NATURAL LANGUAGE PROCESSING (NLP) FUNDAMENTALS

NLP TOOLS

DR LEK HSIANG HUI

Recap: Basic Building Blocks of Text Pre-processing

- Some basic building blocks useful text pre-processing includes:
  - **Sentence Segmentation**
  - **Tokenization**
  - **Text Normalization**
- We also need to be able to perform string manipulation effectively
  - Use **Regular Expression!**

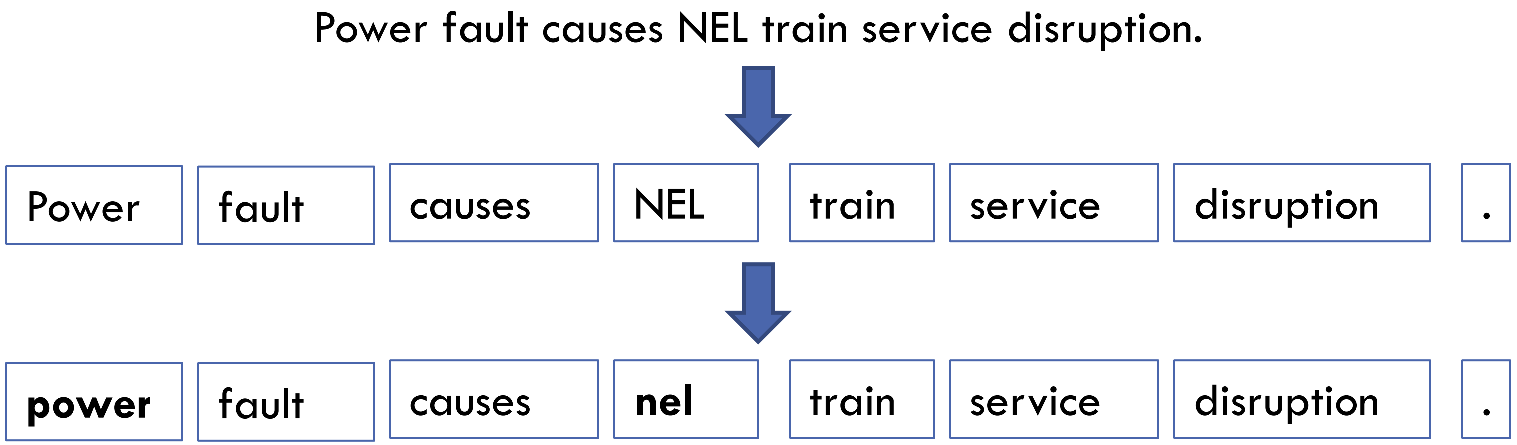
Text Normalization

Text Normalization

- Text Normalization is trying to convert text into a general/standard form
  - E.g. "car" instead of "cars"
  - Why?
    - Ensure that the text is as general as possible
    - So that when doing machine learning, we can derive as many patterns as possible
    - Basically, we want to have as few unique tokens as possible

Convert to Lower Case

- One way to reduce the number of unique tokens is by converting the tokens to lower case



Power fault causes NEL train service disruption.

Power | fault | causes | NEL | train | service | disruption | .

power | fault | causes | **nel** | train | service | disruption | .

- The word ***power*** is then treated as the same token as the ***power*** in "*There is a **power** failure at my area.*"

Convert to Lower Case

- But be careful that this might not always be a good idea:
  - PIE (Pan Island Expressway) vs pie (food)
  - Stephen King (Name) vs King (title)

Remove Punctuation

- Another common text normalization strategy is to remove all the punctuation:
  - Ph.D. → Phd, Mr. → Mr, etc, What?!!! → What
- This essentially reduce the number of features/tokens of the text

Token Replacement

- One effective way to generalize the text is to convert tokens into a more general form
  - E.g.

