Introduction to Topic Modelling – workshop 1

Bertie Vidgen & Taha Yasseri

Oxford Internet Institute (University of Oxford) & Alan Turing Institute

Monday 23rd April 2018

Overview

Session 1

- Introduction to topic modelling
- Practical – implement a topic model
- Questions

Session 2

- Advanced topic modelling
- Practical – analysis & fitting parameters
- Questions

Bring your own data

Text… there's a lot of it

- The "big data revolution" or "data deluge"
  - 500 million tweets sent every day
  - 40 million articles on Wikipedia
  - 2.5 million academic articles published each year
- Unstructured data
  - No obvious way of ordering or categorizing
  - No metadata (e.g. tags) describing what the text contains
  - Estimated 90% of all newly created text is unstructured

Many dimensions of text can be studied

"Britain should NOT launch military action in Syria. Assad is a bad guy but regime ALWAYS leads to chaos and ISIS."

@Nigel_Farage Wednesday 11th April 2018

Content	Style	Expression	Features
Topics Syria, ISIS, military, Britain, security	Spelling No errors	Sentiment Confident, calm, direct	Action Recommendation to (not) act
Policy Rejection of foreign intervention	Lexicon Fairly broad, e.g. 'launch', 'regime'	Polarity Many negative words: 'bad', 'not', 'chaos'	Length Short
Ideology hard to discern – maybe populist?	Syntax Grammatically correct	Urgency Two capitalizations	Tense Future and Present
		Audience No specific individual targeted	Links None embedded

When should you use a topic model?

1. You have a large amount of text
   - 12 million YouTube comments, Ottoni et al. (2018)
   - 2 million tweets, Hong and Davison (2010)
   - 1 million + dissertation abstracts, McFarland et al. (2013)
   - 118,000 Congressional speeches [70 million words], Quinn et al. (2010)
   - 51,000 news-stories, Bonilla and Grimmer (2013)

2. You want to understand the *themes*
   - What are the authors talking about?
   - What are the most prevalent topics?

Rough guide: ~1,000 documents is usually the minimum for a decent topic model (though it depends on the length of each document)

What can we do with topic modelling?

1. Summarize large quantities of text
2. Cluster documents
3. For any document, find similar documents
4. Further analyses
    1. Map relationships between topics
    2. Look at topical prevalence over time
    3. Link topics to another variable e.g. the document author

Topic modelling is very well-suited for ***exploratory*** research

What is good about topic modelling?

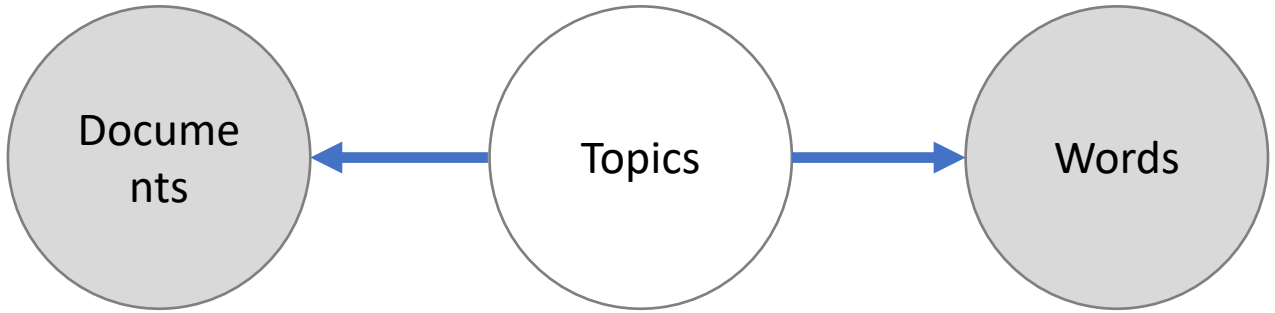
- Fast
  - Can be implemented in real-time
- Scalable
  - Models can be easily extended to include new data
- Robust
  - Not based solely on human judgement and subjective opinion
- Generalizable
  - Works across contexts, languages, geographies
- Easy to implement (relatively)
  - Well-documented versions in R, Python and C

Motivation & aims of today

- Topic models are super interesting and powerful, but…
  - Few introductory articles aimed at non-technical audiences
  - Hard to find suitable implementations and decent walkthroughs
  - Lots of extensions and applications, which can be confusing
- Aims
  - Implement a topic model
  - Understand the output

The basic idea behind topic modelling

- We observe documents and words but there is a hidden layer in between: the *topic*. The goal of topic modelling is to uncover these topics
- Lazy writer assumption – we assume that writers' only goal is to talk about topics (to differing extents). Words are just instantiations of these topics
- Topics are multinomial distributions over words



DOI:10.1145/2133806.2133826

Surveying a suite of algorithms that offer a solution to managing large document archives.

