Web and Social Media Analytics

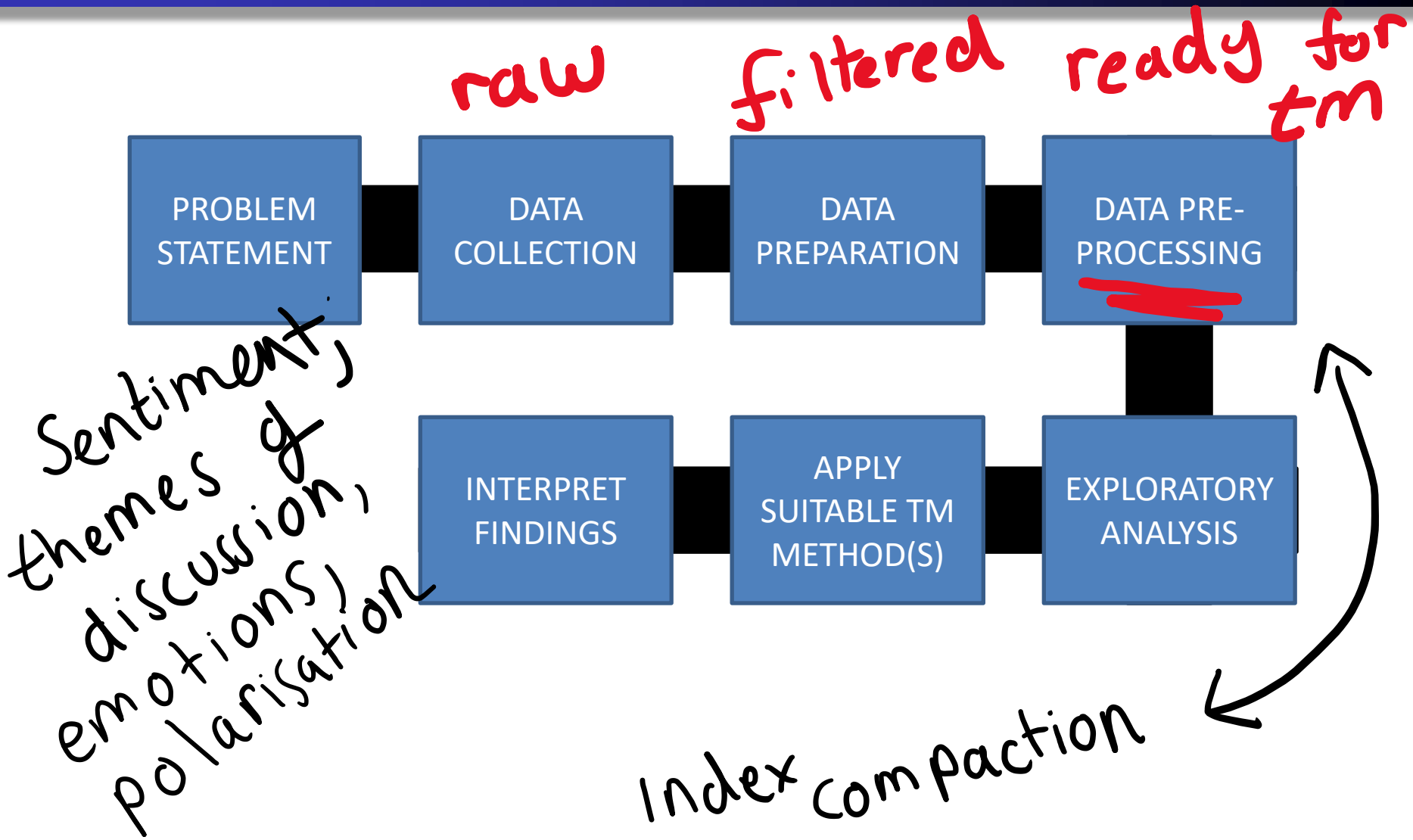
Social media (feature extraction and tokenisation)

Dr Philip Worrall

School of Computer Science and Engineering
115 New Cavendish Street
University of Westminster
London, W1W 6UW
worralph@westminster.ac.uk

LW9

The text mining process…



Week 9

Plan for today

- **Data preparation in Python**
  - Filtering and sorting data
  - Merging data from different sources
- **Building a pre-processing pipeline in Python**
  - Contraction expansion
  - Common case
  - Removal of stopwords
  - Removal of punctuation (numeric substitution)
  - Stemming/Lemmatisation

we may apply some/all of these

loss of precision?

