



# Defoe: A Spark-based for Analysing Digital Historical Textual Data

Dr. Rosa Filgueira, EPCC,  
University of Edinburgh  
Email: [rosa.filgueira@ed.ac.uk](mailto:rosa.filgueira@ed.ac.uk)

# Context

## Working with

- Historians, Humanities and computational linguistics researchers
- Large digital collections been available for research

## Funding

- ATI-SE Data Engineering Programme
- Living with Machines (LwM)
- Text Data Mining (TDM) (staring next month)

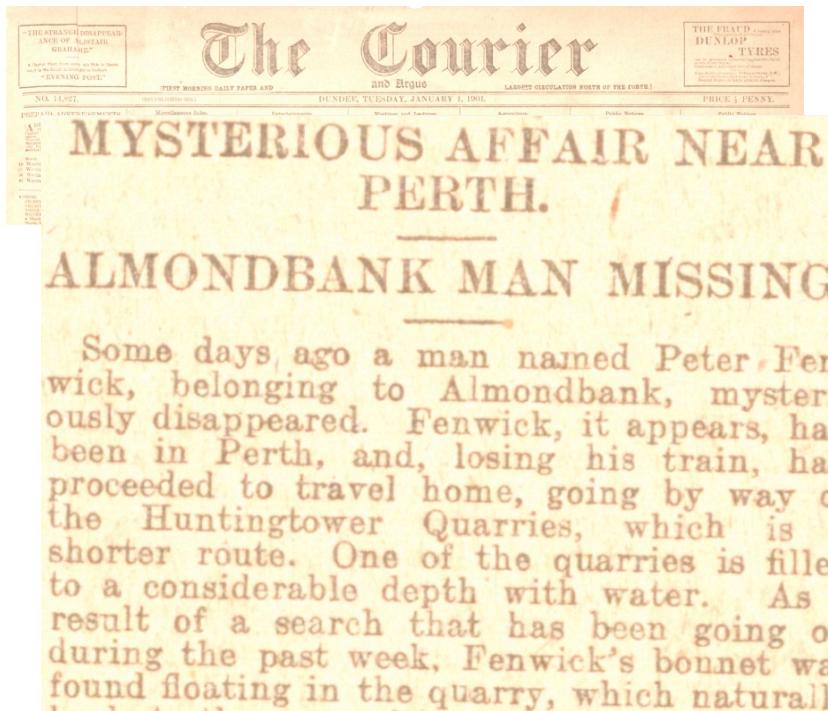
## Motivation – eScience for Historians & Humanities communities

- Hunger for large scale text mining facilities
- Limited capacity and/or skills to use:
  - HPC/Cloud environments
  - analytic frameworks to create applications

# Context

## Challenges

- Several large digital collections (semi-structured data)
- Different levels of quality of data – OCR
- Data with different physical representations and schemas



...

```
<text.title>
<pg pgref="5" clipref="1"
    pos="4069,3036,4949,3154"/>
<p>
    <wd pos="4069,3036,4949,3154">MYSTERIOUS AFFAIR
NEAR PERTH.</wd>
</p>
</text.title>
<text.cr>
<pg pgref="5" clipref="1"
    pos="4039,3191,4987,4235"/>
<p>
    <wd pos="4041,3192,4496,3241">ALMONDBANK</wd>
    <wd pos="4523,3200,4663,3246">MAN</wd>
    <wd pos="4696,3198,4976,3250">MISSING.</wd>
    <wd pos="4085,3290,4189,3323">Some</wd>
    <wd pos="4214,3290,4312,3329">days,</wd>
    ...
```

# *defoe*: new eScience toolbox for historical research

## Digital Collections



Object model



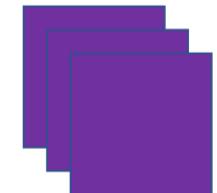
Query

1. Data ingestion
2. NLP preprocessing pipeline
3. Text mining query execution
4. Results gathering



Text analysis pipeline

Results



defoe  
visualization

```
Analyses of the Frequency, Taxonomy using NGrams for female emperors

This notebook is to take a look at what we are going to the track of each individual word series: by plotting the every article that has emperors in it is the number of the words, thoughts, engagement, errors, etc over time - going through each individual timeseries and then looking at the overall trends and patterns emerging over time, and certain changes trend over certain times.

In [1]: from __future__ import print_function
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

Functions
In [2]: def get_top_ngram(text, n=1):
    """Get top ngrams from text"""
    if type(text) == str:
        text = [text]
    ngrams = []
    for doc in text:
        doc_ngrams = list(nltk.ngrams(doc, n))
        ngrams.append(doc_ngrams)
    return ngrams
```

Jupyter Notebook

scalable

portable

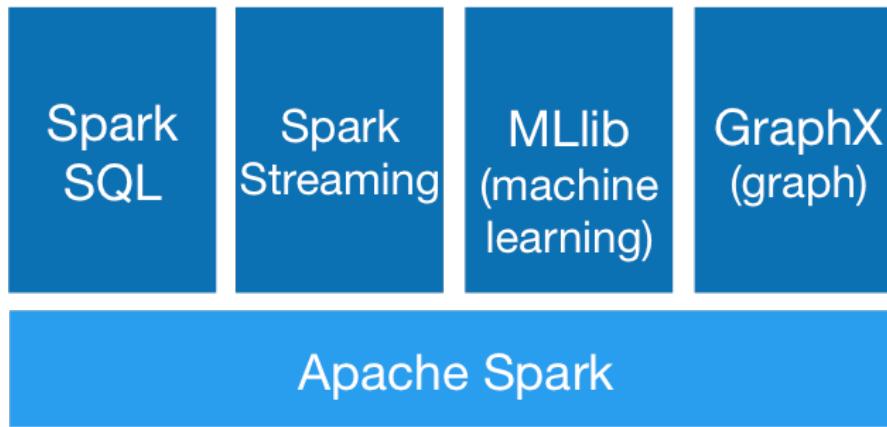
text mining queries

large dataset

<https://github.com/alan-turing-institute/defoe>

[https://github.com/alan-turing-institute/defoe\\_visualization](https://github.com/alan-turing-institute/defoe_visualization)

# Apache Spark



- Analytics engine for large-scale data processing
- High performance for batch and streaming data
- APIs  
Java, Scala, **Python** and R