BY DAVID M. BLEI

Probabilistic Topic Models

For example, consider using themes to explore the complete history of the New York Times. At a broad level, some of the themes might correspond to the sections of the newspaper—foreign policy, national affairs, sports. We could zoom in on a theme of interest, such as foreign policy, to reveal various aspects of it—Chinese foreign policy, the conflict in the Middle East, the U.S.'s relationship with Russia. We could then navigate through time to reveal how these specific themes have changed, tracking, for example, the changes in the conflict in the Middle East over the last 50 years. And, in all of this exploration, we would be pointed to the original articles relevant to the themes. The thematic structure would be a new kind of window through which to explore and digest the collection.

Blei (2012) 'Probabilistic topic models', *Communications of the ACM*, 55: 4, pp. 77-84

First practical

- Script
  - topic-modelling-workshop 1 – practical 1.R
- Data
  - workshop-1-practical-1.Rdata
    - We are working with petitions data from the 2015-2017 parliament – collected from https://petition.parliament.uk

Those topics aren't terribly informative…

- Topic 3, topic 9 and topic 10 only contain generic words ('the', 'for', 'and')
- Topic 5 contains only junk words ('http', 'www', 'com')
- Topic 4 contains lots of words which are quite unusual ('rexie', 'lemmy', 'vorek')
- Topic 1, topic 2, topic 6, topic 7 and topic 8 actually seem quite good
- But it is surprising issues like the Referendum, NHS, Schools and International affairs have not come up at all

What output do you get? – Documents

- For each document we get the distribution over topics
- The topic probabilities for each document sum to one
- Every topic has *some* probability of occurring in each document, even if it is very small

	network	security	finance	technology	technology2
Document 1	0.1662995595	0.0011013216	0.0451541850	0.0011013216	0.0011013216
Document 2	0.0662393162	0.0021367521	0.9209401709	0.0021367521	0.0021367521
Document 3	0.0015432099	0.0015432099	0.9274691358	0.0169753086	0.0478395062
Document 4	0.0017921147	0.0017921147	0.9157706093	0.0017921147	0.0197132616
Document 5	0.1657608696	0.0027173913	0.8179347826	0.0027173913	0.0027173913

Things to note

1. We might want to combine some topics – e.g. ‘technology’ and ‘technology2’
2. You add the topic labels manually

What output do you get? – Topics

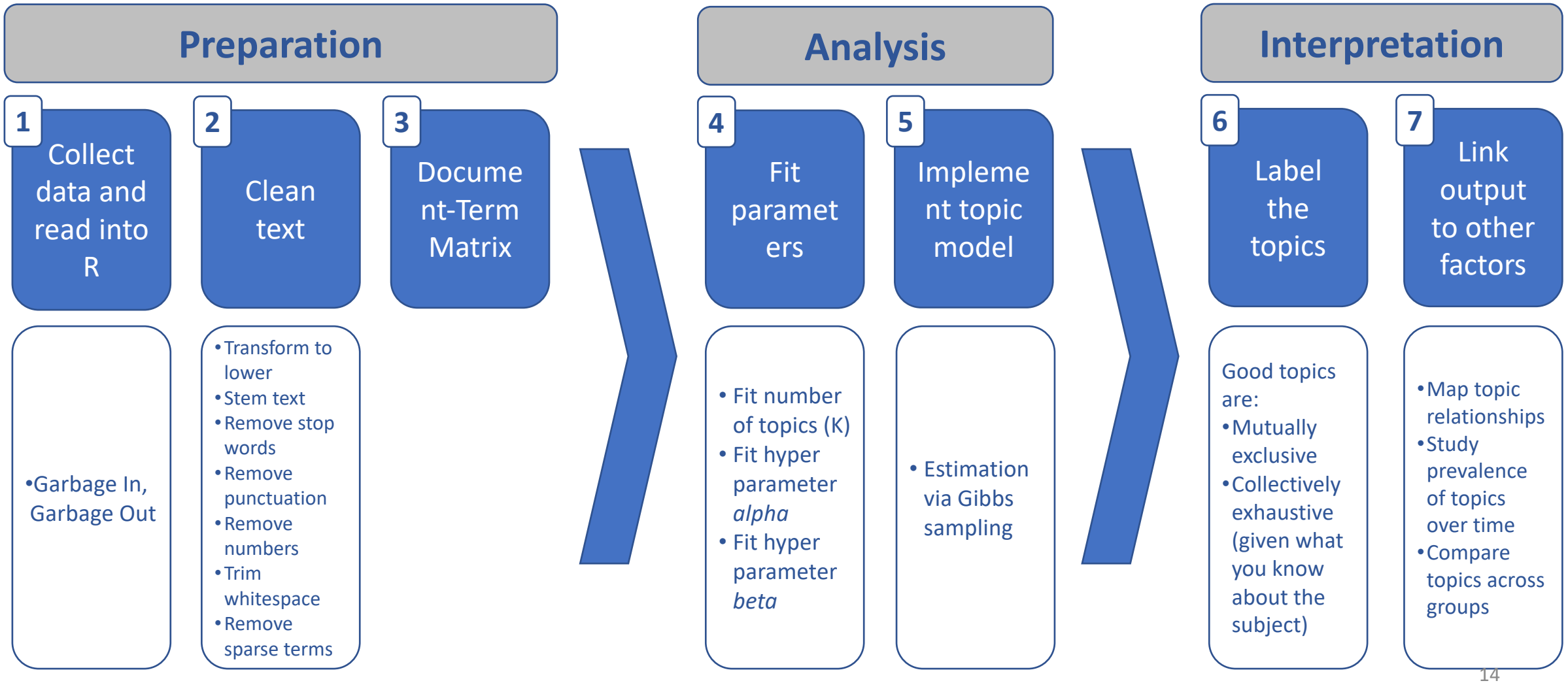
- For each topic we get the distribution over words
- The word probabilities for each topic sum to one
- Every word has *some* probability of occurring in each topic, even if it is very small

