

COMP90042 Workshop Week 2

Introduction and Pre-processing

Zenan Zhai

The University of Melbourne

9 March 2014

Table of Contents

Introduction

Pre-processing

Table of Contents

Introduction

Pre-processing

Canvas - Discussion Board

Subject Coordinator

- Dr. Jey Han Lau (laujh@unimelb.edu.au)

Me

- Zenan Zhai (zenan.zhai@unimelb.edu.au)
- Workshop slides available at
<https://zenanz.github.io/comp90042-2020/>

Table of Contents

Introduction

Pre-processing

Pre-processing Pipeline

1. Formatting
2. Sentence Segmentation
3. Tokenisation
4. Normalisation
5. Stopword Removal

Formatting

Web page you see 😊



The screenshot shows the BBC News website. At the top is a navigation bar with the BBC logo, a 'Sign in' button, and links to Home, News, Sport, Reel, Worklife, Travel, Future, Culture, and More. A search bar is on the right. Below this is a red banner with the word 'NEWS' in white. Underneath the banner is a secondary navigation bar with links to Home, Video, World, Asia, UK, Business, Tech, Science, Stories, Entertainment & Arts, Health, World News TV, In Pictures, Reality Check, Newsbeat, and More. Below this is a third navigation bar with links to World, Africa, Australia, Europe, Latin America, Middle East, and US & Canada. The main content area features a large video player showing a man in a blue uniform and a black cap, possibly a police officer, standing in front of a large potted plant. The video player has a play button and a red 'LIVE' icon. Below the video player is a section with three smaller video thumbnails. The first thumbnail shows a man in a suit and glasses, with the text 'WHO head defends coronavirus role in China' and a duration of 4:10. The second thumbnail shows two people wearing face masks, with the text 'The positive things we learnt from quarantine' and a duration of 3:46. The third thumbnail shows a group of children, with the text 'Children's coronavirus questions - answered' and a duration of 2:41. Each thumbnail has the BBC logo at the bottom.

BBC Sign in Home News Sport Reel Worklife Travel Future Culture More Search

NEWS

Home Video World Asia UK Business Tech Science Stories Entertainment & Arts Health World News TV In Pictures Reality Check Newsbeat More

World Africa Australia Europe Latin America Middle East US & Canada

LIVE Lowest virus cases in China since crisis began

How to wash your hands - in 20 seconds BBC 0:24

WHO head defends coronavirus role in China 4:10 BBC

The positive things we learnt from quarantine 3:46 BBC

Children's coronavirus questions - answered 2:41 BBC

Formatting

Source code you get ☹️

```
1 <!DOCTYPE html>
2 <html lang="en-gb" class="b-pw-1280 no-touch" id="responsive-news">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=1">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
7   <meta name="google-site-verification" content="7x6b1127naCKoqt94L4-D-Of1fdr5gxr27u2Vtj9YI">
8   <link href="//static.bbc.co.uk" rel="preconnect" crossorigin>
9   <link href="//m.files.bbc.co.uk" rel="preconnect" crossorigin>
10  <link href="//nav.files.bbc.co.uk" rel="preconnect" crossorigin>
11  <link href="//chef.bbc.co.uk" rel="preconnect" crossorigin>
12  <link rel="dns-prefetch" href="//mybbc.files.bbc.co.uk">
13  <link rel="dns-prefetch" href="//s1.bbc.co.uk/">
14  <link rel="dns-prefetch" href="//sa.bbc.co.uk/">
15  <link rel="dns-prefetch" href="//ichef.bbc.co.uk">
16
17  <script type="text/javascript">var domain = 'co.uk';var edition = '';var prettyEdition = edition;if (window.NewsPage && window.NewsPage.edition) {edition =
18  window.NewsPage.edition;prettyEdition = edition === 'northernireland' ? 'Northern Ireland' : edition.charAt(0).toUpperCase() + edition.slice(1);}var pathEdition = edition.length
19  > 0 ? "/" + edition.toLowerCase() : "";var sf_async_config = sf_async_config || {};var sf_startpt=(new Date()).getTime();sf_async_config.domain = 'www' + ".bbc." +
20  domain;sf_async_config.uid = 50924;sf_async_config.title = window.document.title.replace(/##edition##/, prettyEdition);sf_async_config.path = 'bbc.' + domain +
21  '/news/live/world-51796781' + pathEdition;sf_async_config.sections = 'News, News - world, News - LIV, News - world - LIV, News - news-category';sf_async_config.mobileApp =
22  undefined;</script>
23
24  <title>Coronavirus: Lowest increase in Chinese cases since crisis began - BBC News</title>
25
26  <meta name="description" content="While the growth in China and South Korea is slowing, cases increase in the US and Europe.">
27  <meta name="robots" content="NOODP,NOYDIR">
28
29  <link rel="canonical" href="https://www.bbc.co.uk/news/live/world-51796781">
30
31  <link rel="alternate" hreflang="en-gb" href="https://www.bbc.co.uk/news">
32  <link rel="alternate" hreflang="en" href="https://www.bbc.com/news">
```

The title is here!

