# Counting Words and Detecting Themes with Python: Corpus Linguistics and Topic Modelling Approaches

Yin Yin Lu, Oxford Internet Institute

**University of Tartu Digital Methods Summer School**

22-23 August 2018

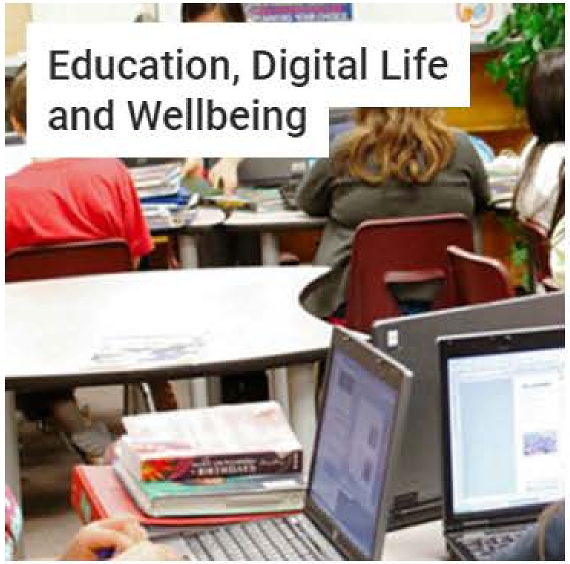
European Union
European Regional
Development Fund

Investing
in your future

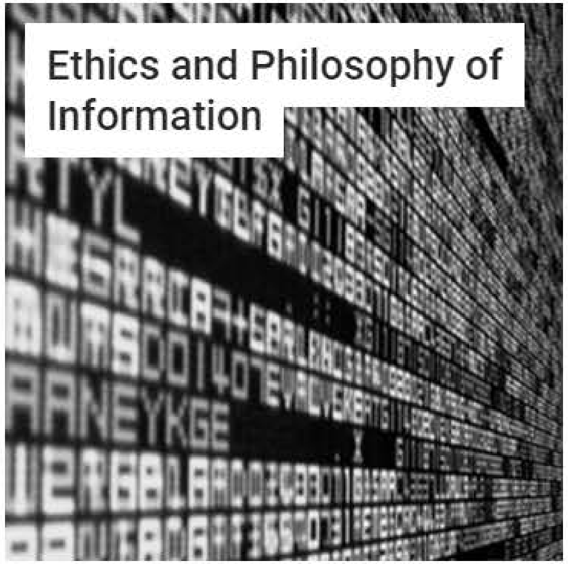
# SELF-INTRODUCTIONS

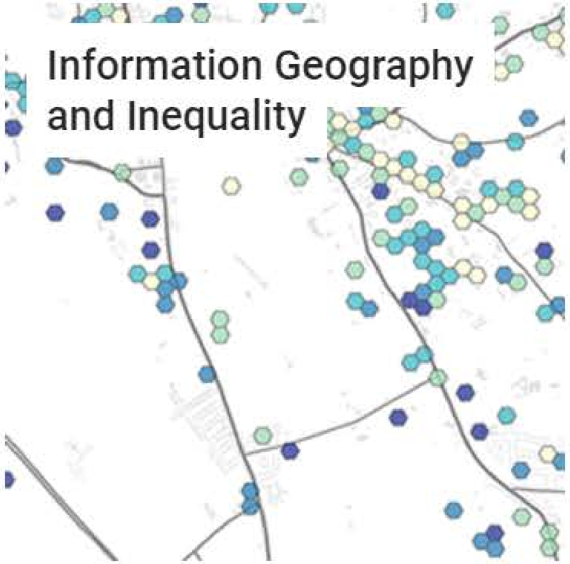
- Name
- Degree and subject
- University
- Previous experience with Python/programming
- Why you registered for this workshop
- Favourite corpus linguistics tool/concept