Data preparation…

- Filtering
  - Data covering a specific period
  - Length above a threshold
  - Data in the same frame (language)
  - Data mentioning a particular entity or keyword
    - E.g. place, person or organisation

- Merging
  - Data from across multiple sources (e.g. subreddits)
  - Multi-period date collection

Filtering…

Consider an initial data frame consisting of 3 rows

```
row1 = {"date": pd.to_datetime("2023-03-19 13:34:55"), "author": "AutoModerator"}
row2 = {"date": pd.to_datetime("2019-03-11 12:34:55"), "author": "Md18922"}
row3 = {"date": pd.to_datetime("2012-01-22 12:34:55"), "author": "RiverLover27"}

df = pd.DataFrame([row1,row2,row3])
print(df)
```

```
                  date         author
0 2023-03-19 13:34:55  AutoModerator
1 2019-03-11 12:34:55        Md18922
2 2012-01-22 12:34:55   RiverLover27
```

N.B time/date is typically set to UTC

Filtering examples (1)

```
from datetime import datetime, timedelta
df_filtered = df[df["date"] >= datetime.now() - timedelta(days = 28)]
df_filtered
```

	date	author
0	2023-03-19 13:34:55	AutoModerator

compliment operator

```
df_filtered = df[~df["author"].isin(["AutoModerator"])]
df_filtered
```

	date	author
1	2019-03-11 12:34:55	Mdl8922
2	2012-01-22 12:34:55	RiverLover27

Filtering examples (2)

Removing data where the text length is below 10 chars

```
row1 = {"author": "AutoModerator", "text": "This submission is not accepting new posts"}
row2 = {"author": "Md18922", "text": "None"}
row3 = {"author": "RiverLover27", "text": "Yeah right lol, I have 4 at a minimum. Who's only eating two??!!!"}

def nchars(row):
  return len(row["text"])

df = pd.DataFrame([row1,row2,row3])
df["nchars"] = df.apply(nchars, axis=1)
df_filtered = df[df["nchars"] >= 10]
df_filtered
```

comparison operator

	author	text	nchars
0	AutoModerator	This submission is not accepting new posts	42
2	RiverLover27	Yeah right lol, I have 4 at a minimum. Who's o...	65

NB axis=1 ⇒ by row
=0 ⇒ by column

Filtering examples (3)

```
            author                 date lang
0     AutoModerator 2023-03-19 13:34:55   en
1      caprisun1990 2023-03-19 14:44:46   en
2  ExplodingDogs82 2023-03-19 14:46:37   en
3  Beatrix_-_Kiddo 2023-03-19 13:54:08   en
4      RiverLover27 2023-03-19 14:46:40   en
5     Excession-OCP 2023-03-19 14:16:01   en
6        Ok-Bag3000 2023-03-19 13:53:08   so
7   shinysilvereye 2023-03-19 15:29:23   en
8      MikeSizemore 2023-03-19 14:03:19   en
9           Md18922 2023-03-19 13:44:30   en
```

← the filter

```
df_filtered = df[df["lang"] == "en"]
print(df_filtered[["author", "date", "lang"]])
```

a named subset of columns

See: https://www.loc.gov/standards/iso639-2/php/code_list.php

Filtering examples (4)

Excluding data that does not mention an important entity or phrase

```
row1 = {"text": "Next I'm buying Coca-Cola to put the cocaine back in"}
row2 = {"text": "It's a new day in America."}
row3 = {"text": "Congratulations to the Astronauts that left Earth today. Good choice"}

df = pd.DataFrame([row1,row2,row3])
df_filtered = df[df["text"].str.contains("Coca-Cola")]
df_filtered
```

	text
0	Next I'm buying Coca-Cola to put the cocaine b...

Merging data frames…

Common scenario.