Off-the-shelf packages come to aid (e.g. beautifulsoup)

Word-level Tokenisation

'Coronavirus: Lowest virus cases in China since crisis began.'



['Coronavirus', ':', 'Lowest', 'cases', 'in', 'China', 'since', 'crisis',
'began', '.']

- Tokenisation
 - ▶ Rule-based / Machine Learning
 - ▶ Subject to languages/domains (e.g. Medicine Chemistry)
- Off-the-shelf implementations
 - ▶ NLTK
<https://www.nltk.org/>
 - ▶ OpenNLP
<https://opennlp.apache.org/>
 - ▶ StanfordNLP
<https://stanfordnlp.github.io/stanfordnlp/>

Word-level Tokenisation

'Coronavirus: Lowest virus cases in China since crisis began.'



['Coronavirus', ':', 'Lowest', 'cases', 'in', 'China', 'since', 'crisis',
'began', '.']

- Tokenisation
 - ▶ Rule-based / Machine Learning
 - ▶ Subject to languages/domains (e.g. Medicine Chemistry)
- Off-the-shelf implementations
 - ▶ NLTK
<https://www.nltk.org/>
 - ▶ OpenNLP
<https://opennlp.apache.org/>
 - ▶ StanfordNLP
<https://stanfordnlp.github.io/stanfordnlp/>

Wait, why did you skip sentence segmentation?

Why tokenisation?

Why tokenisation?

- Easier for machine to understand.

Why tokenisation?

- Easier for machine to understand.

Is there a better unit of text than word?

- Let's take the word 'Coronavirus' as an example.
 - ▶ Do you know this word before the outbreak?
 - ▶ If so, do you understand it when you first saw it?

Okay, now I know sub-words are fantastic.
Tell me how to get sub-word vocab?

Byte-pair Encoding

1. Break the entire piece of text into single characters tokens.
2. Count frequency of two tokens being together.
3. Merge most frequent pair of characters into one token.
4. Repeat from step 2.

BPE in action

Coronavirus: Lowest virus cases in China since crisis began.

```
Vocab = defaultdict(<class 'int'>, {'C': 2, 'o': 3, 'r': 4, 'n': 5, 'a': 4, 'v': 2, 'i': 7, 'u': 2, 's': 8, ' ': 1, '</w>': 1})
Vocab['</w>'] = 1
Vocab['in'] = 1
Vocab['China'] = 1
Vocab['since'] = 1
Vocab['crisis'] = 1
Vocab['began'] = 1

=====
Tokens Before BPE
Tokens: defaultdict(<class 'int'>, {'C': 2, 'o': 3, 'r': 4, 'n': 5, 'a': 4, 'v': 2, 'i': 7, 'u': 2, 's': 8, ' ': 1, '</w>': 9, 'L': 1, 'w': 1, 'e': 4, 't': 1, 'c': 3, 'h': 1, 'b': 1, 'g': 1, '.': 1})
Number of tokens: 20
=====
Iter: 0
Best pair: ('s', '</w>')
Tokens: defaultdict(<class 'int'>, {'C': 2, 'o': 3, 'r': 4, 'n': 5, 'a': 4, 'v': 2, 'i': 7, 'u': 2, 's': 5, ' ': 1, '</w>': 6, 'L': 1, 'w': 1, 'e': 4, 't': 1, 'c': 3, 'h': 1, 'b': 1, 'g': 1, '.': 1, 's</w>': 3})
Number of tokens: 21
=====
Iter: 1
Best pair: ('i', 'n')
Tokens: defaultdict(<class 'int'>, {'C': 2, 'o': 3, 'r': 4, 'n': 2, 'a': 4, 'v': 2, 'i': 4, 'u': 2, 's': 5, ' ': 1, '</w>': 6, 'L': 1, 'w': 1, 'e': 4, 't': 1, 'c': 3, 'h': 1, 'b': 1, 'g': 1, '.': 1, 's</w>': 3, 'in': 3})
Number of tokens: 22
=====
Iter: 2
Best pair: ('v', 'i')
Tokens: defaultdict(<class 'int'>, {'C': 2, 'o': 3, 'r': 4, 'n': 2, 'a': 4, 'v': 0, 'i': 2, 'u': 2, 's': 5, ' ': 1, '</w>': 6, 'L': 1, 'w': 1, 'e': 4, 't': 1, 'c': 3, 'h': 1, 'b': 1, 'g': 1, '.': 1, 's</w>': 3, 'in': 3, 'vi': 2})
Number of tokens: 23
```

Normalisation

Normalisation techniques

- Lower casing
- Spelling correction
- Abbreviation expansion
- Removing Morphology
- ...

