COMP90042 Natural Language Processing Workshop Week 2

Haonan Li – haonan.li@unimelb.edu.au

9, March 2020

Outline

- Text Processing Applications
- Concepts about Text
- Text Preprocessing
- Practice: Preprocessing
- Porter Stemmer
- Byte-pair Encoding
- Practice: Byte-pair Encoding Algorithm

Text Processing Applications

Text Processing Applications

- Search Engine
  - Google, Baidu, Yahoo!
- Translattion apps
  - Google Translation, Youdao Translation
- Grammer checking apps
  - Grammerly
- Chatbot
  - Siri, Cortana
- And more fancy demos
  - Allennlp demos: Sentiment Analysis, Question Ansering

Comcepts

- Corpus
- Documents
- Sentences
- Words (Tokens)
- Characters

Comcepts

- Corpus
  - a collection of documents.
- Documents
  - one or more sentences.
- Sentences
  - consist of one or more words that are grammatically linked.
- Words (Tokens)
  - Words? Tokens?
- Characters (Extension)
  - Why characters?

Why preprocessing

- Most NLP applications have documents as inputs.
- **Key point**: language is compositional. As humans, we can break these documents into individual components. To understand language, a computer should do the same.
- Preprocessing is the first step.

Preprocessing Steps

Preprocessing Steps

- **Remove unwanted formatting**
  - For example?
- **Sentence segmentation**: break documents into sentences.
- **Word tokenisation**: break sentences into words (tokens).
- **Word normalisation**: transform words into canonical forms.
  - Lemmasation
  - Stemming
- **Stopword removal**: usually refers to the most common words in a language.
  - May be different for different tools.

Morphology

- Inflectional Morphology
  - grammatical variants
  - e.g. swim, swam, swims, swimming
- Derivational Morphology
  - another word with different meaning
  - e.g. Chinese, China
  - e.g. write, writer

Lemmatisation vs Stemming

Lemmatisation vs Stemming

- Both are mechanisms for transforming a token into a canonical form.
- Both operate by applying a series of rewrite operations to remove or replace affixes (primarily suffixes).
- Lemmatisation: Works in conjunction with a lexicon. The goal is to turn the input token into an element of the lexicon using the rewrite rules.
- Stemming: Simply applies rewrite rules, Mainly just strip suffixes from the end of the word.

Practice: Preprocessing

- Python 3 (Virtual environment “Conda” recommend)
- Jupyter Notebook
- NLTK
- Wordnet

The Porter Stemmer

- c = consonant, C = ?
- v = vowel, V = ?
- Word represent: $[C](VC)^m[V]$
- Apply rewrite rules:
  - Step 1:plurals and past participles
  - Step 2, 3, 4: derivational inflections
  - Step 5: tidying up

Practice: Byte-Pair Encoding

- What? Subword Tokenisation
- Why? Misspellings, Rare words, and Multilingual sources
- Concepts: Dictionary, Pair, Vocabulary
- Core idea: Iteratively merge frequent pairs of characters.

Questions ☺