Text Mining and Language Analytics

Lecture 1

Text pre-processing

Dr Stamos Katsigiannis

2023-24

Durham University

What is a language?

"A system of conventional spoken, manual (signed), or written symbols by means of which human beings, as members of a social group and participants in its culture, express themselves."

Encyclopaedia Britannica

"A system of communication consisting of sounds, words, and grammar, or the system of communication used by people in a particular country or type of work."

Cambridge Dictionary

"The words, their pronunciation, and the methods of combining them used and understood by a community."

Merriam-Webster Dictionary

Durham University

Natural Language Processing (NLP)

How to program computers to process and analyse natural language data

Natural Language Processing

- Machine Translation
- Sentiment Analysis
- Information Retrieval
- Information Extraction
- Spam Detection
- Question Answering

And others …

What is text?

Text is a sequence of characters, words, sentences, etc….

The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down a very deep well. First, she tried to look down and make out what she was coming to, but it was too dark to see anything; then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves; here and there she saw maps and pictures hung upon pegs.

Tokenisation

- Tokenisation → The task of chopping a text into pieces (tokens)
- Words typically separated by space character
- Example:

In another moment down went Alice after it, never once considering how in the world she was to get out again. The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down a very deep well.

Durham University

Tokenisation

- **Input:** “Alice in Wonderland”
- **Output:**
  - *Alice*
  - *in*
  - *Wonderland*

- **Token** → An instance of a character sequence
- But what are valid tokens?

Tokenisation: Common issues

Bob's car	• Bob / s / car ? • Bobs / car ? • Bob's / car ?
Newcastle Upon Tyne	• "Newcastle Upon Tyne" ? • Newcastle / Upon / Tyne ? • 1 token or 3 tokens ?
state-of-the-art	• state-of-the-art ? • state / of / the / art ?
I'm	• I'm ? • I / m ? • I / am ?
lowercase	• lowercase ? • lower-case ? • lower / case ?
WHO	• WHO ? • World / Health / Organisation ? • who (pronoun) ?
mi/h	• mi/h ? • mph ? • miles / per / hour ?
rule-of-thumb	• rule-of-thumb ? • rule / of / thumb ?

Durham University

Tokenisation: Language issues (I)

- French
  - **L’ensemble** → 1 or 2 tokens ?
    - L ? / L’ ? / Le ?
- German
  - German noun compounds are not segmented
    - **Lebensversicherungsgesellschaftsangestellter**
    - (=“life insurance company employee”)
    - A compound splitter is needed

Tokenisation: Language issues (II)

- Arabic and Hebrew written right to left
- Although certain terms (e.g. numbers) written left to right

- Arabic: في عام 2020 ، كان هناك جائحة عالمي.
- Hebrew: בשנת 2020 הייתה מגפה עולמית.

- English: In 2020, there was a global pandemic.

Tokenisation: Language issues (III)

- Chinese and Japanese have no spaces between words and may use different scripts
- Chinese
  - 鲍伯现在居住在美国东南部的佛罗里达。

鲍伯	现在	居住	在	美国	东南部	的	佛罗里达	。
Bob	now	lives	in	US	southeast	of	Florida	.

- Japanese
  - フォーチュン500社は情報不足のため時間あた$500K(約6,000万円)

Katakana Hiragana Kanji Rōmaji

  - Available information or queries may be in one type of script

Tokenisation: Maximum matching (I)

- Chinese words are composed of characters
  - Characters are generally 1 syllable and 1 morpheme
  - Average word length → 2.4 characters

- Standard baseline algorithm → **Maximum Matching**
  - Given a word list (dictionary) of Chinese and a string
  - Start a pointer at the beginning of the string
  - Find the longest word in the dictionary that matches the string starting at the pointer
  - Move the pointer over the word in string

Repeat

Tokenisation: Maximum matching (II)

Maximum matching

- Greedy algorithm
- Doesn't generally work well in English

e.g. "Thetabledownthere" $\longrightarrow$ Theta / bled / own / there

instead of

The / table / down / there

- But works astonishingly well in Chinese
  - Modern probabilistic segmentation algorithms work even better

Durham University

Stop words

- Stop words
  - Most common words in a natural language
  - Hold no semantic meaning of their own
  - e.g. “the”, “a”, “an”, “and”, “to”,

- Stop words lists can be used to remove stop words

- “The table is red” → table red
  - Stop words: “The”, “is”

Stop words list: English

English stop words list from the NLTK Python package:

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours',
'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they',
'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am',
'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the',
'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into',
'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under',
'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few',
'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can',
'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn',
"couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't",
'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn',
"wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Custom lists can be used based on the application

Durham University

Stop words removal (I)

Stop words removal leads to:

- Reduced irrelevance
  - Restricts analysis to meaningful words
  - Reduces the noise stop words introduce to meaning
- Reduced feature dimension
  - Reduces the number of extracted tokens

- "This film was not great" → film great
  - Stop words → "This", "was", $\boxed{\text{"not"}}$

Is this a good idea?

Stop words removal (II)

he picked up the cake,
and the rake, and the gown,
and the milk, and the strings,
and the books, and the dish,
and the fan, and the cup,
and the ship, and the fish.
and he put them away.
then he said, 'that is that.'
and then he was gone
with a tip of his hat.

~~he~~ picked up ~~the~~ cake,
~~and~~ ~~the~~ rake, ~~and~~ ~~the~~ gown,
~~and~~ ~~the~~ milk, ~~and~~ ~~the~~ strings,
~~and~~ ~~the~~ books, ~~and~~ ~~the~~ dish,
~~and~~ ~~the~~ fan, ~~and~~ ~~the~~ cup,
~~and~~ ~~the~~ ship, ~~and~~ ~~the~~ fish.
~~and~~ ~~he~~ put ~~them~~ away.
~~then~~ ~~he~~ said, '~~that~~ ~~is~~ ~~that~~.'
~~and~~ ~~then~~ ~~he~~ ~~was~~ gone
~~with~~ ~~a~~ tip ~~of~~ ~~his~~ hat.