What is normalisation?

Normalisation

Normalisation techniques

- Lower casing
- Spelling correction
- Abbreviation expansion
- Removing Morphology
- ...

What is normalisation?

- Converting words to a standard format

Why do we want normalisation?

Normalisation

Normalisation techniques

- Lower casing
- Spelling correction
- Abbreviation expansion
- Removing Morphology
- ...

What is normalisation?

- Converting words to a standard format

Why do we want normalisation?

Normalisation

Normalisation techniques

- Lower casing
- Spelling correction
- Abbreviation expansion
- Removing Morphology
- ...

What is normalisation?

- Converting words to a standard format

Why do we want normalisation?

- Reduce noises

Normalisation

Normalisation techniques

- Lower casing
- Spelling correction
- Abbreviation expansion
- Removing Morphology
- ...

What is normalisation?

- Converting words to a standard format

Why do we want normalisation?

- Reduce noises
- Reduce data sparsity

Morphology

- Inflectional Morphology
 - ▶ Grammatical variants
- Derivational morphology
 - ▶ Another word with different meaning

Inflectional Morphology

began → begin

cases → case

Derivational morphology

Ethiopia → Ethiopian

Lemmatization and Stemming

Rule-based deterministic algorithm for normalisation

Lemmatisation	Stemming
Remove all inflections Matches with lexicons Product: Lemma	Remove all suffixes No matching required Product: Stem

```
import nltk

sentence = ['Coronavirus', ':', 'Lowest', 'virus', 'cases', 'in', 'China', 'since', 'crisis', 'began', '.']
lemmatiser = nltk.stem.wordnet.WordNetLemmatizer()
stemmer = nltk.stem.porter.PorterStemmer()

# Code below from ...
def lemmatise(word):
    lemma = lemmatiser.lemmatize(word, 'v')
    if lemma == word:
        lemma = lemmatiser.lemmatize(word, 'n')
    return lemma
# End of copied code

lemmatised_sent = [lemmatise(word) for word in sentence]
stemmed_sent = [stemmer.stem(word) for word in sentence]

print('Lemmatised Sentence: ', lemmatised_sent)
print('Stemmed Sentence: ', stemmed_sent)

Lemmatised Sentence: ['Coronavirus', ':', 'Lowest', 'virus', 'case', 'in', 'China', 'since', 'crisis', 'begin', '.']
Stemmed Sentence: ['coronaviru', ':', 'lowest', 'viru', 'case', 'in', 'china', 'sinc', 'crisi', 'began', '.']
```

Porter Stemmer

Symbols

- Case sensitive

$V \rightarrow$ sequence of vowels $C \rightarrow$ sequence of consonants
 $v \rightarrow$ a single vowel $c \rightarrow$ a single consonant

Measure

1. Convert **STEM of the word** in the form of $[C](VC)^m[V]$
2. Take m as measure

Rules

- Example: ($m > 0$ *not* $*o$) $e \rightarrow \text{NULL}$
- $*o =$ stem ends *cvc* and second *c* is not *w*, *x* or *y*
(e.g. -HIL, -HOP)

Porter Stemmer - Exercise

Rules

1. ($m > 0$) ational \rightarrow ate
2. ($m > 1$) ate \rightarrow NULL

computational

Porter Stemmer - Exercise

Rules

1. ($m > 0$) ational \rightarrow ate
2. ($m > 1$) ate \rightarrow NULL

computational

Step	Rule	Stem	Form	m	Result
1	1	comput	$[C](VC)^2$	2	compute

Porter Stemmer - Exercise

Rules

1. ($m > 0$) ational \rightarrow ate
2. ($m > 1$) ate \rightarrow NULL

computational

Step	Rule	Stem	Form	m	Result
1	1	comput	$[C](VC)^2$	2	compute
2	2	comput	$[C](VC)^2$	2	comput

What about national?

Stopword

Stopword

- Short functional words that are very common
- Examples (NLTK): me, what, by, with, into, above ...

How would stopwords affect text classification?

Take away

1. Pre-processing pipeline

2. Tokenisation

- ▶ word level
- ▶ sub-word level (BPE)

3. Normalization

- ▶ Morphology (inflectional v.s. derivational)
- ▶ Lemmatisation v.s. Stemming