    14 Sep 2022, 08/08/2000 → DATE
    51, twenty → NUM
    John Smith → PERSON
    :) :] :)) → HAPPY_FACE

Spelling Correction and Standardization

- To reduce the number of unique tokens, we can also attempt to fix typos and standardization (e.g. convert UK wording to US, etc)
  - E.g.
    finaly → finally
    colour → color
    U.S.A → USA
    United Kingdom → UK
    café → cafe
- Spelling Correction
  - In practice, take up a lot of time to do the correction but does not always equate to better overall accuracy

Stemming

Stemming

- In additional to the obvious transformation (change to lower case, etc), we can also transform words to their **stem** (or root form)
  - E.g.
    books ⟶ book
    beautiful ⟶ beauty
    eats ⟶ eat
- This process is called **stemming**

Words

- Words are made up of 2 main parts:
  - **Stem**: root form of the word (core meaning unit)
  - **Affixes**: "add ons" to the stem to form new words with different meaning
    - Prefix (**anti**social)
    - Suffix (sleep**ing**)
    - Circumfix (**en**light**en**)

Porter Stemmer

□ Porter stemmer is a popular rule-based stemming algorithm:

1. Remove plurals, -ed, -ing
2. Turn terminal y to i when there is another vowel in the stem (furry → furri, fry → fry)
3. Maps double suffixes to single ones (playfulness → playful)
4. Deals with suffixes, -full, -ness etc
5. Takes off –ant, -ence, etc
6. Removes the final -e

Lemmatization

Lemmatization

- Stemming might not result in proper words
  - E.g.
    Europe imports more diesel from MidEast, Asia to replace Russia
    → (apply porter stemming)
    ***Europ** import more diesel from MidEast, Asia to **replac** Russia*
- An alternative to stemming is **lemmatization**
  - Lemmatization also has the same general objective of converting a word to its base form
  - Except that this is done using the help of a **dictionary** (e.g. WordNet)

Lemmatization

- Since this process is achieved using a dictionary (e.g. WordNet), **lemmatization** results in proper words in its base form *(also known as **lemma**)*
  - E.g.

    Europe imports more diesel from MidEast, Asia to replace Russia

    → (lemmatization using WordNet)

    *Europe **import** more diesel from MidEast, Asia to replace Russia*

    He went home happily

    → (lemmatization using WordNet)

    *He **go** home happily*

Lemmatization

- Just do lemmatization instead of stemming?
  - Lemmatization is considerably slower though since we often look up a dictionary
  - In addition, we often also need to provide the **Part of Speech** tags for it to work properly

Lemma, Lexeme, Sense

- **Lemma** = base form or head word that represents the **lexeme**
- **Lexeme** = set of inflected word forms of the lemma
  - E.g.
    Lemma = eat
    Lexeme = {eating, ate, eats}
- Words have specific meaning based on how they are used (aka different **sense** of the word)
  - The **Word Sense Disambiguation (WSD)** task is to determine the correct sense of a word in context

Relationship of Words

- In linguistics, there are different types of relationship of words
  - **Homonyms**
  - **Polysemes**
  - **Synonyms**
  - **Antonyms**
  - **Hyponyms**
  - **Hypernyms**

Homonyms

- **Homonyms**: words that share the same spelling but have different *unrelated* meaning
  - Bank
    - Sloping land beside a body of water (i.e. river **bank**)
    - Financial institution (i.e. investment **bank**)