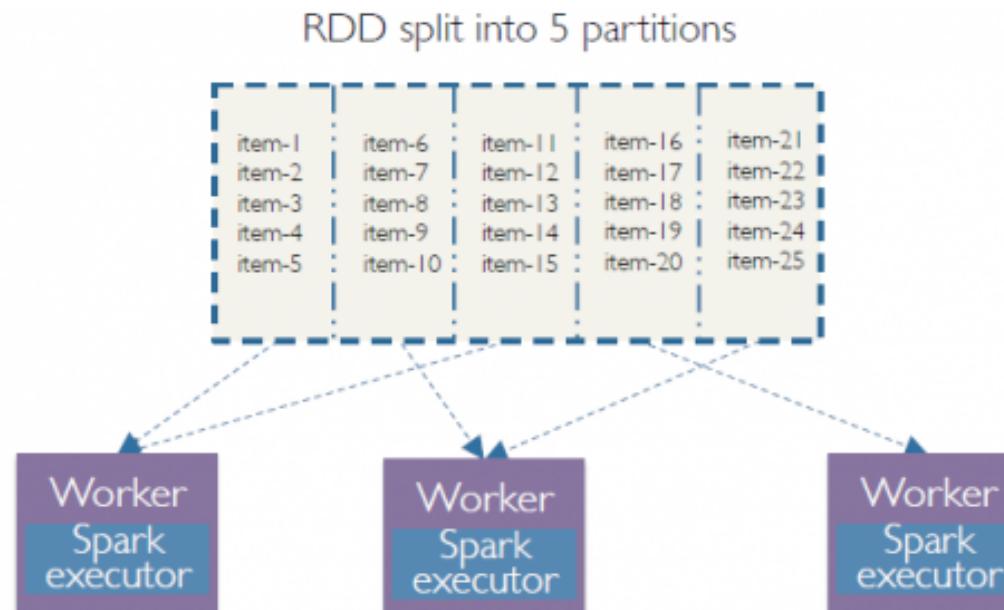
<https://spark.apache.org/>

<https://github.com/EPCCed/prace-spark-for-data-scientists>

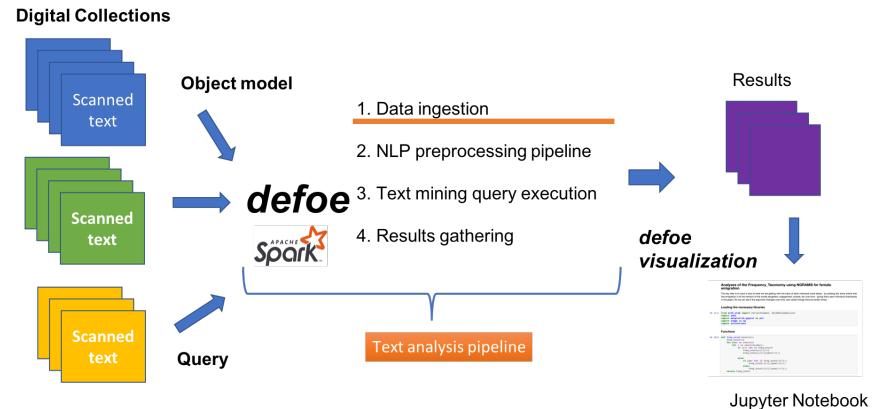
# Apache Spark

## Resilient Distributed Datasets (RDD)

- Represent data or transformations on data
- It is distributed collection of items – partitions
- Read-only → they are immutable
- Enables operations to be performed in parallel



# Data Ingestion



Support for three physical representations:

- 1) one XML document per issue
- 2) one XML document with search results including several articles
- 3) one XML metadata document and a XML per page

Object models -- loading data into RDD:

PAPER (physical rep num1)

NZPP (physical rep num2)

ALTO (physical rep num3)

FMP (physical rep num3)

# PAPER object model (British Library Newspapers)

RDDs



0000164- The Courier and Argus  
0000187- The Bath Chronicle  
0000195- Archer Bath Chronicle  
0000321- The Nottingham Evening Post  
0000452- Edinburgh Evening News

**Class Issue** → Representation of an issue (XML document)  
Each XML holds articles that belong to the same issue

issue

filename  
issue\_tree  
issue\_id  
date  
page\_count  
day\_of\_the\_week

attributes

article list

article  
1

article  
2

article  
3

article  
4

**Class Article** → Representation of an article

article

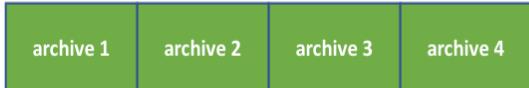
attributes

article\_tree  
filename  
quality  
title  
preamble  
content  
article\_id  
page\_ids

words = content + title + preamble

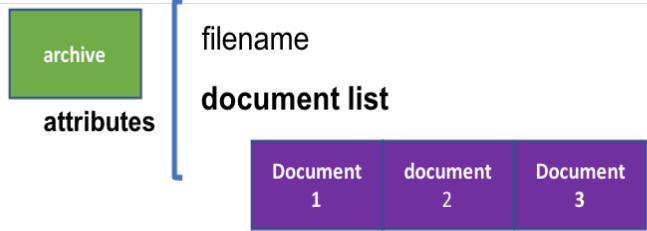
# ALTO object model (British Library books)

RDDs

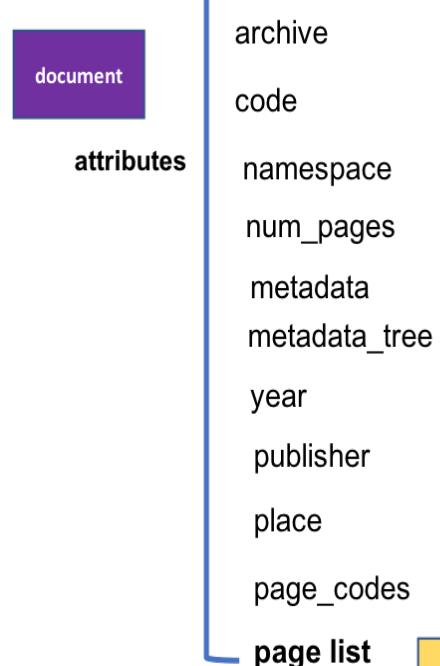


1510\_1699/000001143\_0\_1-20pgs\_\_560409\_dat.zip  
1510\_1699/000000874\_0\_1-22pgs\_\_570785\_dat.zip  
1510\_1699/000051983\_0\_1-92pgs\_\_568584\_dat.zip  
1510\_1699/000987728\_0\_1-92pgs\_\_567840\_dat.zip

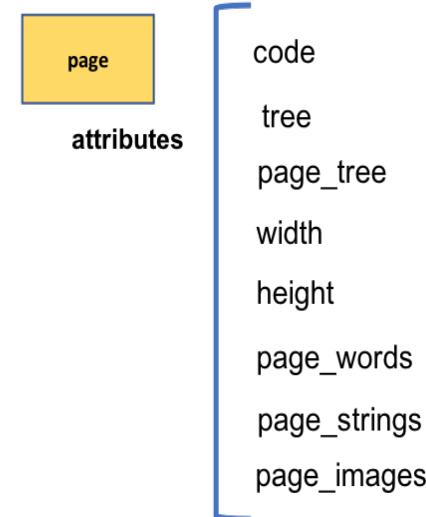
**Class Archive** → Representation of a zipped archive



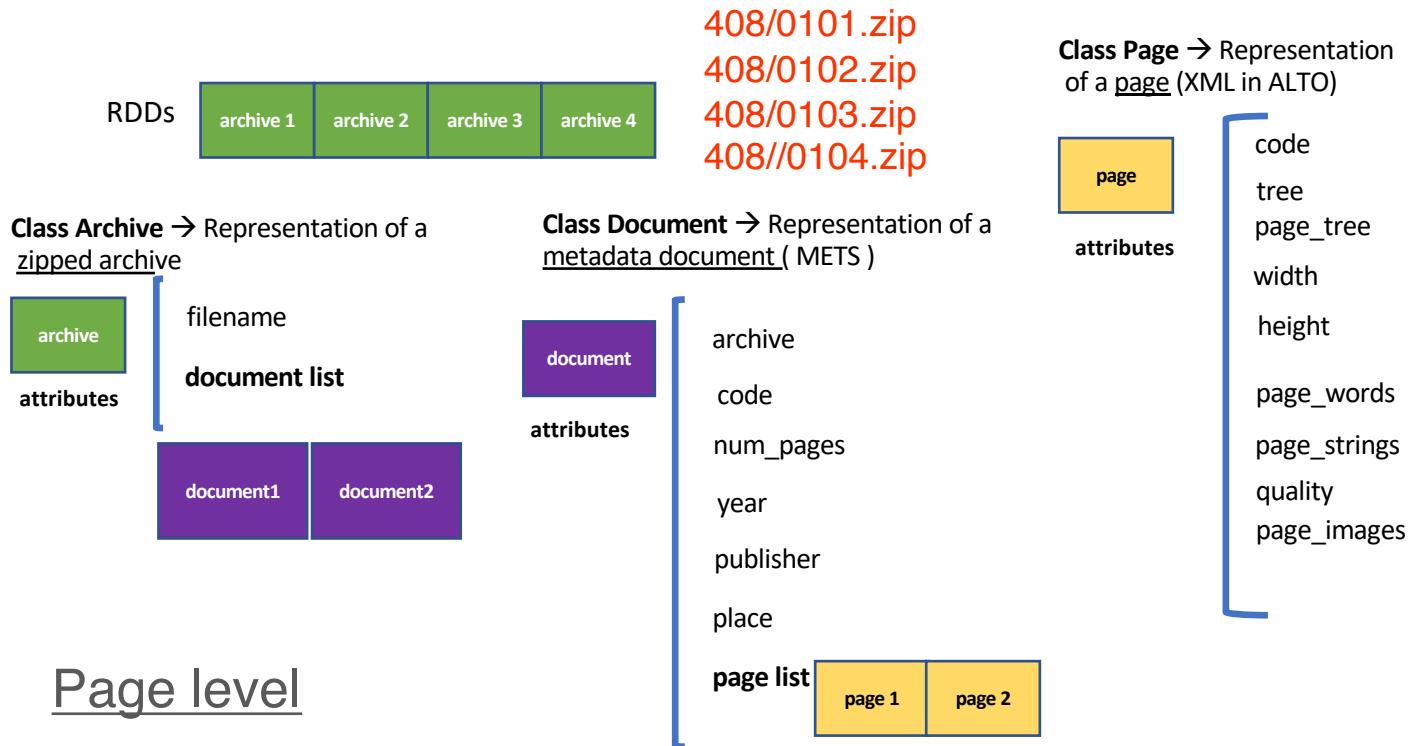
**Class Document** → Representation of a metadata document ( XML in METS/MODS) and an ALTO directory