Case folding

- Case folding → Reduce all letters to lower case
  - e.g. "This table is red" → this table is red

- **Exception:** Upper case in mid-sentence ?
  - **Fed vs. fed** ("Federal Reserve" vs. past tense of verb "feed")
  - **General Motors vs. general motors** (company vs. two valid words)

- Sometimes best to lower case everything
  - Authors typically ignore correct capitalisation in casual text, e.g. Twitter, Facebook, online comments, etc.

Durham University

Normalisation to terms

- Some words need to be "normalised" into the same form
  - U.K. and UK
  - lower-case and lowercase

- Results to terms → Normalised word types

- Typically achieved by implicitly defining equivalence classes of terms, e.g.:
  - Removing periods (U.S.A → USA)
  - Removing hyphens (anti-bacterial → antibacterial)
  - Enforcing spelling variants (customize → customise, center → centre)
  - Using aliases (MS Word → Microsoft Word)
  - Spelling correction (colection → collection)

Durham University

Normalisation to terms: Other languages

- Accent removal
  - e.g. French: réunion → reunion, German: mäuse → mause
- Different scripts
  - e.g. Japanese

Kanji	Hiragana	Katakana	Rōmaji	(English)
私	わたし	ワタシ	watashi	(I, me)

- Tokenisation and normalisation may depend on the language
  - MIT vs. mit
  - (Massachusetts Institute of Technology vs. German “mit”=“with”)

- **Attention:** Identical normalisation rules must be applied to both input and query text

Durham University

Normalisation to terms: Synonyms and homonyms

- Synonyms: Different word same meaning
  - e.g. Car vs. Automobile
  - Can be addressed by hand-crafted equivalence classes
- Homonyms: Same word different meaning
  - e.g. address (noun) vs. address (verb)
  - Can be addressed by syntax analysis and tagging

- What about spelling mistakes?
  - computer vs. compiter ← Easy to correct using a lexicon
  - bear vs. beer ← **Context needed!**

Lemmatisation

- Reduce inflectional/variant forms to base form
  - am, are, is → be
  - car, cars, car's, cars' → car
  - updated, updates, updating → update

- e.g. My friends are living in flats → My friend be live in flat

- "Proper" reduction to dictionary headword needed!

Stemming

- Stemming → Reduction of terms to their stems

- Stem → The core meaning-bearing unit of a word
  - e.g. accepted → $\underbrace{\text{accept}}_{\text{stem}}$ $\underbrace{\text{-ed}}_{\text{suffix}}$

- Stemming algorithms are typically rules-based

For example compressed and compression are both accepted as equivalent to compress

$\xrightarrow{\text{stemming}}$

For **exampl compress** and **compress ar** both **accept** as **equival** to compress

Porter's stemming algorithm

- Most common English stemming algorithm
- Consists of 5 phases of word reductions
- Each phase contains rules that apply to the longest suffix

Example:

SSES	→ SS
IES	→ I
SS	→ SS
S	→ ∅
ATIONAL	→ ATE
ISER	→ ISE
ATOR	→ ATE
ABLE	→ ∅
…	

caresses	→ caress
ponies	→ poni
caress	→ caress
cats	→ cat
relational	→ relate
adviser	→ advise
operator	→ operate
adjustable	→ adjust
…	

Test Porter's stemmer online: http://textanalysisonline.com/nltk-porter-stemmer

Other stemmers

- Lovins stemmer
- Dawson stemmer
- Krovetz stemmer
- Xerox stemmer
- N-Gram stemmer
- Snowball stemmer (multi-lingual)
- …

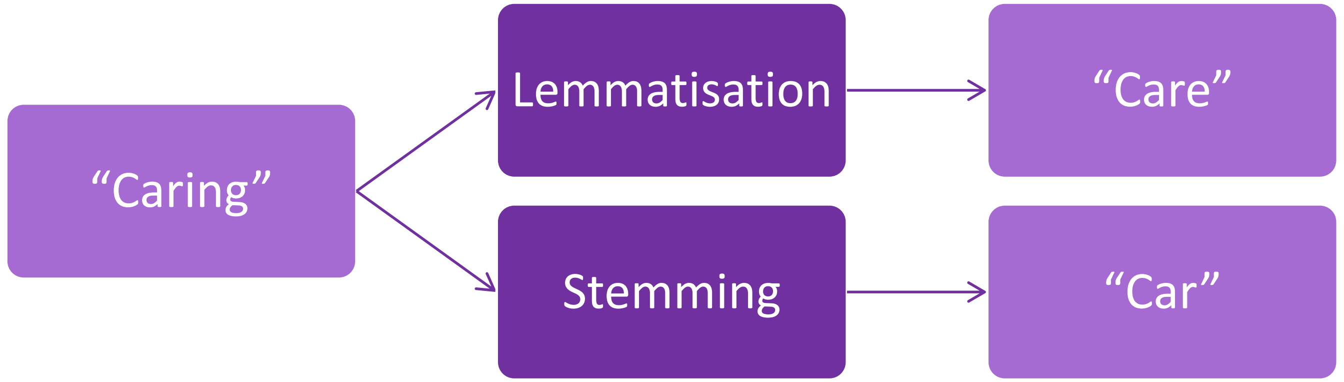
Durham University

Lemmatisation vs. Stemming

- Lemmatisation considers the context and converts the word to its meaningful base form

- Stemming removes/alters the last few characters (suffixes)
  - Often leads to incorrect meanings and spelling errors

Example:



Questions?