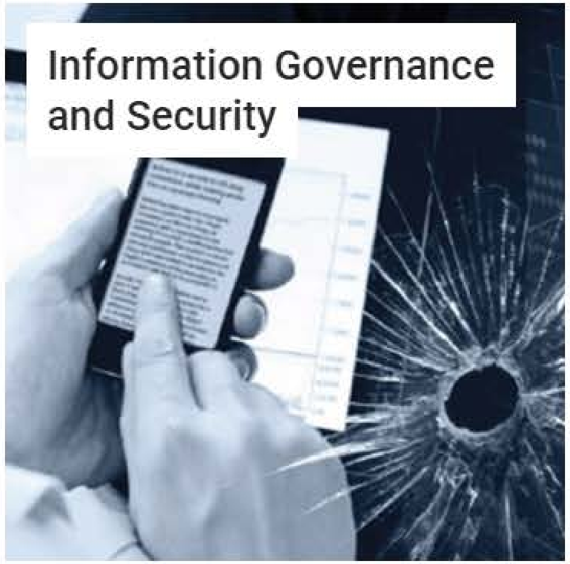


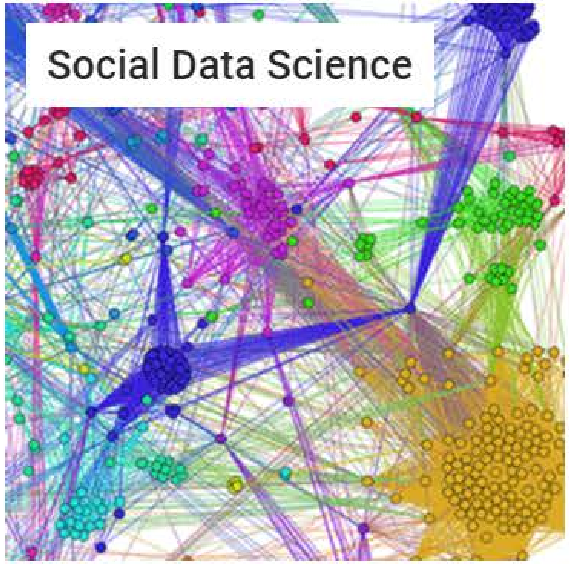




UNIVERSITY OF
OXFORD



# THE SCHEDULE

## WEDNESDAY

- **10.45-12.45**: Introductions & troubleshooting / setup
- **14.00-15.30**: Review of Python / Corpus Linguistics with NLTK
- **16.00-17.30**: Corpus Linguistics with NLTK continued

## THURSDAY

- **10.45-12.45**: Topic modelling overview
- **14.00-15.30**: Vectorization & topic modelling with scikit-learn
- **16.00-17.30**: Topic modelling visualisation / closing remarks