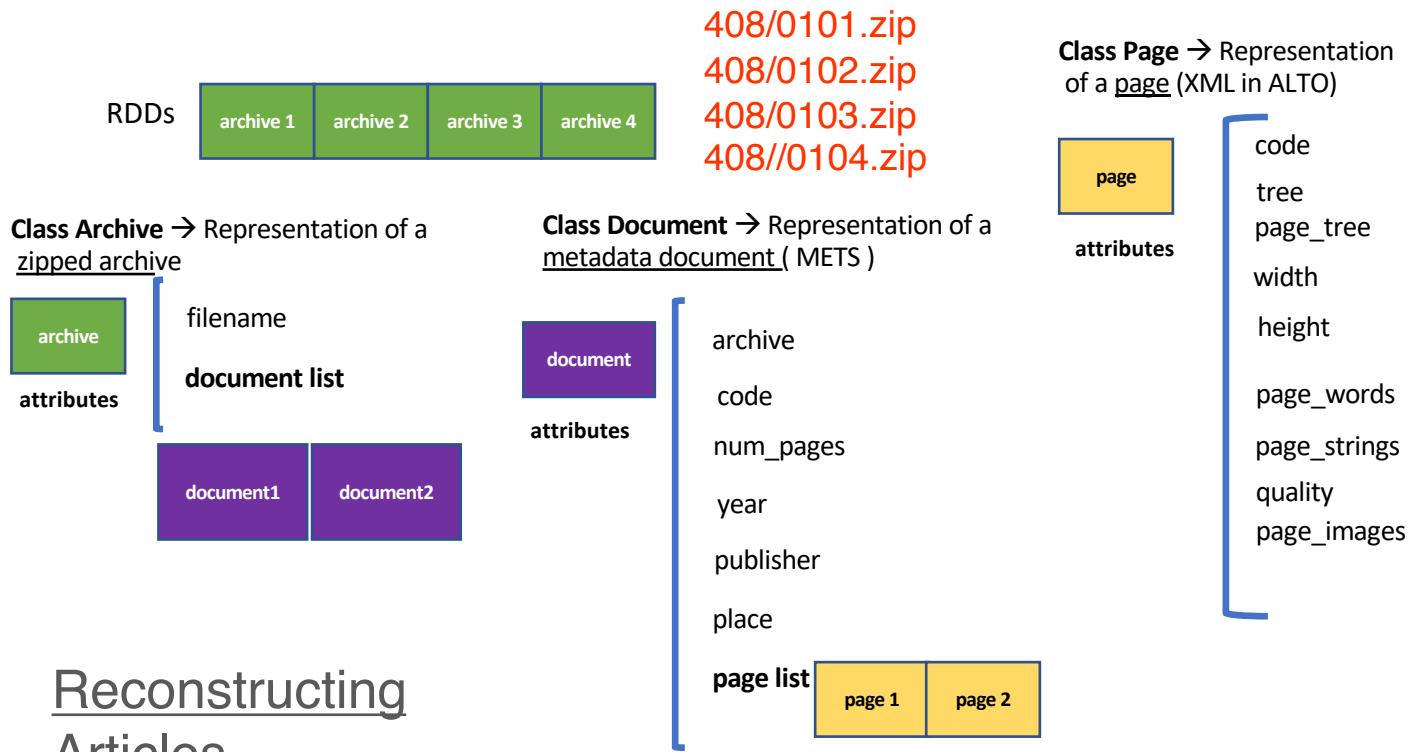
**Class Page** → Representation of a page (XML in ALTO).



# FMP object model (Find My Past Newspapers)



# FMP object model (Find My Past Newspapers)



## Reconstructing Articles

### METS

logical structure - how many articles has an issue, and each article id (*art0001, art0002, art0003*)

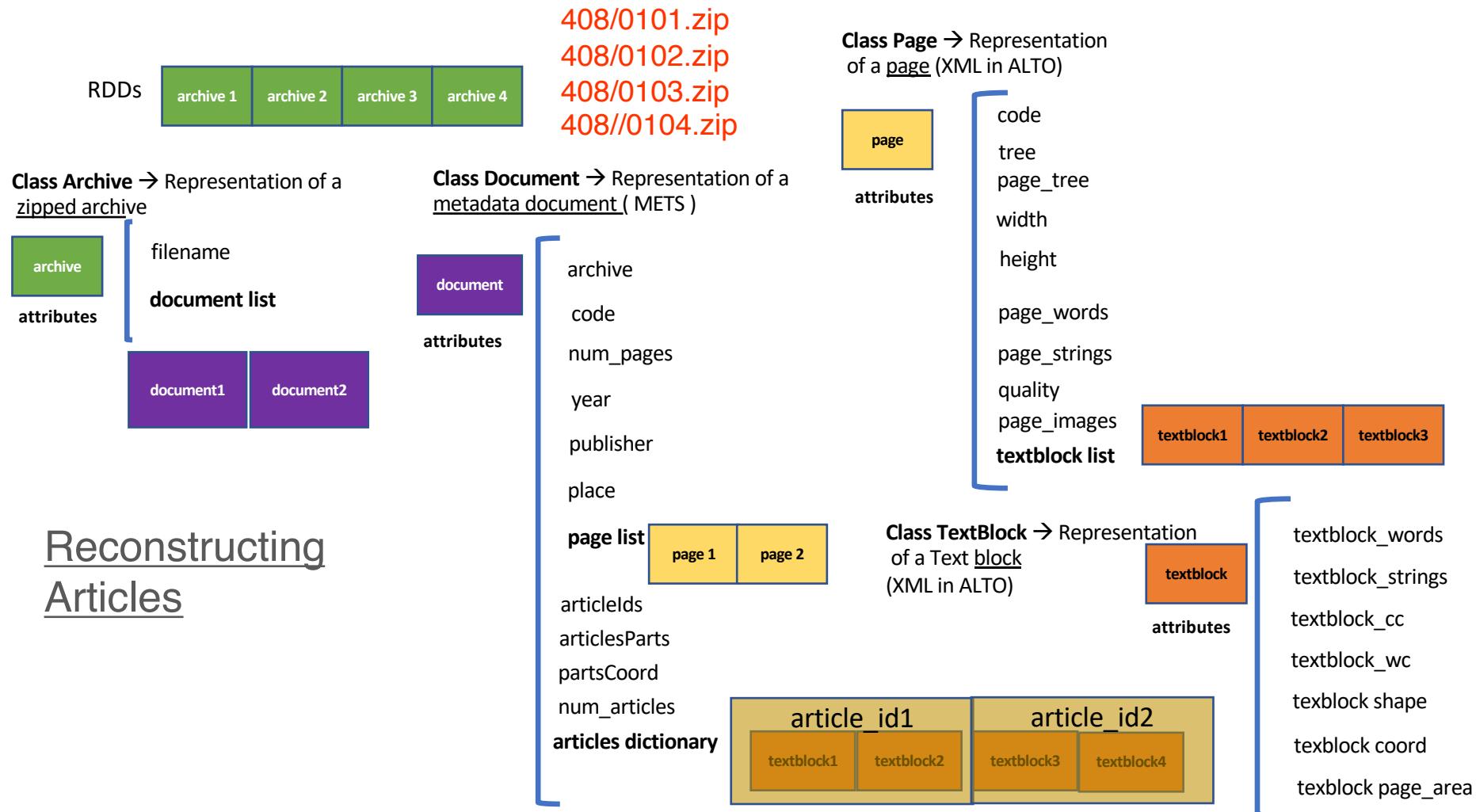
structural link - how many text blocks/parts has each article, and each text blocks id (*pa0001001, pa0001002, .*)

physical structure - coordinates of each text block (*COORDS="1220,5,2893,221"*)

### ALTO

Text blocks/parts ids that has each page and their content

# FMP object model (Find My Past Newspapers)



## Reconstructing Articles

(It also works with unzipped archives)