Polysemes

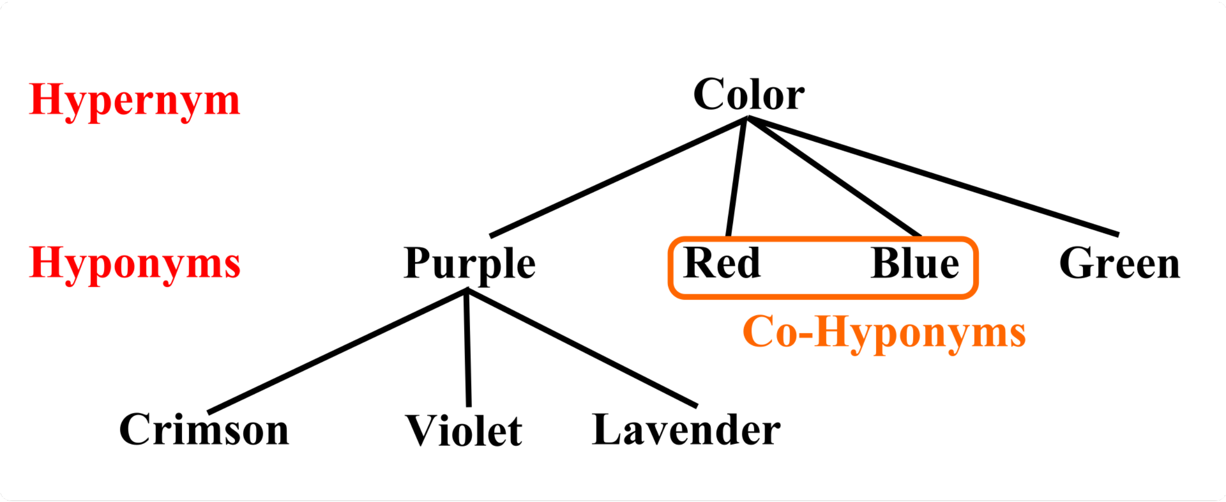
- **Polysemes**: words that share the same spelling but have different *related* meaning
  - Serve
    - Do duty or hold offices (e.g. he **served** as the head of department)
    - Spend time in prison (e.g. he **served** his time for embezzlement
  - Different sense of the words within a lexeme might exhibit homonymy and polysemy
    - WordNet does not distinguish homonym from polysemy

Synonyms & Antonyms

- **Synonyms**: different words with the same meaning
  - Happy
    - Synonyms = {well-chosen, felicitous, glad}
- **Antonyms**: words with opposite meaning
  - Happy
    - Antonyms = {unhappy}

Hyponyms & Hypernyms

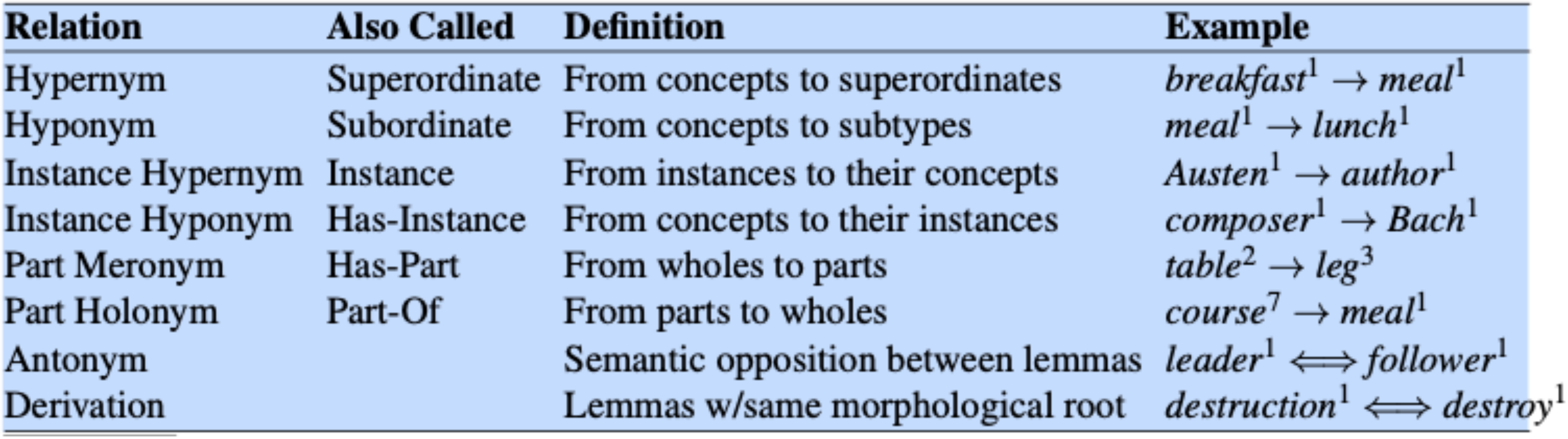
- **Hyponyms**: semantic relationship that is a *subtype*
- **Hypernyms**: semantic relationship that is a *supertype*