- Data collected across different subreddits/videos
- Day1…Day2…Day3…

```
import pandas as pd

row1 = {"text": "Next I'm buying Coca-Cola to put the cocaine back in"}
row2 = {"text": "It's a new day in America."}
row3 = {"text": "Congratulations to the Astronauts that left Earth today. Good choice"}

rowA = {"text": "teamwork makes the dream work."}
rowB = {"text": "this is what happens when you don't recycle your pizza boxes"}
rowC = {"text": "hello literally everyone"}

left_df = pd.DataFrame([row1,row2,row3])
right_df = pd.DataFrame([rowA,rowB,rowC])

df_combined = pd.concat([left_df, right_df], axis=0, ignore_index=True)
```

the data frames to merge

Merging data frames…

	text
0	Next I'm buying Coca-Cola to put the cocaine b...
1	It's a new day in America.
2	Congratulations to the Astronauts that left Ea...
3	teamwork makes the dream work.
4	this is what happens when you don't recycle yo...
5	hello literally everyone

Encoding textual data…

DM

Customer Database

Monthly Spend	Complaints Made	Customer Age	…	…	…	Has Left
6	1	34	…	…	…	TRUE
10	0	37	…	…	…	FALSE
45	1	18	…	…	…	TRUE
33	2	22	…	…	…	TRUE
22	0	19	…	…	…	FALSE
6	9	54	…	…	…	FALSE
9	1	63	…	…	…	FALSE
12	5	43	…	…	…	TRUE
…	…	..	..	..	..	..

TM

```
2014-03-15 23:58:43 @Cameron_839 @Duncanaallan excuse me..gravity is amazing, do
nt blame Duncan, although Sandra bullock with short hair is a no no
2014-03-15 23:58:36 5 mins into Gravity and I'm already freaking out
2014-03-15 23:58:33 What a nice game: http://t.co/U6llTs1E3c #flappybird #iPhone
 #apple #game http://t.co/nw6d0QoZK1
2014-03-15 23:58:33 My mom loves this game: http://t.co/dT3hD5FaJW #flappybird #
iOS http://t.co/I09TVwWom9
2014-03-15 23:58:32 RT @EthanPabrezis: Best recent iPhone game : http://t.co/UHG
7v58DBw #flappybird #apple #flappy
2014-03-15 23:58:29 If this movie doesn't end w/Sandy Bullock making it back to
earth only to find the apes are now in charge, Ima be disappointed. #gravity
2014-03-15 23:58:29 I just realised that the episode of Futurama where Bender ge
ts shot out a space ship and floats through space is basically the film Gravity
2014-03-15 23:58:29 I can't get good score in this game :( http://t.co/KZb8JW5by
U #flappybird #apple #iOS #game #flappy ##fun http://t.co/2tvRAPl1xH
2014-03-15 23:58:26 RT @EarthaBelisle: This game has a simple and intuitive desi
```