# Digital Collections

Important

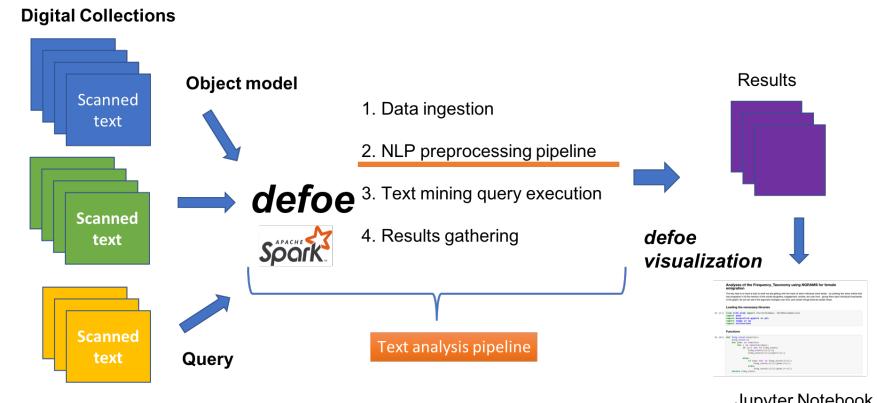
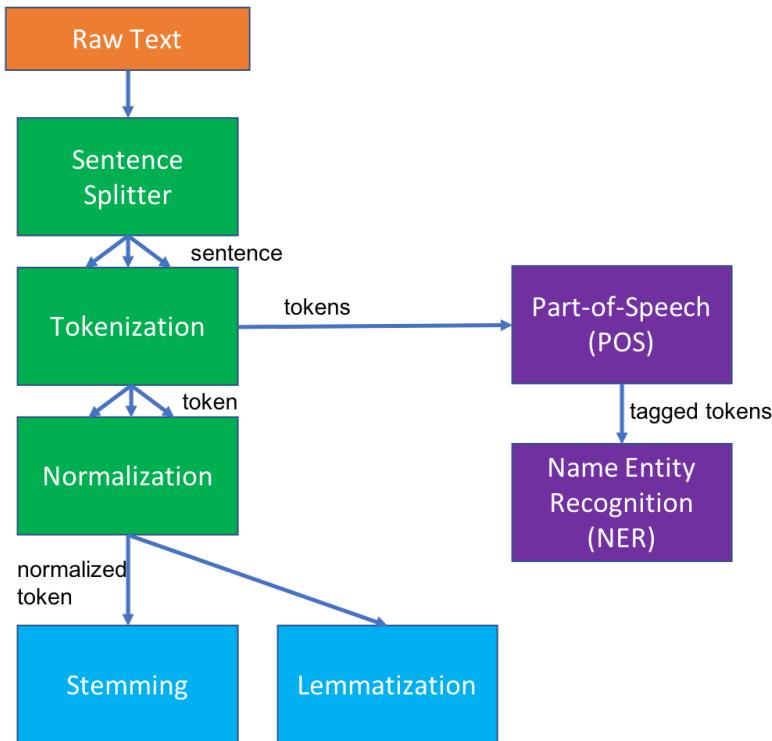


ATI-  
SE

LwM

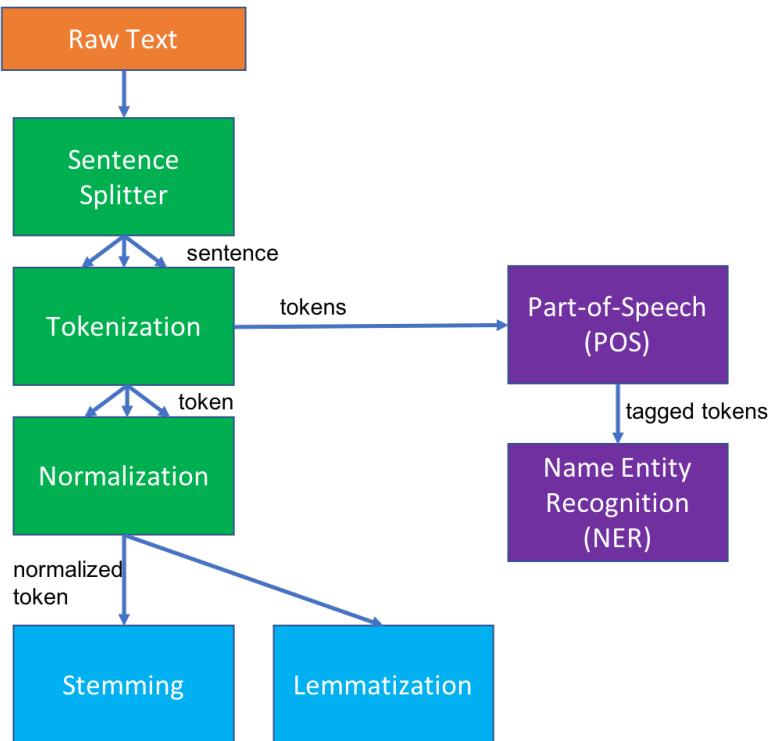
Dataset	Period	Structure	XML Schema	Space	Model
British Library Books (BLB)	1510-1899	ZIP per book - XML metadata - XML per page	METS and ALTO schemas	~220GB	<u>ALTO</u>
British Library Newspapers (BLN)	1714-1950	XML per issue	GALEN Schema	~1TB	<u>PAPERS</u>
Times Digital Archive (TDA)	1785-2009	XML per issue	GALEN Schema	~324GB	<u>PAPERS</u>
Papers Past New Zealand and Pacific newspapers (NZPP)	1839-1863	XML per 22 articles	XML from a search via an API	~4GB	<u>NZPP</u>
FindMyPast (FPM)	1752 - 1957	-XML metadata -XML per newspaper page	METS and ALTO schemas	~20TB	<u>FMP</u>

# NLP Preprocessing



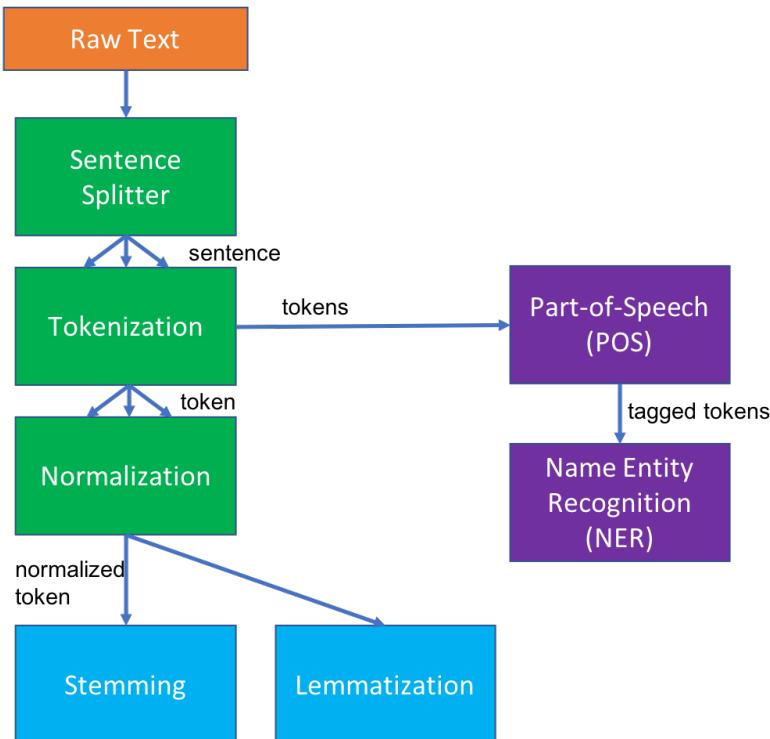
Jupyter Notebook

# NLP Preprocessing



1. Divide the text in sentences
2. Each word of a sentence is tokenised
3. Each token is normalised
- 4.a) Stemming: reduces words to their roots
- 4.b) Lemmatisation: reduces words to a common base
5. Tag each token  
(e.g. nouns, verbs, adjectives, etc.)
6. Classify tagged tokens  
(e.g. names of persons, locations, etc)

# NLP Preprocessing



## Introducing NLP to our queries: NLTK and spaCy

Sentence: "And devoted some time to social work in London."

