1.3 Text Normalisation

Edwin Simpson

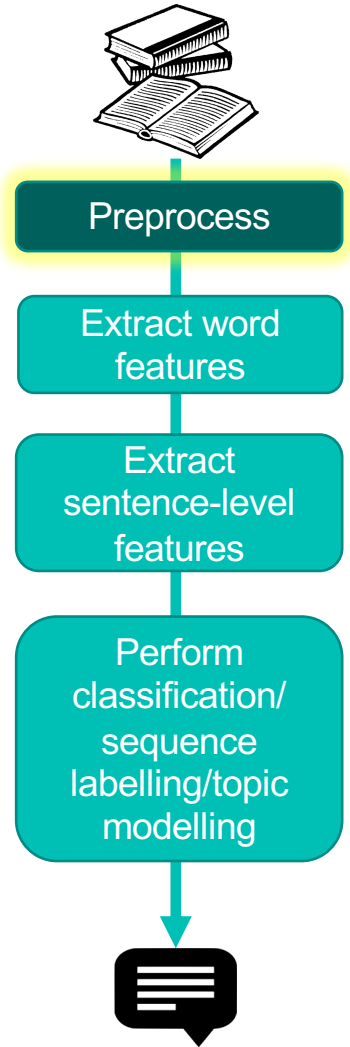
Department of Computer Science,

University of Bristol, UK.

Text Normalisation

- Most text processing methods require us to normalise the text first as a preprocessing step.
- Typical steps are:

1. Tokenisation;
2. Word normalisation;
3. Sentence segmentation.



Tokenisation

- Aim: split a string into a sequence of words
- **Don’t we just split the sequence where the whitespace is?**

What Counts as a Word in English?

- Punctuation
- Abbreviations and acronyms
- Hashtags and URLs
- Should we split contractions like “don’t”?
- Disfluencies such as “err…”
- Should we split suffixes and prefixes, e.g., “re-” or “de-”?
- Multi-word phrases like “rock ‘n’ roll” or “New York”?

Tokenisation

- Aim: split a string into a sequence of **tokens**
- Tokens may include words, punctuation, numbers, multi-word phrases, parts of words and special tokens for things like URLs

Tokenisation Methods

- Penn Treebank defines a standard for how to tokenise English
- Identify common multi-word phrases and contractions using **dictionaries**
  - Keep multi-word phrases as one token
  - Expand “doesn’t” into two tokens, “does” and “n’t”
- Define different types of token using **regular expressions**
  - Split on whitespace
  - Handle punctuation within words like hyphens

Tokenising Other Languages

- Whitespace separates words in Latin-based scripts
- However, in languages like Turkish, words need to be split into sub-words as they have complex internal structure
- Chinese (汉字), Thai (อักษรไทย) and Japanese (日本語), scripts do not mark word boundaries
- For Thai and Japanese, tokenisation is hard and machine learning approaches work best

Tokenising Chinese

姚明进入总决赛 “Yao Ming reaches the finals”

- Can be tokenised in different ways:
  1. 姚明 YaoMing, 进入 reaches, 总决赛 finals
  2. 姚 Yao, 明 Ming, 进入 reaches, 总 overall, 决赛 finals
  3. 姚 Yao, 明 Ming, 进 enter, 入 enter, 总 overall, 决 decision, 赛 game
- We typically use character tokens:
  – They represent units of meaning called **morphemes**
  – Many rare words leads to huge vocabulary

Normalising Word Formats

- Put words into a standard format:
  - Reduce the number of **word types** in the vocabulary
  - Ensure that different forms of a word are treated the same
  - E.g., searching for US or USA
- It's a trade-off between simplifying the vocabulary…
- …and losing information in the original forms of the tokens.

Normalising Word Formats

Step	Example input	Output
Replacing emojis with text	🔥 🎅🏾 🇲🇽	*Fire* *Santa Claus: medium-dark skin tone* *flag: Mexico*
Normalising URLs, hashtags	*http://www.bristol.ac.uk* *#NLProc*	*URL* *HASHTAG*
Stopword removal	*the*	
Case folding	*The* *THE*	*the* *the*
Lemmatisation/ Stemming	*is* *reading*	*be* *read*

Lemmatization

- Words have internal structure:
  - They are composed of stems and affixes
  - “Cats” contains the stem “cat” and an affix “s”
- Replace each word with its root form or **lemma**
- Implement by applying a series of regular expression substitutions
- Example implementation: WordNet Lemmatizer

Quick and Dirty Lemmatization using Porter Stemmer

```
This was not the map we found in Billy Bones's chest, but
an accurate copy, complete in all things-names and heights
and soundings-with the single exception of the red crosses
and the written notes.
```

produces the following stemmed output:

```
Thi wa not the map we found in Billi Bone s chest but an
accur copi complet in all thing name and height and sound
with the singl except of the red cross and the written note
```

Page 20 of Chapter 2, Speech and Language Processing 3rd edition, Jurafsky and Martin (2021).

Summary

- Text is often preprocessed through tokenisation, word normalisation and sentence segmentation;
- Tokenisation is highly language-specific and requires many rules for handling punctuation and special tokens like URLs;
- Lemmatisation replaces words with a root form to simplify the vocabulary;
- Text normalisation is implemented by applying a series of regular expressions.

Reading

- Dan Jurafsky and James H. Martin. Speech and language processing (3rd edition draft). **Chapter 2.**
- https://web.stanford.edu/~jurafsky/slp3/