	network	security	finance	technology	technology2
abandon	0.00001145161	0.00001942728	0.00001578308	0.000006538555	0.000039807861
abat	0.00001145161	0.00001942728	0.00001578308	0.000006538555	0.000006634643
abbrevi	0.00001145161	0.00001942728	0.00001578308	0.000006538555	0.000006634643
abnorm	0.00001145161	0.00011656370	0.00025252924	0.000006538555	0.000006634643
abolit	0.00001145161	0.00001942728	0.00001578308	0.000006538555	0.000006634643

Things to note:

1. Most words have a very low probability of occurring in *any* topic
2. Words are *stemmed* so may have slightly unusual spellings

Topic modelling pipeline



Cleaning text – the Goldilocks approach

2

TOO HOT

- Terms which contain little topical information and appear everywhere
  - e.g. stop words like ‘the’ & ‘and’
- Terms which are domain-specific and occur in every document
  - e.g. if we are researching blockchain then we would remove the terms ‘blockchain’ and ‘bitcoin’

TOO COLD

- Terms which are very ‘sparse’
  - e.g. they occur in just 2 or 3 documents
- Idiosyncratic variations of words so we *stem*
  - e.g. ‘photography’, ‘photo’, ‘photographic’, ‘photogenic’

Create a Document Term Matrix (DTM)

3

- Each document is a row and each term is a column
- Every term in the vocabulary has a separate column
  - A DTM is a ‘one hot encoding’ matrix as it tends to be very sparse. Most documents contain only a few of the terms in the vocabulary
- Example:
  - “The cow goes moo”
  - “The dog goes bark”
  - “The lion is suspiciously silent’

Terms	the	cow	goes	moo	dog	bark	lion	is	suspicio usly	silent
Doc 1	1	1	1	1	0	0	0	0	0	0
Doc 2	1	0	1	0	1	1	0	0	0	0
Doc 3	1	0	0	0	0	0	1	1	1	1

Parameters – choose the number of topics

4

- The number of topics is the most important input to your model
  1. Aim for a few topics (e.g. 10-20) which can each easily be interpreted
  2. Aim for quite a few topics (e.g. 50) but then manually combine to reduce to an interpretable number
  3. Aim for many topics (e.g. 100) but then use clustering to sort into e.g. 5-10 higher level groups

Note – this is just a guideline! Selecting the right number of topics depends on your data and research goals

Implement topic model

5

- Topic models work by assuming that there is a ‘hidden layer’ in between the words and documents – *the topics*
- We are using ‘Latent Dirichlet Allocation’ (LDA): this algorithm tries to estimate this hidden layer based on the word co-occurrences
  - *Topic modelling algorithms don’t understand English! They work based purely on the word co-occurrences and document structure*
- Every time we run a topic model we get slightly different results – the algorithm finds a local optimum

Label the topics

6

- Very important for interpreting the output
  - Often we only work with the topics, so what you call them really matters
- Good topics are:
  - Mutually exclusive
  - Collectively exhaustive (given what you know about the subject)
- Labelling topics is hard – it's a very active area of LDA research
  - Often, topics are very 'noisy'- sometimes 30% of your topics are too messy to label
  - A great paper is Chang et al. (2009) 'Reading tea leaves: how humans interpret topic models' *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, pp. 288-296