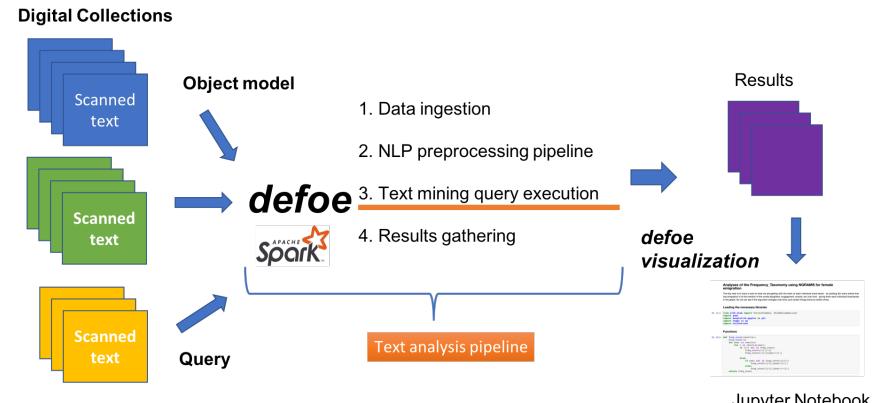
| spaCy preprocessing |

Word	Normaliz.	Lemma	PoS	Tag	NER
And	and	and	CCONJ	CC	
devoted	devoted	devote	VERB	VBN	
some	some	some	DET	DT	
time	time	time	NOUN	NN	
to	to	to	ADP	IN	
social	social	social	ADJ	JJ	
work	work	work	NOUN	NN	
in	in	in	ADP	IN	
London	london	London	PROPN	NNP	GPE
.	.	.	PUNCT	.	.

| NLTK preprocessing |

Word	Normaliz.	Lemma	Stem	PoSTag	NER
And	and	and	and	CC	(S And/CC)
devoted	devoted	devoted	devot	VBN	(S devoted/VBN)
some	some	some	some	DT	(S some/DT)
time	time	time	time	NN	(S (NP time/NN) )
to	to	to	to	TO	(S to/TO)
social	social	social	social	JJ	(S social/JJ)
work	work	work	work	NN	(S (NP work/NN) )
in	in	in	in	IN	(S in/IN)
London	london	london	london	NNP	(S (GPE London/NNP) )
.	.	.	.	.	(S ./.)

# Text Mining Queries



Jupyter Notebook

## ALTO Model

- [colocates\\_by\\_year.py](#)
- [keyword\\_by\\_word.py](#)
- [keyword\\_by\\_year.py](#)
- [keyword\\_concordance\\_by\\_word.py](#)
- [keyword\\_concordance\\_by\\_year.py](#)
- [normalize.py](#)
- [ocr\\_quality\\_by\\_year.py](#)
- [total\\_documents.py](#)
- [total\\_pages.py](#)
- [total\\_words.py](#)

## PAPERS Model

- [colocates\\_by\\_year.py](#)
- [keysentence\\_by\\_year.py](#)
- [keyword\\_by\\_year.py](#)
- [keyword\\_concordance\\_by\\_date.py](#)
- [keywords\\_by\\_year.py](#)
- [lda\\_topics.py](#)
- [normalize.py](#)
- [ocr\\_quality\\_by\\_year.py](#)
- [target\\_and\\_keywords\\_by\\_year.py](#)
- [target\\_and\\_keywords\\_count\\_by\\_ye...](#)
- [target\\_concordance\\_collocation\\_b...](#)
- [total\\_articles.py](#)
- [total\\_issues.py](#)
- [total\\_words.py](#)
- [unique\\_words.py](#)

## NZPP Model

- [keyword\\_by\\_year.py](#)
- [keyword\\_concordance\\_by\\_date.py](#)
- [normalize.py](#)
- [total\\_articles.py](#)
- [total\\_words.py](#)

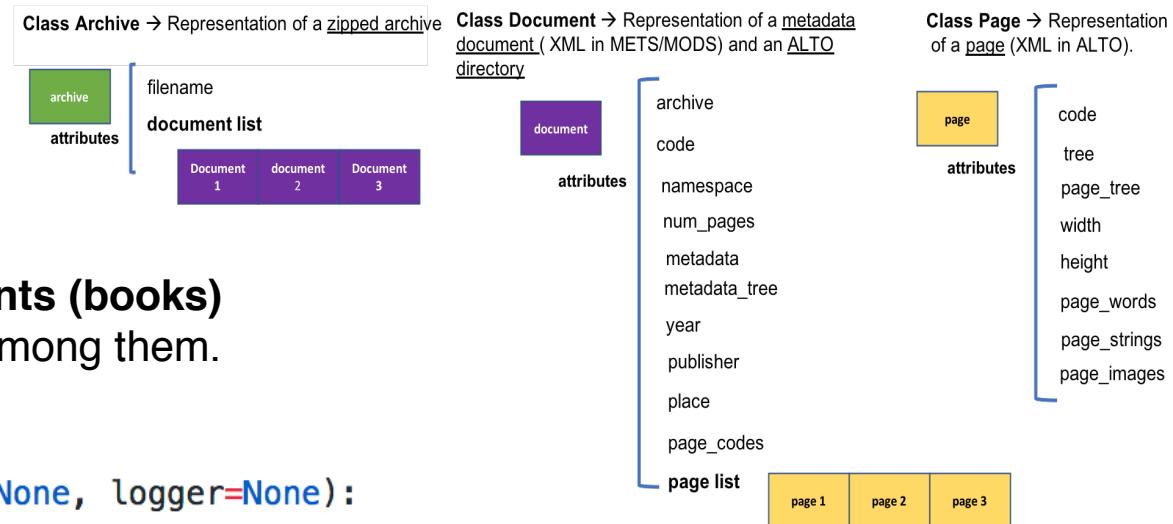
## FMP Model

- [keyword\\_metadata\\_by\\_word.py](#)
- [keyword\\_segmentation.py](#)
- [normalize.py](#)
- [target\\_segmentation.py](#)
- [total\\_articles.py](#)
- [total\\_documents.py](#)

# Text Mining Queries



## ALTO object model



### ***total\_words*** query:

- Iterates through archives
- Count **total number of documents (books)** and **total number of words** among them.

```
def do_query(archives, config_file=None, logger=None):  
    # [archive, archive, ...]  
    documents = archives.flatMap(lambda archive: list(archive))  
    # [num_words, num_words, ...]  
    num_words = documents.map(lambda document: len(list(document.words())))  
    result = [documents.count(), num_words.reduce(add)]  
    return {"num_documents": result[0],  
            "num_words": result[1]}
```

### Sample results

Query over British Library Books

{`num_documents: 63701, num_words: 6866559285`}

# Text Mining Queries

## **total\_words** query:

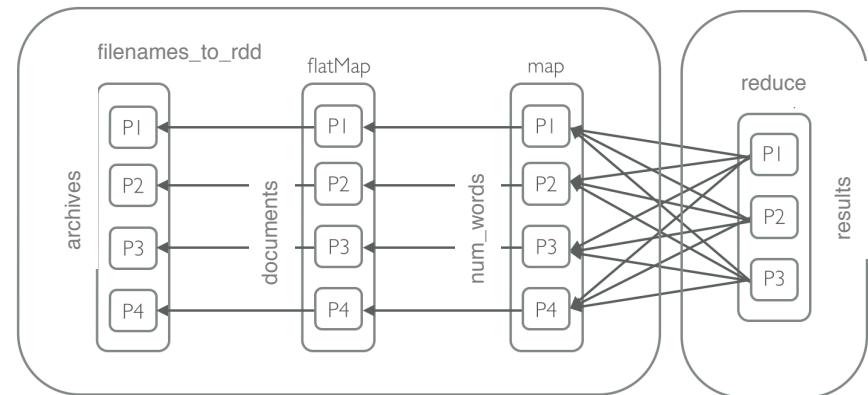
- Iterates through archives
- Count **total number of documents (books)** and **total number of words** among them.

```
def do_query(archives, config_file=None, logger=None):  
  
    # [archive, archive, ...]  
    documents = archives.flatMap(lambda archive: list(archive))  
    # [num_words, num_words, ...]  
    num_words = documents.map(lambda document: len(list(document.words())))  
    result = [documents.count(), num_words.reduce(add)]  
    return {"num_documents": result[0],  
            "num_words": result[1]}
```

## Sample results

Query over British Library Books

{num\_documents: 63701, num\_words: 6866559285}



*Direct Acyclic Graph of RDD objects.  
Each RDD points to the parent.*

# Installing defoe

## Install Spark and Java 8

```
sudo apt install openjdk-8-jdk
wget http://apache.mirror.anlx.net/spark/spark-2.4.4/spark-2.4.4-bin-hadoop2.7.tgz
tar xvf spark-2.4.2-bin-hadoop2.7.tgz
```