# GITHUB REPOSITORY

tinyurl.com/TartuNLP

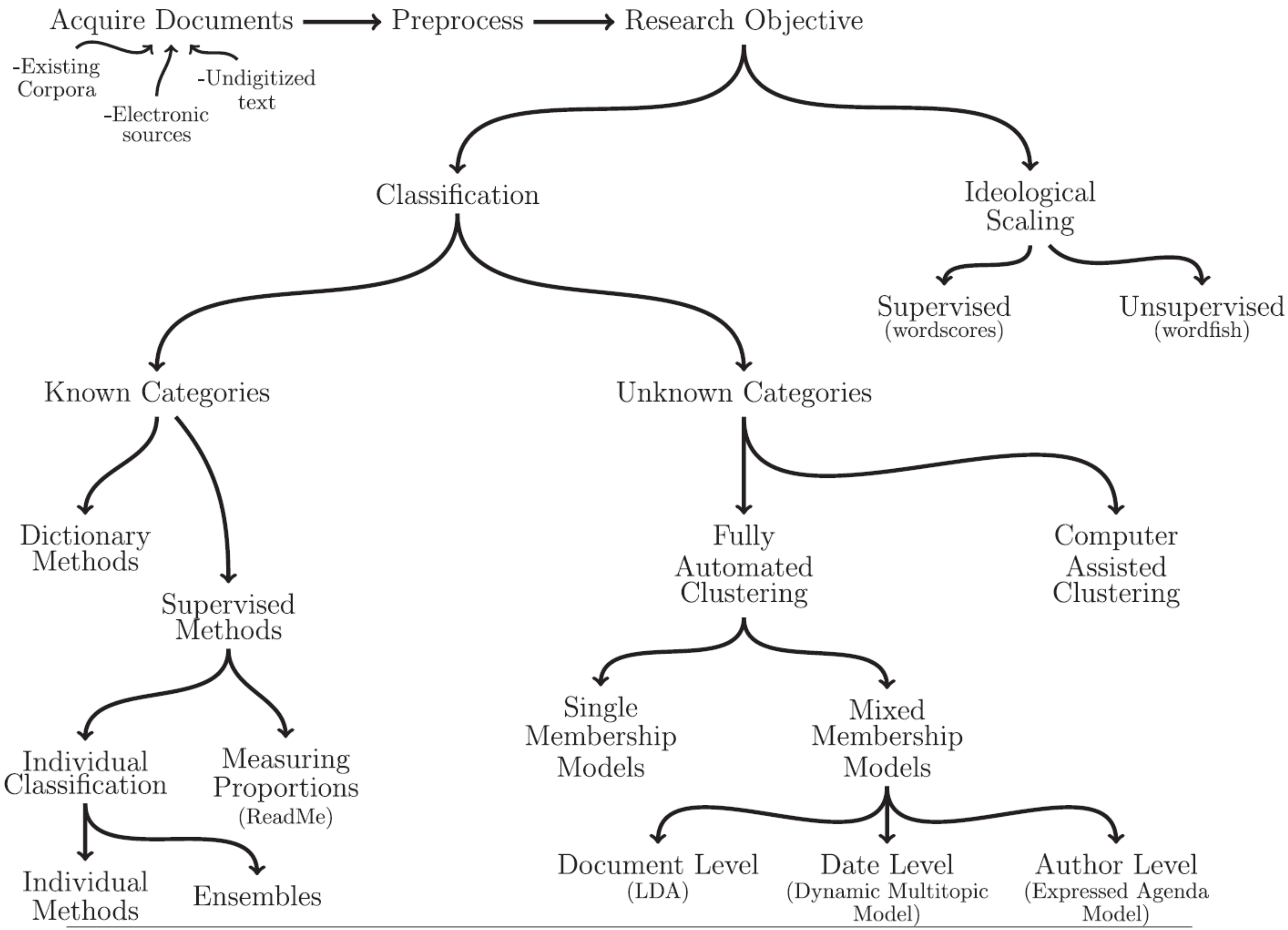



**Fig. 1** An overview of text as data methods.

Grimmer and Stewart, 2013

# TOPIC MODELLING APPLICATIONS

Topic modelling is an extremely popular NLP technique with applications across many domains. It is broadly connected to **text summarisation**. Some specific industrial and academic examples:

1. Categorisation of limitless volumes of legal documents and news stories by lawyers and journalists.
2. Restriction of JSTOR search results to specific categories.
3. Agenda setting of U.S. Congressional statements.
4. Author gender in 19th-century literature.
5. Trends in academic fields based on PhD abstracts.

# TOPIC MODELLING ASSUMPTIONS

All algorithms share the same core assumptions:

1. **Documents** are composed of mixtures of **topics**.
2. **Topics** are composed of mixtures of **words**.
3. Topics can be *inferred* from **word-document co-occurrences**.

On a broader level, topic models are grounded upon the idea that *meanings of documents are governed by **latent variables*** (topics). The goal is to uncover them, and there are various approaches for doing so.

# TOPIC MODELLING ALGORITHMS

There are two basic types of topic models:

1. **Matrix decomposition**, as represented by *Latent Semantic Analysis* (LSA, also known as truncated Singular Value Decomposition).
2. **Probabilistic inference**
   - *Probabilistic LSA* (pLSA)—rarely used on its own. Document probabilities are fixed.
   - *Latent Dirichlet Allocation* (LDA)—most popular and generalizable ('distribution over distributions'). Bayesian pLSA.

The approaches have the same input and similar output, but different maths.

# BAG OF WORDS (VECTORIZATION)

“This is how you get ants.”

tokenizer

['this','is','how','you','get','ants']

Build a vocabulary over all document

['aardvak','amsterdam','ants', ...'you','your','zyxst']

Sparse matrix encoding

aardvak ants get you zyxst

[0, …, 0, 1, 0, … , 0, 1 , 0, …, 0, 1, 0, …., 0 ]

Andreas Mueller, 2017

# Corpus

## (document-term frequency matrix)

| | Word 1 | Word 2 | Word 3 | ... | Word k |
|---|---|---|---|---|---|
| Doc 1 | $c_{11}$ | $c_{12}$ | $c_{13}$ | ... | $c_{1k}$ |
| Doc 2 | $c_{21}$ | $c_{22}$ | $c_{23}$ | ... | $c_{2k}$ |
| Doc 3 | $c_{31}$ | $c_{32}$ | $c_{33}$ | ... | $c_{3k}$ |
| ... | ... | ... | ... | ... | ... |
| Doc n | $c_{n1}$ | $c_{n2}$ | $c_{n3}$ | ... | $c_{nk}$ |