WordNet

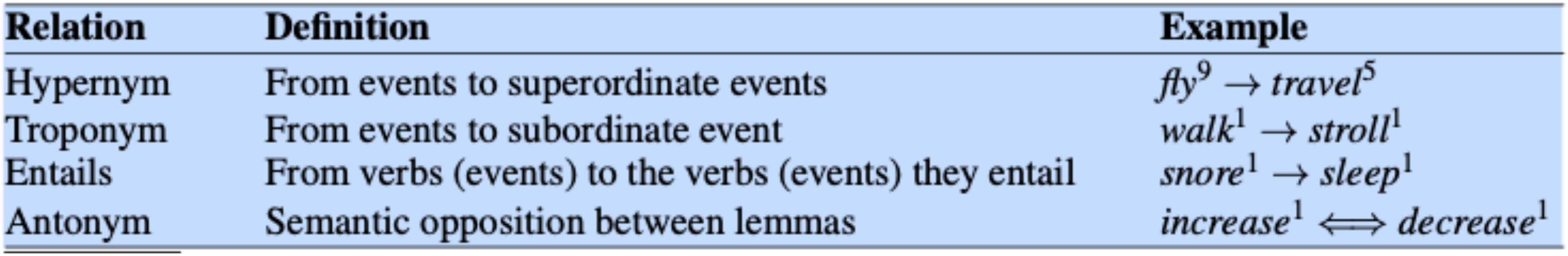
- The most widely used English dictionary for building NLP applications
  - WordNet 3.0:
    - 117,798 nouns
    - 11,529 verbs
    - 21,479 adjectives
    - 4,481 adverbs

Noun Relations in WordNet



Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	*breakfast*^{1} → *meal*^{1}
Hyponym	Subordinate	From concepts to subtypes	*meal*^{1} → *lunch*^{1}
Instance Hypernym	Instance	From instances to their concepts	*Austen*^{1} → *author*^{1}
Instance Hyponym	Has-Instance	From concepts to their instances	*composer*^{1} → *Bach*^{1}
Part Meronym	Has-Part	From wholes to parts	*table*^{2} → *leg*^{3}
Part Holonym	Part-Of	From parts to wholes	*course*^{7} → *meal*^{1}
Antonym		Semantic opposition between lemmas	*leader*^{1} $\Longleftrightarrow$ *follower*^{1}
Derivation		Lemmas w/same morphological root	*destruction*^{1} $\Longleftrightarrow$ *destroy*^{1}

Verb Relations in WordNet



Relation	Definition	Example
Hypernym	From events to superordinate events	$\textit{fly}^9 \rightarrow \textit{travel}^5$
Troponym	From events to subordinate event	$\textit{walk}^1 \rightarrow \textit{stroll}^1$
Entails	From verbs (events) to the verbs (events) they entail	$\textit{snore}^1 \rightarrow \textit{sleep}^1$
Antonym	Semantic opposition between lemmas	$\textit{increase}^1 \Longleftrightarrow \textit{decrease}^1$