## Conda environment (Python 2.7)

```
git clone https://github.com/alan-turing-institute/defoe.git
conda create -n LwMpy27 python=2.7 anaconda
conda activate LwMpy27
conda update -n base -c defaults conda
```

## Install dependencies

```
./requirements.sh
pip install Pillow==4.0.0
```

## Install NLTK

```
    |   python
    |   |   import nltk
    |   |   nltk.download('wordnet')
```

# Running defoe

Submit the source code to Spark along with information about your query:

```
spark-submit --py-files defoe.zip defoe/run_query.py <DATA_FILE> <MODEL_NAME> <QUERY_NAME> <QUERY_CONFIG
```

1. Data file: URLs or file paths

0000164- The Courier and Argus

0000187- The Bath Chronicle

Model Name: object model to be used

papers

Query name: name of the python query module

defoe.papers.queries.articles\_containing\_words

Query configuration: (optional) - query-specific configuration file

queries/emigration.yml

**Complete information:**

<https://github.com/alan-turing-institute/defoe/blob/master/docs/run-queries.md>

# Case Studies -Stranger Danger

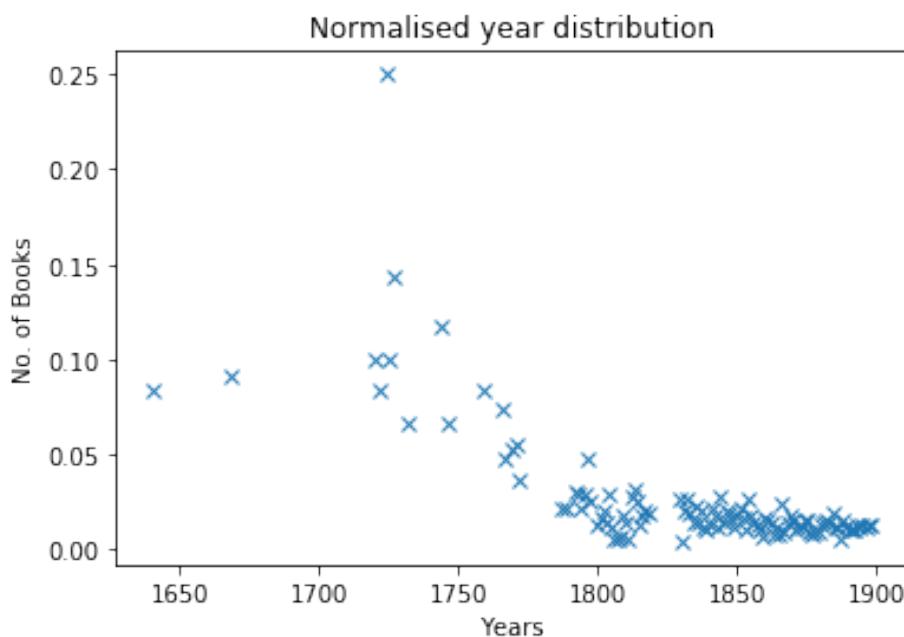
*Detect the origin of ‘Stranger Danger’ expression*

- ***colocates\_by\_year*** query:  
Searches for sentences where the words “stranger” and “danger”  
**(matching criteria)** appear within the same sentence.  
Results are grouped by the publication dates.
- ***Normalise*** query:  
Counts total number of documents, pages and words per year
- Jupyter Notebook (\*)
  - Compare results - plot them by year
  - Normalise the results
  - Sentiment analyses
  - Visualise which words appear more often near the phrase



[https://github.com/alan-turing-institute/defoe\\_visualization/blob/master/Stranger\\_Danger/Stranger\\_Danger.ipynb](https://github.com/alan-turing-institute/defoe_visualization/blob/master/Stranger_Danger/Stranger_Danger.ipynb)

# Case Studies -Stranger Danger



Visualization of how the 'Stranger Danger' terms are affected by the way that the number of books were published

Getting the first book, which both terms together:

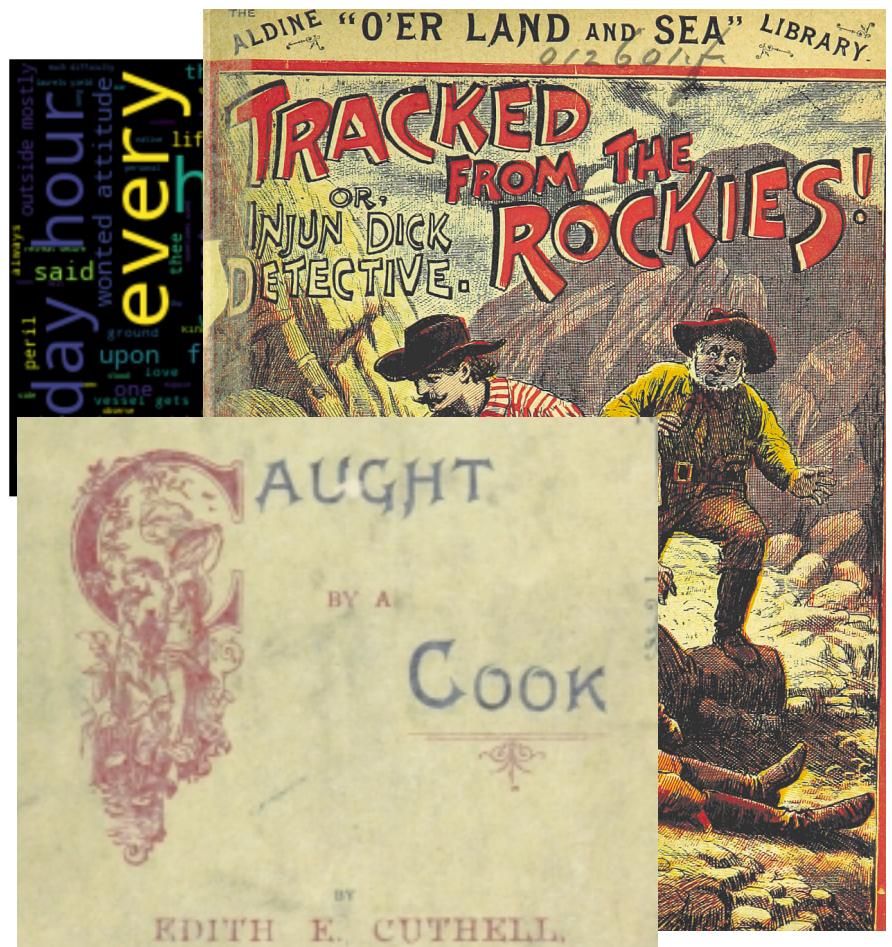
"Caught by a Cook (1895)", Page: 105.

Sentence: "who knew most of the villagers by sight perceived a stranger danger."

Getting the total number of matching sentences, and the book which has the max. num of them:

Total sentences found is 1038

The book "Aldine "O'er Land and Sea" (1890) , has the max number of sentences: 29

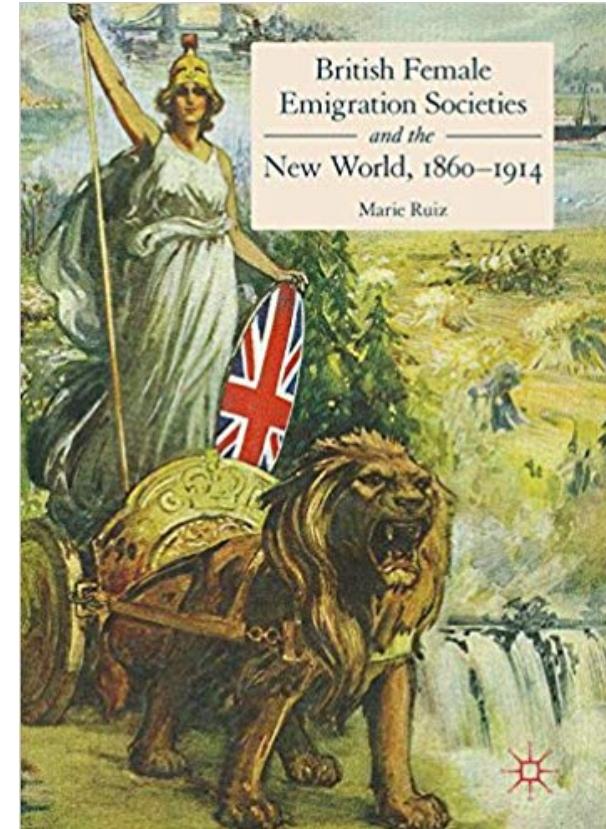


# Case Studies – Female Emigration

*Mine the TDA and BLN archives for attitudes towards female emigration from Great Britain to the ‘Colonies’ and North America from 1850 to 1914*