Vocabulary

SVD

Inference

| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| Doc 1 | 0.2 | 0.4 | 0.4 |
| Doc 2 | 0.9 | 0.05 | 0.05 |
| Doc 3 | 0.1 | 0.1 | 0.8 |
| ... | ... | ... | ... |
| Doc n | 0.8 | 0.1 | 0.1 |

LSA


LDA

| | Topic 1 | Topic 2 | Topic 3 |
|---|---|---|---|
| Word 1 | 0.001 | 0.002 | 0.005 |
| Word 2 | 0.009 | 0.019 | 0.009 |
| Word 3 | 0.002 | 0.001 | 0.004 |
| ... | ... | ... | ... |
| Word n | 0.000 | 0.001 | 0.002 |

LSA


LDA

William Zheng, 2017

# LATENT DIRICHLET ALLOCATION

LDA is a *generative model*: it specifies a procedure through which documents are written (generated). Its assumption about how to write a text is quite naïve:

1. Choose the number of words *N* in your document
2. Choose a topic mixture θ for your document (e.g., 60% about topic 1 and 40% about topic 2)
3. While the number of generated words is smaller than *N*, generate a word $w_i$ by:
   - Choosing a topic according to the chosen topic mixture θ
   - Choosing a word according to the topic's vocabulary distribution

# LATENT DIRICHLET ALLOCATION

According to this procedure, we might write (generate) a document as follows:

1. We decide that it will be 4 words long.
2. 25% will be about topic 1 (religion) and 75% about topic 2 (cars).
3. By choosing from the vocabularies of topic 1 and topic 2, we generate four words: *Ferrari* (T2), *engine* (T2), *Jesus* (T1), *drive* (T2). This is our document.

This **bag of words (BoW)** approach is linguistically absurd, but allows us to *infer* topic mixtures and their associated words.

# Real-World Example

Finally, I applied LDA to a set of Sarah Palin's emails a little while ago (see here for the blog post, or here for an app that allows you to browse through the emails by the LDA-learned categories), so let's give a brief recap. Here are some of the topics that the algorithm learned:

- **Trig/Family/Inspiration**: family, web, mail, god, son, from, congratulations, children, life, child, down, trig, baby, birth, love, you, syndrome, very, special, bless, old, husband, years, thank, best, …
- **Wildlife/BP Corrosion**: game, fish, moose, wildlife, hunting, bears, polar, bear, subsistence, management, area, board, hunt, wolves, control, department, year, use, wolf, habitat, hunters, caribou, program, denby, fishing, …
- **Energy/Fuel/Oil/Mining:** energy, fuel, costs, oil, alaskans, prices, cost, nome, now, high, being, home, public, power, mine, crisis, price, resource, need, community, fairbanks, rebate, use, mining, villages, …
- **Gas**: gas, oil, pipeline, agia, project, natural, north, producers, companies, tax, company, energy, development, slope, production, resources, line, gasline, transcanada, said, billion, plan, administration, million, industry, …
- **Education/Waste**: school, waste, education, students, schools, million, read, email, market, policy, student, year, high, news, states, program, first, report, business, management, bulletin, information, reports, 2008, quarter, …
- **Presidential Campaign/Elections**: mail, web, from, thank, you, box, mccain, sarah, very, good, great, john, hope, president, sincerely, wasilla, work, keep, make, add, family, republican, support, doing, p.o, …

'having', 'lived', 'played', 'and', 'worked', 'and', 'near', 'the', 'navajo', 'reservation', 'for', 'number', 'years', 'can', 'confirm', 'this', 'ancient', 'pattern', 'found', 'petroglyphs', 'dated', 'years', 'old', 'also', 'the'

Edwin Chen, 2011

# DEEP LEARNING & LDA

One of the most recent extensions of LDA is **lda2vec**, which is also an extension of **word2vec**. It is a deep learning model that jointly learns word, document, and topic vectors (embeddings).

- *word2vec* uses a skipgram neural network model to generate **word vectors**. These vectors are used to predict **context words**.
- *lda2vec* uses a **context vector** to make predictions: the **word vector** + the **document vector** (topic weight vector + topic matrix).

More information: [www.github.com/cemoody/lda2vec](www.github.com/cemoody/lda2vec)

# THANK YOU & STAY IN TOUCH! ☺

yin.lu@oii.ox.ac.uk

@yinneth

linkedin.com/in/periwynkle