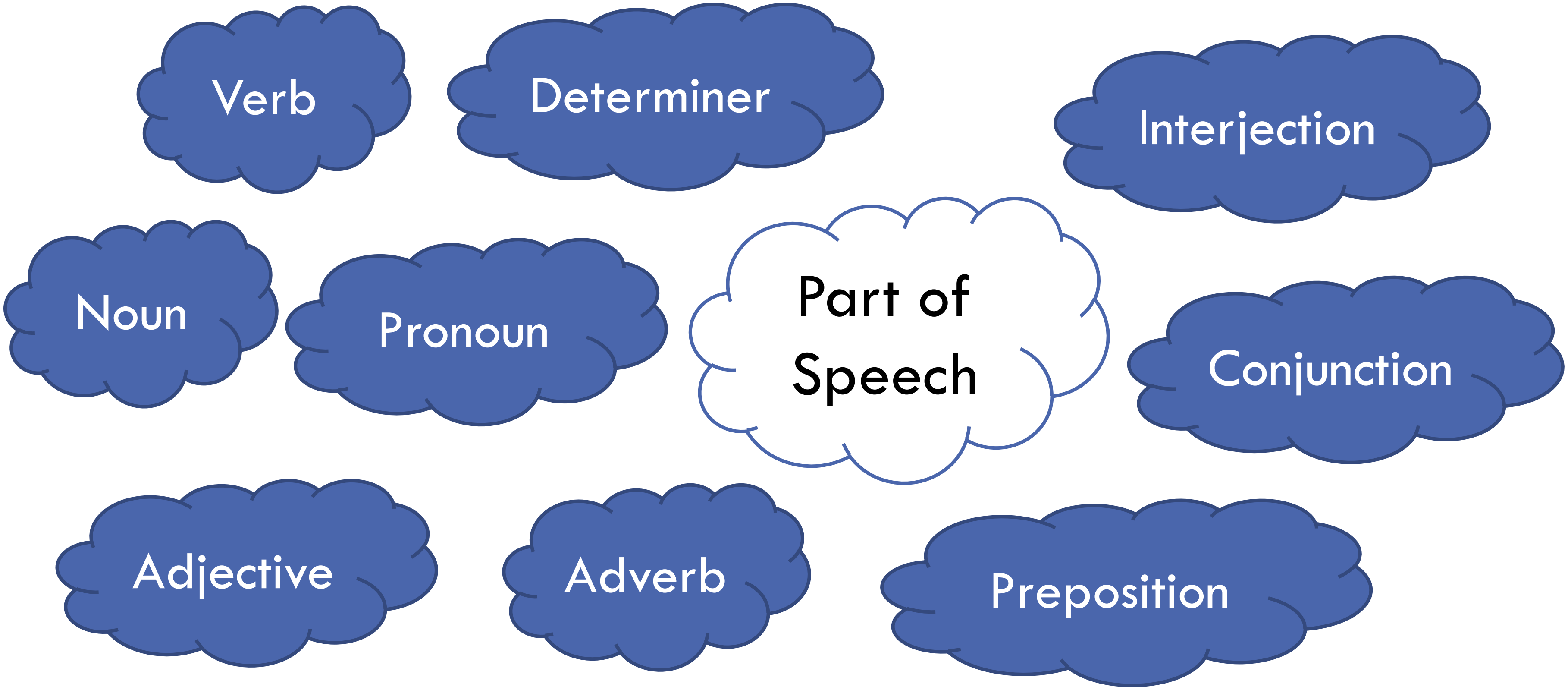
Recap: Lemmatization

- **Lemmatization** aims to convert words to its base form
- We often also need to provide the **Part of Speech** tags for it to work properly