Normalised frequencies of female emigration societies:  
***keysentence\_by\_year*** & ***normalize*** queries

Normalised frequencies of taxonomy terms relating to female emigration:  
***target\_and\_keywords\_count\_by\_year*** & ***normalize*** queries



# Case Studies – Female Emigration

## Taxonomy terms

emigration  
bookkeeping  
Colony  
Colonies  
Colonial  
daughters  
engagement  
empire  
British empire  
failure  
female labourer  
female welfare  
feminine  
genteel  
good character  
governess  
guardian  
guardianship  
happiness  
hardship

### Female taxonomy terms co-located with “emigration”

#### **1850:**

- [teacher, 23]
- [loan, 138]
- [mother, 102]
- [happiness, 15]
- [genteel, 6]
- [marriage, 57]
- [respectable, 177]

#### **1851:**

- [indecent, 2]
- [superintendence, 13]
- [loan, 130]
- [suitable, 54]
- [colony, 768]
- [bookkeeping, 1]
- [success, 78]

Using TDA corpus:  
Only newspapers  
From 1850 to 1914.

### Frequencies of female societies

#### **1850:**

- [governess benevolent institution, 1]
- [emigration agent, 17]
- [emigration scheme, 4]
- [colonial land and emigration commission, 2]
- [family colonization loan society, 2]

### Total number of issues, articles and words per year

- 1850: [313, 22288, 33449528]  
1851: [313, 21247, 33843690]  
1852: [314, 19586, 33903669]

## Societies

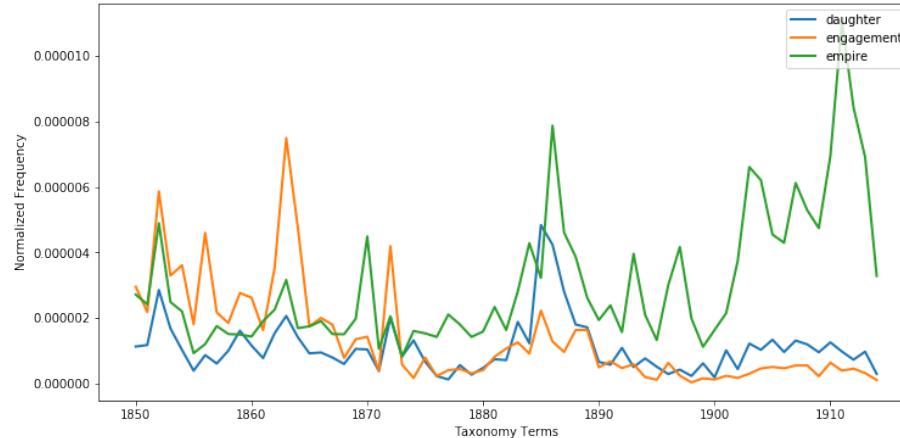
South African Colonisation Society  
Juvenile Emigration Society  
Child Emigration Society  
Church Emigration Society  
Self Help Emigration Society  
East End Emigration Fund  
Church Army Emigration Department  
Carlton Emigration Society to Canada

# Case Studies – Female Emigration

## Taxonomy terms

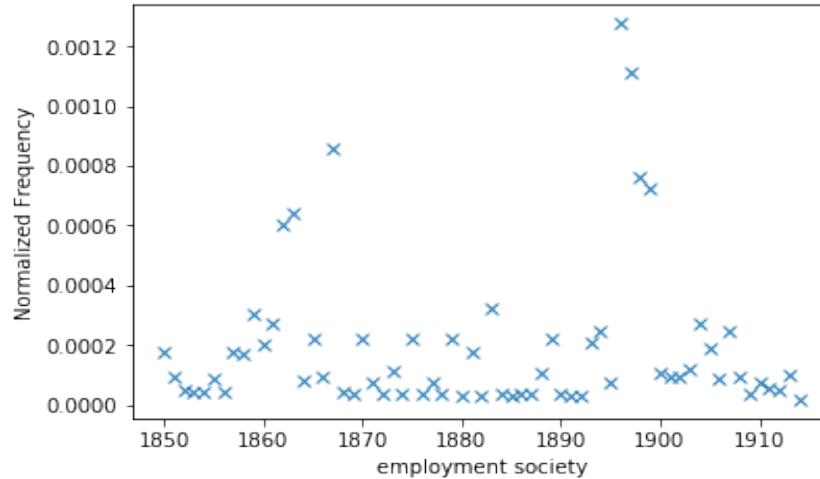
emigration  
bookkeeping  
Colony  
Colonies  
Colonial  
daughters  
engagement  
empire  
British empire  
failure  
female labourer  
female welfare  
feminine  
genteel  
good character  
governess  
guardian  
guardianship  
happiness  
hardship

Normalised N-grams of female taxonomy terms co-located with “emigration”



Using TDA corpus:  
Only newspapers  
From 1850 to 1914.

Normalised frequencies of female societies



## Societies

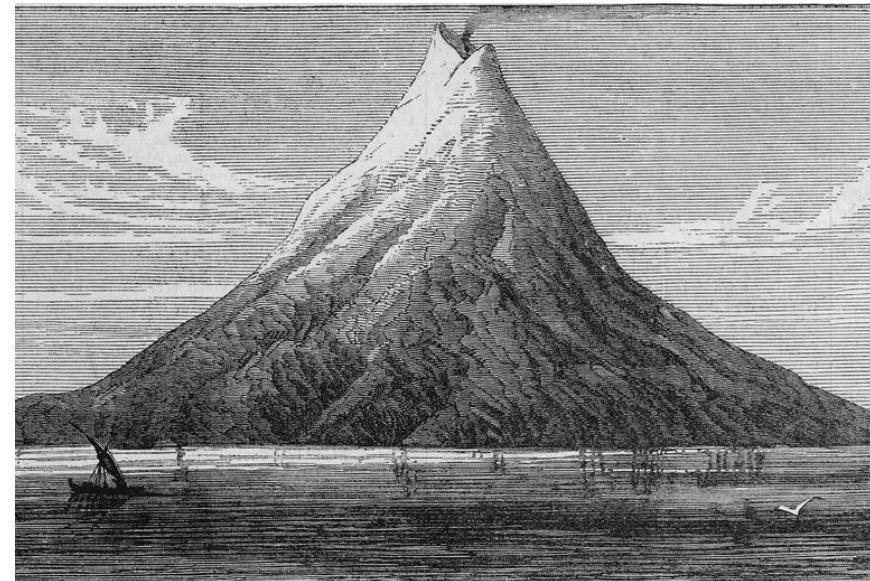
South African Colonisation Society  
Juvenile Emigration Society  
Child Emigration Society  
Church Emigration Society  
Self Help Emigration Society  
East End Emigration Fund  
Church Army Emigration Department  
Carlton Emigration Society to Canada

# Case Studies -Eruption of Krakatoa Volcano in 1883

*Track copying and transmission of text*

Krakatoa (Krakatau in Indonesian) erupted over 26-27th August 1883 and was one of the most spectacular volcanic eruptions in contemporary times.

*Identify this eruption using  
BLN, TDA and NZPP papers from late 1883*



**keyword\_and\_concordance\_by\_date** query: searches for occurrences of “krakatoa” and “krakatua” and returns information on each matching article.