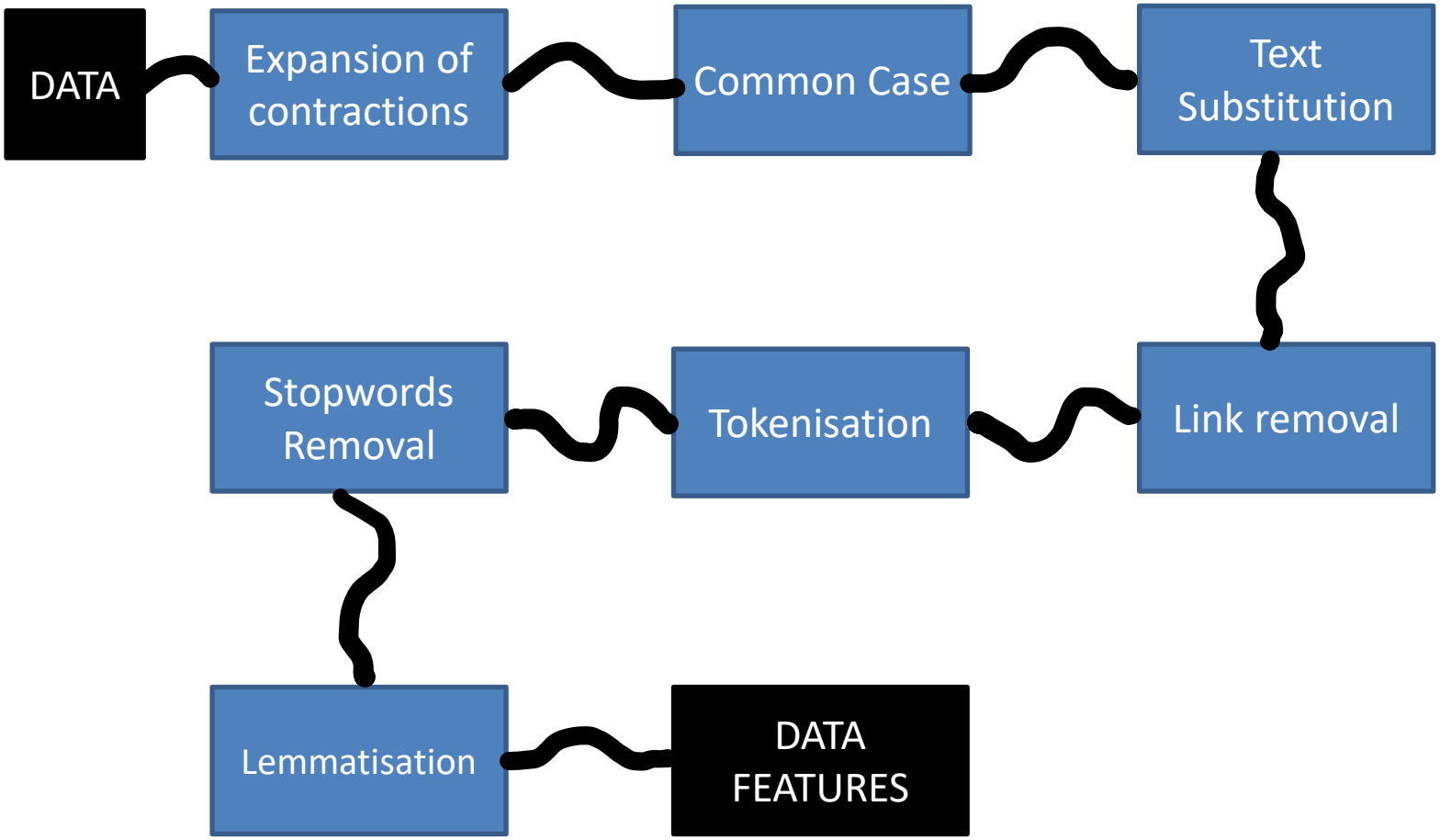
The bag of words model...

The bag of words model (BoW)

The bag of words model (BoW) is one of the simplest representations used to model textual data. In the BoW model, text is recorded using 1s and 0s to indicate the presence of a word in a piece of text. All unique words are used as features.

document \ word	Happy	happy	care	caring
doc_1	1	0	0	1
doc_2	0	1	1	0

The Pre-Processing Pipeline…



Expansion of contractions…

- The inflationary stage of pre-processing

: a shortening of a word, syllable, or word group by omission of a sound or letter

also **:** a form produced by such shortening

> "They'll" is a *contraction* for "they will."

Source: Merriam Webster

shouldn't **wouldn't** **couldn't** **don't** **she'll**

would not

She shall

(Roll your own) Contractions…

```
contractions = {"I'm": "I am", "won't": "will not", "don't": "do not"}

text = "Next I'm buying Coca-Cola to put the cocaine back in"
words = []
for word in text.split():
  if word in contractions:
    words.append(contractions[word])
  else:
    words.append(word)
text = ' '.join(words)
text
```

```
'Next I am buying Coca-Cola to put the cocaine back in'
```

Using the contractions package…

```
!pip install contractions
import contractions

text = "Next I'm buying Coca-Cola to put the cocaine back in"
words = []
for word in text.split():
  words.append(contractions.fix(word))
text = ' '.join(words)
text
```

Adding the first stage to the pipeline..

```
!pip install contractions
import contractions

texts = ["Next I'm buying Coca-Cola to put the cocaine back in"]

# stage 1
def text_with_contractions(text):
  words = []
  for word in text.split():
    words.append(contractions.fix(word))
  return ' '.join(words)

for text in texts:
  text = text_with_contractions(text)
  print(text)
```

Adding a common case stage…

```python
# stage 2 - common case
def text_lowercase(text):
    return text.lower()
```

Adding text substitution…

- In **unstructured text**, we are likely to come across *prices*, *sizes*, *volumes*, *weights* etc.
- These entities are routinely substituted with a marker indicating that a specific entity was used without recording its precise value.
- *For example*

  $1.99 => ${price}

  11/01/2023 => ${date}

  291kg => ${weight}

Adding text substitution…

- Regular expressions are special sequences of text that can be used to define a **search pattern**.
- These sequences can be used to identify relevant matches in a set of text.