Part of Speech (POS) Tagging

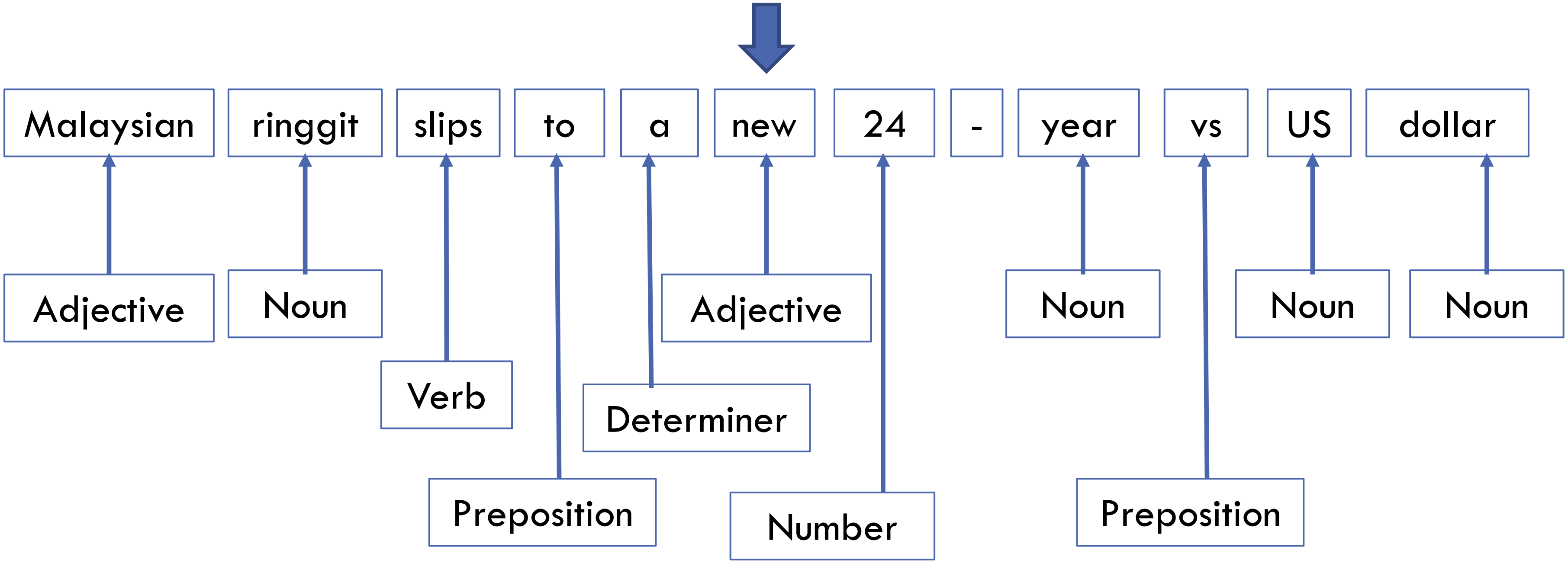
Part of Speech (POS) Tagging

- Each English word can be assigned to one of the 9 **Part of Speech (POS)** tag



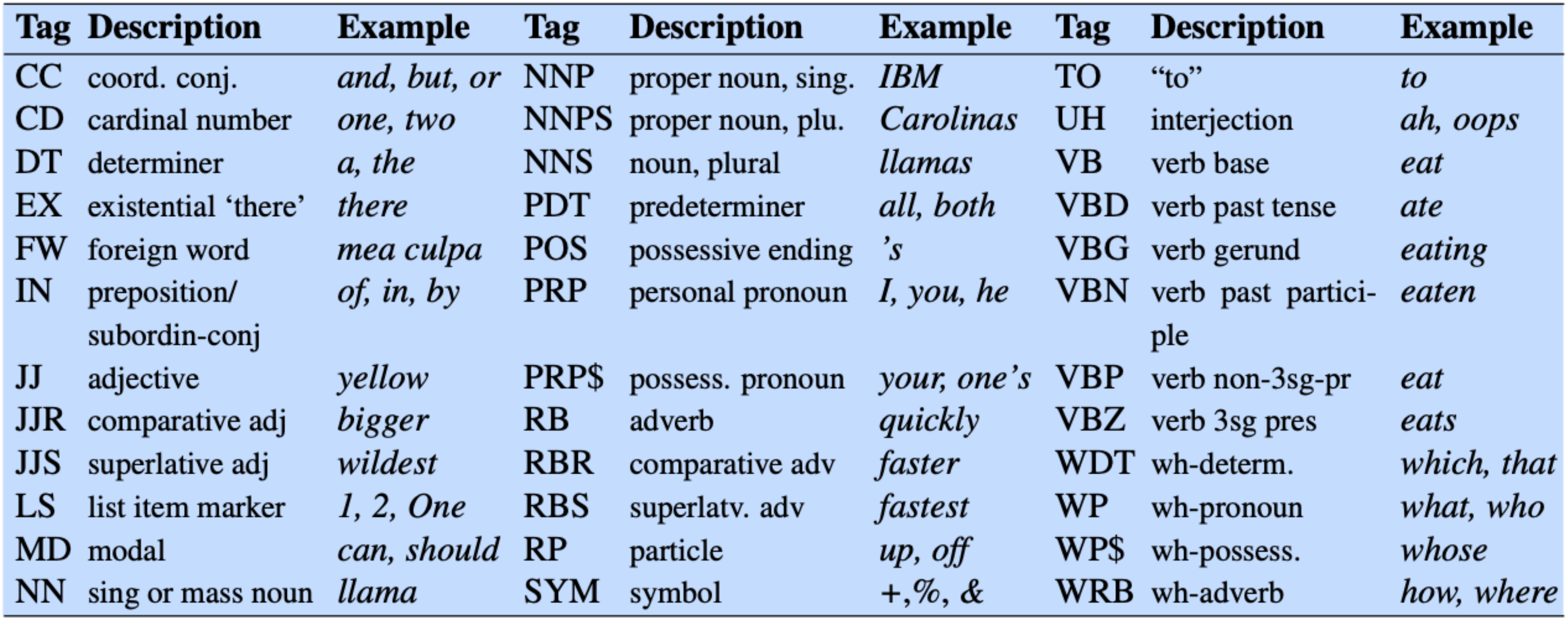
POS Tagging

Malaysian ringgit slips to a new 24 - year low vs US dollar



Penn Treebank POS Tags

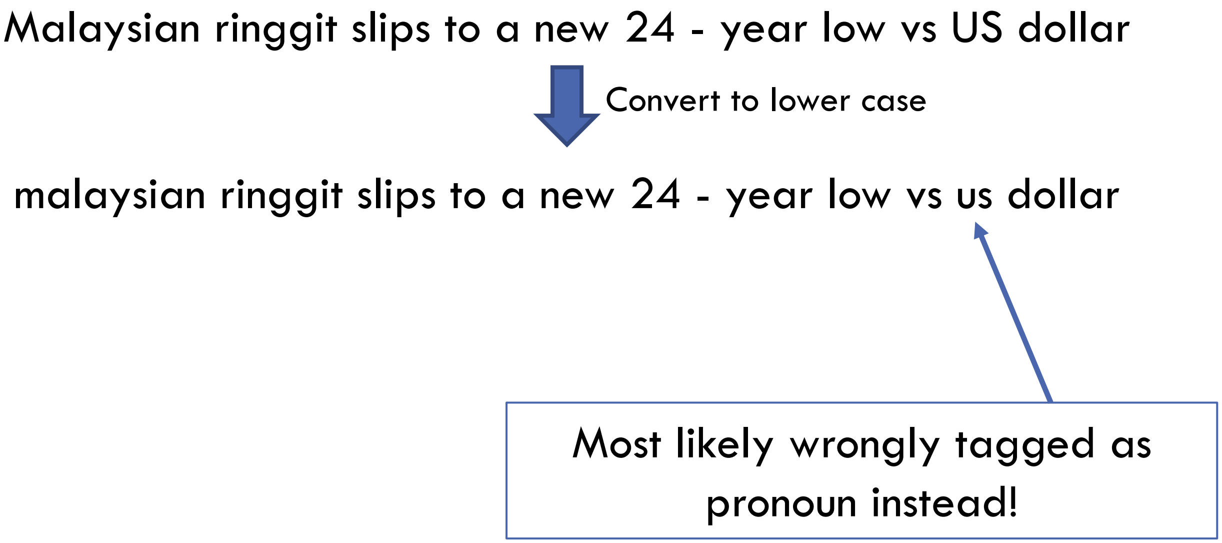
- One of the most commonly used POS tagset is the **Penn Treebank POS Tagset**



Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	“to”	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential ‘there’	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*’s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past participle	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your, one’s*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	+,%, &	WRB	wh-adverb	*how, where*

Order of performing Text Normalization

- Text preprocessing is often performed as a pipeline
- But be careful of the order by which we perform the different text processing tasks
- E.g.



Malaysian ringgit slips to a new 24 - year low vs US dollar

Convert to lower case

malaysian ringgit slips to a new 24 - year low vs us dollar

Most likely wrongly tagged as pronoun instead!

Other NLP Tools

Other NLP Tools

- Other NLP Tools
  - Shallow Parsing (Chunking)
  - Named Entity Recognizer

NATURAL LANGUAGE PROCESSING (NLP) FUNDAMENTALS

NLP TOOLS

DR LEK HSIANG HUI