[https://github.com/alan-turing-institute/defoe\\_visualization/tree/master/Krakatoa\\_1883](https://github.com/alan-turing-institute/defoe_visualization/tree/master/Krakatoa_1883)

# Case Studies -Eruption of Krakatoa Volcano in 1883

1	date	newspaper title	search term	sequence	text
2	1883-05-24	0000453- The Evening Telegraph	krakatoa	1	VOLCANIC ERUPTION. Rata via, Thursday. A violent eruption ha* be
3	1883-05-25	0000327- The Derby Daily Telegraph	krakatoa	1	VOLCANIC ERUPTION IN THE STRAITS OF SUNDA. A Reuter's tele
4	1883-08-27	0000321- Nottingham Evening Post	krakatoa	1	ALARMING VOLCANIC DISTURBANCES. DISTURBANCES. A Calami
5	1883-08-27	0000325- The Citizen Gloucester	krakatoa	1	TERRIBLE VOLCANIC DISTURBANCE IN THE WEST INDIES : A VILL
6	1883-08-28	0000321- Nottingham Evening Post	krakatoa	1	THE VOLCANIC ERUPTION. r • -Manager. 2473 c " OFFICII,"^s"~ > .
7	1883-08-28	0000452- Edinburgh Evening News	krakatoa	1	GREAT VOLCANIC ERUPTION IN THE EAST INDIES. A Keuter's tele
8	1883-08-28	0000325- The Citizen Gloucester	krakatoa	1	THE VOLCANIC DISTURBANCES IN THE EAST INDIES. Batavia, Aug
9	1883-08-28	0000327- The Derby Daily Telegraph	krakatoa	1	SUMMARY. ®be Her bp ®axlg ®eUjraplj DERBY, August 28, 1883. Th
10	1883-08-29	0000327- The Derby Daily Telegraph	krakatoa	1	LATEST NEWS. "Telegraph" Officii, 1.0 P.M. MINISTERIAL CRISIS IN
11	1883-08-29	0000325- The Citizen Gloucester	krakatoa	1	THE VOLCANIC DISTURBANCES IN JAVA: A TOWN DESTROYED BY
12	1883-08-29	0000321- Nottingham Evening Post	krakatoa	1	A TOWN COMPLETELY DESTROYED. S^OIALjEDITION.! POST" OFF

~50 newspaper articles from all over England  
Rich data set where we can track copying and transmission of text

# Case Studies - Zooniverse and *Industrial accidents*

**Zooniverse-** Largest and most popular citizen science projects

The screenshot shows the Zooniverse website interface. At the top, there's a black navigation bar with links for 'PROJECTS', 'ABOUT', 'GET INVOLVED', 'TALK', 'BUILD A PROJECT', 'NEWS', 'NOTIFICATIONS', 'MESSAGES', and 'ROSA FILIGRANA'. Below this is a teal header bar featuring the 'LIVING WITH MACHINES' logo (a white circle with the text 'LIVING WITH MACHINES') and the project title 'Living with Machines'. To the right of the title are links for 'ABOUT', 'CLASSIFY', 'TALK', 'COLLECT', and 'RECENT'. The main content area has a dark background with green radial patterns. It displays the question 'How did the Industrial Revolution impact ordinary lives?' in white text. Below this is a white rectangular button with the text 'Learn more' in blue.

Victorian newspapers (FMP newspapers) didn't talk about 'industrial accidents', but they are full of accounts of workplace injuries linked to machines.

<https://www.zooniverse.org/projects/bldigital/living-with-machines>

# Case Studies - Zooniverse and *Industrial accidents*

The screenshot shows a historical newspaper clipping from The Hull Packet, dated 1841-03-05. The main text discusses a shipwreck, mentioning the Apollo, Sadler, which struck on the Hasbro Sand yesterday and has gone to pieces. It also mentions other ships like Jeune Clemence, Smit, and Eugene Hansen. Below the main text is a table of subject metadata:

SUBJECT METADATA	X
82	
attribution	Image © THE BRITISH LIBRARY BOARD. ALL RIGHTS RESERVED.
newspaper date	1841-03-05
newspaper place	Hull, Humberside, England
newspaper title	The Hull Packet.

**ANTWERP, Feb. 24—Arrived :** Jeune Clemence, Smit, from Hull. 26—Eugene, Hansen, from Hull.—23—Sailed : Diana, Nahmens ; Ludd, Grass ; Peace, Brecon ; August, Frericks ; for Hull. 25—Kezia, Spencer ; Zephir, Witteveen ; for Hull.

**ANTWERP, Feb. 24—**The Belgian barque Zephir, Witteveen, for Hull, left the docks yesterday, and made sail, but was thrown against the Kattendyk, and grounded.—The Princess Victoria sustained an accident in the river, on Sunday, and was thereby prevented putting to sea.—27—The Zephir, Witteren, hence for Hull, which was on shore near this, got afloat on the evening's flood, and has gone down the river, apparently without damage.

## TASK

## TUTORIAL

Does this article mention a specific industrial or workplace accident?

There's an industrial or workplace accident

There's a transport (train, tram, etc) accident

There's some other kind of accident

This is an ad, headline, etc, or an article where no specific accident is mentioned

There's something wrong and I can't complete the task

NEED SOME HELP WITH THIS TASK?

Victorian newspapers (FMP newspapers) didn't talk about 'industrial accidents', but they are full of accounts of workplace injuries linked to machines.

# Case Studies - Zooniverse and *Industrial accidents*

## Defoe – Filtering and Cropping article's images – FMP newspapers

**Target\_segmentation** : Crops articles textblocks that at least have either **MACHINE** or **MACHINERY** (**target words**), **AND** any of the **keywords** (*accident, crush, smash, etc.*) listed in the [lexicon](#).

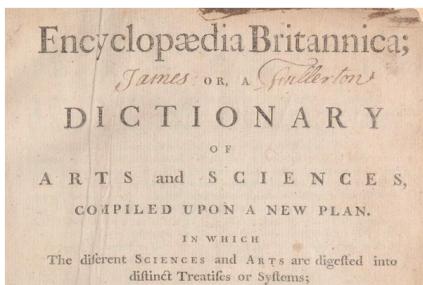
```
spark-submit --py-files defoe.zip defoe/run_query.py data.txt  
fmp defoe.fmp.queries.target_segmentation queries/accident.yml
```

Results: Metadata file (article\_id, publication, year)  
and **Images files (upload to Zooniverse)**

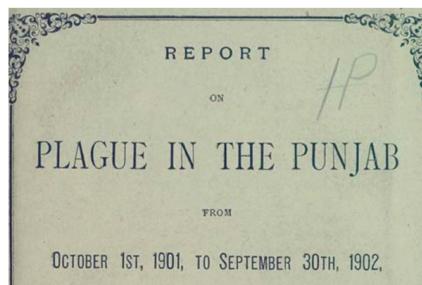
# National Library Scottish (New!)

## Digitised collections

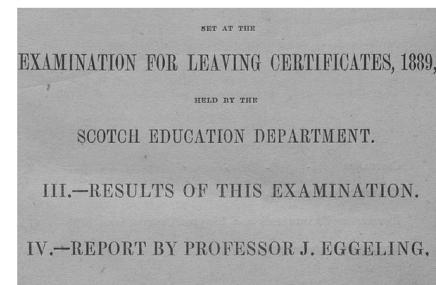
Download the ALTO, METS, image and plain text files for our digitised collections.



Encyclopædia Britannica, 1768-1860



A Medical History of British India



Scottish School Exam Papers, 1888-1963

# National Library Scottish (New!)



HOME ABOUT DATA TOOLS CONTACT

## Encyclopaedia Britannica, 1768–1860



Original OCR: no  
clean-up



155,388 ALTO  
XML files at page  
level



155,388 image  
files



METS metadata  
files at item level



19,257,785 lines  
and 166,729,009  
words



Covers years  
1768-1860

<https://github.com/alan-turing-institute/defoe/tree/master/defoe/nls>

Physical representation: one XML METS per volume and a XML ALTO per page

# Computing Environments – defoe - portable

## Cray Urika-GX system:

- High-performance analytics cluster
- **Apache Spark**, Apache Hadoop, Jupyter Notebooks, etc.
- 12 computing nodes: 36 CPUs and 256GB
- 60TB of storage – HDFS & Lustre

## EDDIE :

- University of Edinburgh HPC cluster.
- 7000 Intel cores with up to 3TB per compute node

## Microsoft Azure

- Alan Turing Allocation
- HDInsight: Cloud-based service for big data analytics -- it includes Spark
- Apache Livy: Submitting (Spark) Jobs from Anywhere

# Conclusions

- New digital toolbox for extracting knowledge from historical data.
- Enables running text analyses across large collections in parallel.
- Offers a rich set of text mining queries.
- Includes NLP prepossessing techniques to mitigate OCR errors.
- Tested portability on different computing environments and digital collections.

*"All this work provides the means to search across large scale datasets and to return results for further analysis and interpretation by historians."*



# Defoe: A Spark-based for Analysing Digital Historical Textual Data

Dr. Rosa Filgueira, EPCC,  
University of Edinburgh  
Email: rosa.filgueira@ed.ac.uk