\@ ~ matches the ampersand symbol,

? ~ match once, + ~ match 1 or more times

\d ~ matches a digit, \s ~ matches a space

\w ~ matches an alphanumeric character

2-4 ~ a digit between 2 and 4 inclusive

Examples of Regular Expression patterns in Python

Adding text substitution…

- In Python regular expressions are handled in the **re** package, part of the std library

```
# stage 3 - text substitution
import re
def replace_markers(text):
    pattern = r"\$\d+"
    text = re.sub(pattern, "${price}", text)
    return text
```

How to remove multiple exclamation marks?

Replacing URLs with a marker…

```
# stage 4 - http link replacement
import re
def replace_http_links(text):
  pattern = r"(https://|http://)[A-Za-z0-9\/\.]+"
  text = re.sub(pattern, "${link}", text)
  return text
```

Tokenisation of the dataset…

Tokenisation

Tokenisation concerns breaking up written speech (sentences) into individual words, sentences and/or phrases. During pre-processing stage each “token” is considered for inclusion in the dataset according to the pre-processing rules applied.

It's a new day in America.

[It, ', s, a, new, day, in, America, .]

Adding the tokenisation stage…

```
# stage 5 - tokenization
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize, word_tokenize
def text_to_tokens(text):
  tokens = []
  for sentance in sent_tokenize(text):
    for word in word_tokenize(sentance):
      tokens.append(word)
  return tokens
```

pre-trained English tokeniser

['next', 'i', 'am', 'buying',
'coca-cola', 'to', 'put',
'the', 'cocaine', 'back', 'in']

Adding the stopwords stage…

```
# stage 6 - stopwords
def tokens_without_stopwords(tokens):
  stopwords = ["to", "from", "and", "they", "them", "he", "she"]
  keep = []
  for token in tokens:
    if not token in stopwords:
      keep.append(token)
  return keep
```

where else can we find lists of stopwords?

Adding a lemmatisation stage…

- Many words have a similar meaning
- Words with a common meaning belong to the same lemma.

```
import nltk
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer
# stage 7 - lemmatisation
def apply_lemmatisation(tokens):
  keep = []
  lemmatizer = WordNetLemmatizer()
  for token in tokens:
    keep.append(lemmatizer.lemmatize(token))
  return keep
```

Putting it all together…

```
import pandas as pd

texts = ["Next I'm buying Coca-Cola to put the cocaine back in",
         "This ice-cream cost me $100",
         "Visit https://tinyurl.com/1291 for a prize.."]
df = pd.DataFrame()
df["text"] = texts

def preprocessing_pipeline(row):
    text = row["text"]
    text = text_with_contractions(text)
    text = text_lowercase(text)
    text = replace_markers(text)
    text = replace_http_links(text)
    tokens = text_to_tokens(text)
    tokens = tokens_without_stopwords(tokens)
    tokens = apply_lemmatisation(tokens)
    prepared_text = ' '.join(tokens)
    return prepared_text

df["prepared_text"] =  df.apply(preprocessing_pipeline, axis=1)
```

The resulting text…

index	text
0	Next I'm buying Coca-Cola to put the cocaine back in
1	This ice-cream cost me $100
2	Visit https://tinyurl.com/1291 for a prize..

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

	prepared_text
	next i am buying coca-cola put the cocaine back in
	this ice-cream cost me $ { price }
	visit $ { link } for a prize ..

In Summary

- Data preparation (filtering and merging) and data pre-processing (index compaction) are important stages of the text mining process.
- Filtering/merging assists the analysist in identifying the data that is of most interest and best suited to answering the research question.
- Pre-processing involves the creation of a set of procedures that transform the input text into a set of analysable data features.
- A pre-processing pipeline refers to a set of procedures that are applied sequentially to each data instance.
- The precise steps that are applied will depend on the problem domain (our focus is essentially on social media data).
- Next week we will see how this dataset can be explored further and organised into abstract topics of conversion.

End