'.join([f'{word}/{tag}' for word, tag in zip(sentence_ws, sentence_pos)]))
```

```
我最喜歡自然語言處理了
我	最	喜歡	自然	語言	處理	了
我/Nh	最/Dfa	喜歡/VK	自然/Na	語言/Na	處理/VC	了/Di
```

Reference

[Speech and Language Processing]