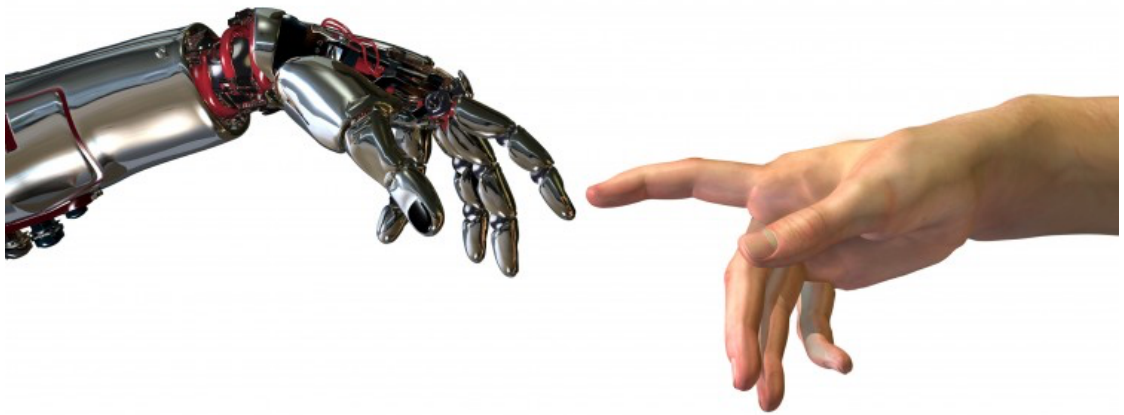
Second practical

- Script
  - topic-modelling-workshop 1 – practical 2.R
- Data
  - workshop-1-practical-2.Rdata

Three principles of computational text analysis

1. All quantitative models of language are wrong – but some are useful
2. Quantitative methods augment humans, not replace them
3. There is no single best method for automated text analysis

Grimmer and Stewart (2013) 'Text as Data: the promise and pitfalls of automatic content analysis methods for political texts'



Some final points to remember

- Topic models are hard to evaluate. There is no ‘right answer’
  - The only question worth asking – is the model *useful*?
- Your output is only as good as the (cleaned) data you use
  - Garbage In, Garbage Out
- Topic models find local optimums, so every time you’ll get a different result
  - Use the seed argument in R to get replicable output

Thank you!

- The second workshop is tomorrow, same time and place (14.00 here at the OII)
- Email me if you have any issues with the code
- Bring your own data tomorrow and we can discuss how to model it

bertie.vidgen@oii.ox.ac.uk

References

- David Blei et al. (2003) ‘Latent Dirichlet Allocation’, *Journal of Machine Learning Research*, 3, pp. 993-1022
  - The original paper on LDA – it gets a bit technical but the first part is very interesting
- David Blei (2012) ‘Probabilistic topic models’, *Communications of the ACM*, 55:4, pp. 77-84
  - Fantastic accessible introduction by the LDA granddaddy himself
- John Mohr and Petko Bogdanov (2013) ‘Topic models and the cultural sciences’, *Poetics*, 41: 6, pp. 545-770
  - A special edition of *Poetics* with both a good overview of LDA and some nice applications
- Justin Grimmer and Brandon Stewart (2013) ‘Text as data: the promise and pitfalls of automatic content analysis methods for political texts’, *Political Analysis*, Summer, pp. 1-31
  - A good general introduction to computational text analysis
- Ted Underwood (2012) ‘Topic modeling made just simple enough’ - https://tedunderwood.com/2012/04/07/topic-modeling-made-just-simple-enough/
  - Very accessible blog post on topic models; particularly helpful if you’re still struggling with the general concept of topic modelling

Useful terms

- Corpus = a large selection of texts; typically, we use this term to refer to all of the texts we are studying. Corpuses usually have a degree of structure and in-built analysis, such as the text being cleaned and ordered.
- Document = any single bit of text we study, e.g. a petition, a tweet, a publication. We can also aggregate separate bits of text into documents, e.g. documents could be all of the tweets produced by each person
- Document-Term Matrix (DTM) = matrix representation of text, showing the counts of each term within each document
- LDA = Latent Dirichlet Allocation
- Lemmatization = reduce words to their 'lemma', which is the canonical version of a word (i.e. the word which appears in the dictionary). Often too difficult to implement, so we use stemming instead
- LSA = Latent Semantic Analysis
- Multinomial = describes the distribution of a random variable which can taken on one of *i* possible values (e.g. within a topic there are *i* different words, each one of which has a certain probability of occurring). In LDA we are technically using a special case of the multinomial – the categorical – where only one trial takes place.
- Sparsity = how many of your data points record an actual value (i.e. in a DTM, for each row how many of the column entries are a zero). 'Sparseness' is used to describe cases where there are insufficient data points to model language. This is a well-established problem in data science (and why word embeddings are so popular).
- Stemming = reducing inflected/derived words to their word stem, usually using the Porter Stemming algorithm. See here for a discussion on stemming and lemmas: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html