UNIVERSITY OF EXETER

# Natural Language Processing / Text-As-Data Workshop

Julian Dyer

# Why should economists care about text analysis?

- Vast amount of data that economists haven't made use of yet.
  - Storing and accessing massive text databases becoming easier
  - Digitizing old documents (newspapers, archives, etc.) also becoming easier

- Processing becoming easier too
  - Computing power much cheaper
  - Lots of off-the-shelf tools available

- Most importantly: extremely rich source of data
  - Can measure outcomes of interest as well as measuring treatments
  - Captures subtle outcomes: opinions, worldviews, concepts, etc.
  - Strong external validity since it comes from the "real" world

# Topics for Today

- Cover basic tools, introduce a few important python packages
- Topics:
  1. "Bag-of-Words" approach
  2. Topic Modelling
  3. Sentiment Analysis
  4. Word-Vector Embedding
  5. Semantic Similarity & Arithmetic

# Bag of Words (BoW)

- Simplest way to model language - just treat a text as a collection of words (or “bag of words”)
- Look at the count of different words in a document after cleaning the text
- Simple, but can be very useful: identify occurrences of an event, identify specific keywords of interest, etc.
    - Even the total count of words can be informative

# Topic Modelling

- Now, we can think not just about counting words, but understanding which are the relevant keywords that characterise a document
- Term-Frequency Inverse Document-Frequency
  - Intuitively, identify words that are common in a document, taking into account how many documents contain that word
- From TF-IDF scores, we can then at which keywords occur together in documents to identify *topics* that are charactarised by keyword weights
  - Then use this modelling to assign topic-scores to a document

# Sentiment Analysis

- In addition to understanding what topics we are talking about, we can also look at how people feel about different topics

- For this we use a *valence-aware* sentiment analysis package, which is able to interpret negation words (e.g "that wasn't terrible") or other modifiers ("that was absolutely terrible")

# Word Embedding Models

- Now we go from thinking about which words appear in the same documents to identify topics to thinking about which words appear in the same context
- We do this by representing words as vectors, where each dimension is a *feature* that describes something that makes words similar in meaning
- By representing words in this way, we can then look at the distance in semantic space to measure how similarity of words
  - By looking at embeddings trained on different bodies of text, we can measure how different sources/people view the world

# Word-Vector Arithmetic

- These vector representations of words can be used to do intuitive arithmetic to “move around” semantic space
- How might you try to describe the concept of a ‘queen’?

# Word-Vector Arithmetic

- These vector representations of words can be used to do intuitive arithmetic to “move around” semantic space
- How might you try to describe the concept of a ‘queen’?
  - One explanation might be to say *“A queen is a ruler like a king, but who is a woman instead of a man”*

# Word-Vector Arithmetic

- These vector representations of words can be used to do intuitive arithmetic to “move around” semantic space

- How might you try to describe the concept of a ‘queen’?
  - One explanation might be to say *“A queen is a ruler like a king, but who is a woman instead of a man”*
  - In arithmetic: $\text{queen} \simeq \text{king} + \text{woman} - \text{man}$

# Word-Vector Arithmetic

- These vector representations of words can be used to do intuitive arithmetic to "move around" semantic space
- How might you try to describe the concept of a 'queen'?
  - One explanation might be to say *"A queen is a ruler like a king, but who is a woman instead of a man"*
  - In arithmetic: $\text{queen} \simeq \text{king} + \text{woman} - \text{man}$
- This arithmetic works in semantic space

# Conclusion

- There are many, many more things you can do with Natural Language Processing
- More advanced packages using more complex models including BERT built on recurrent neural networks (RNNs) that do a better job learning contextual understanding of words
- Happy to chat if you have a specific application in mind!