Centre for
Data Analytics

Insight

Topic Modelling with Scikit-learn

Derek Greene
University College Dublin

PyData Dublin – 2017

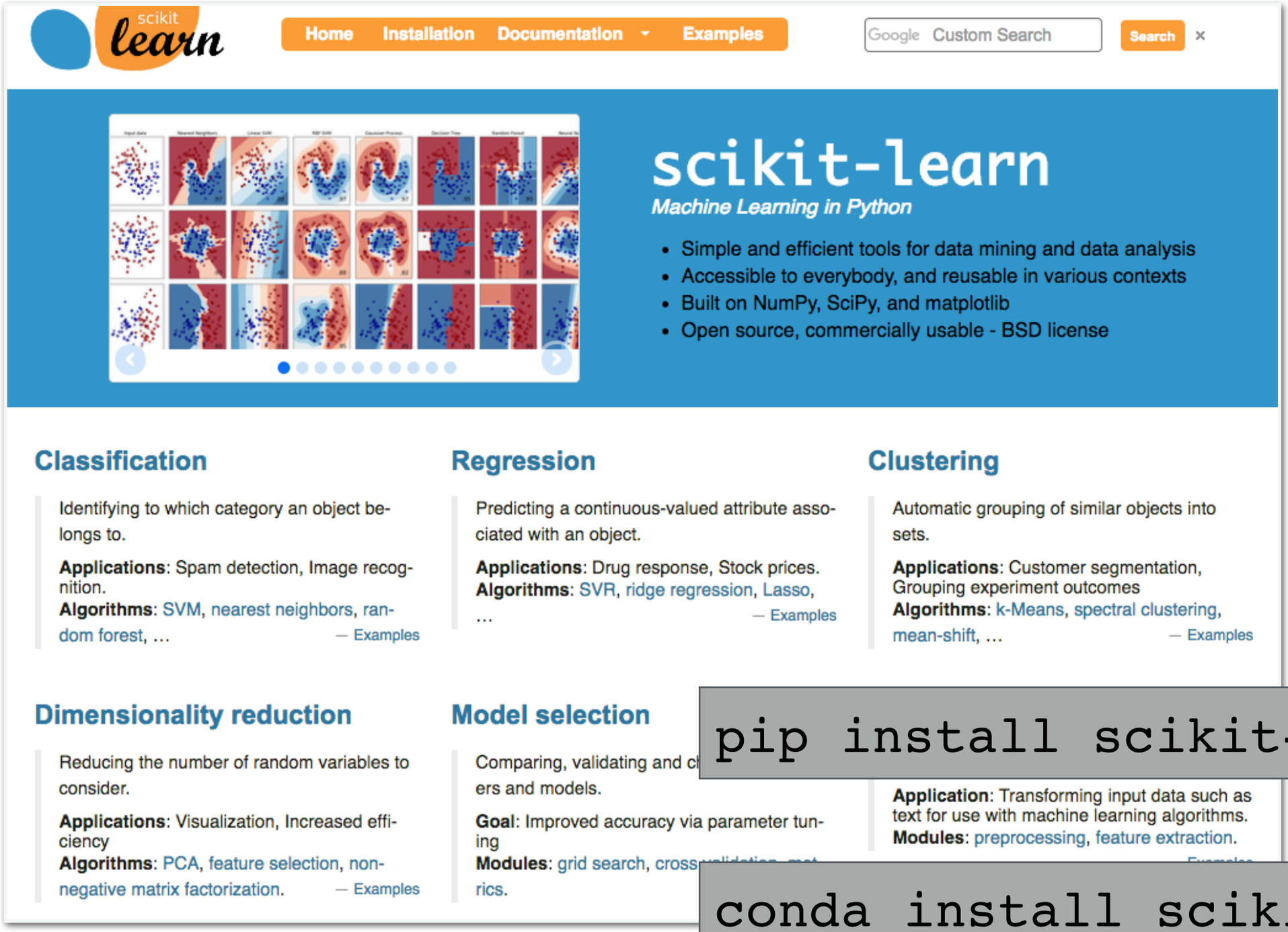
Overview

- Scikit-learn
- Introduction to topic modelling
- Working with text data
- Topic modelling algorithms
- Non-negative Matrix Factorisation (NMF)
- Topic modelling with NMF in Scikit-learn
- Parameter selection for NMF
- Practical issues

Code, data, and slides:

https://github.com/derekgreene/topic-model-tutorial

Scikit-learn



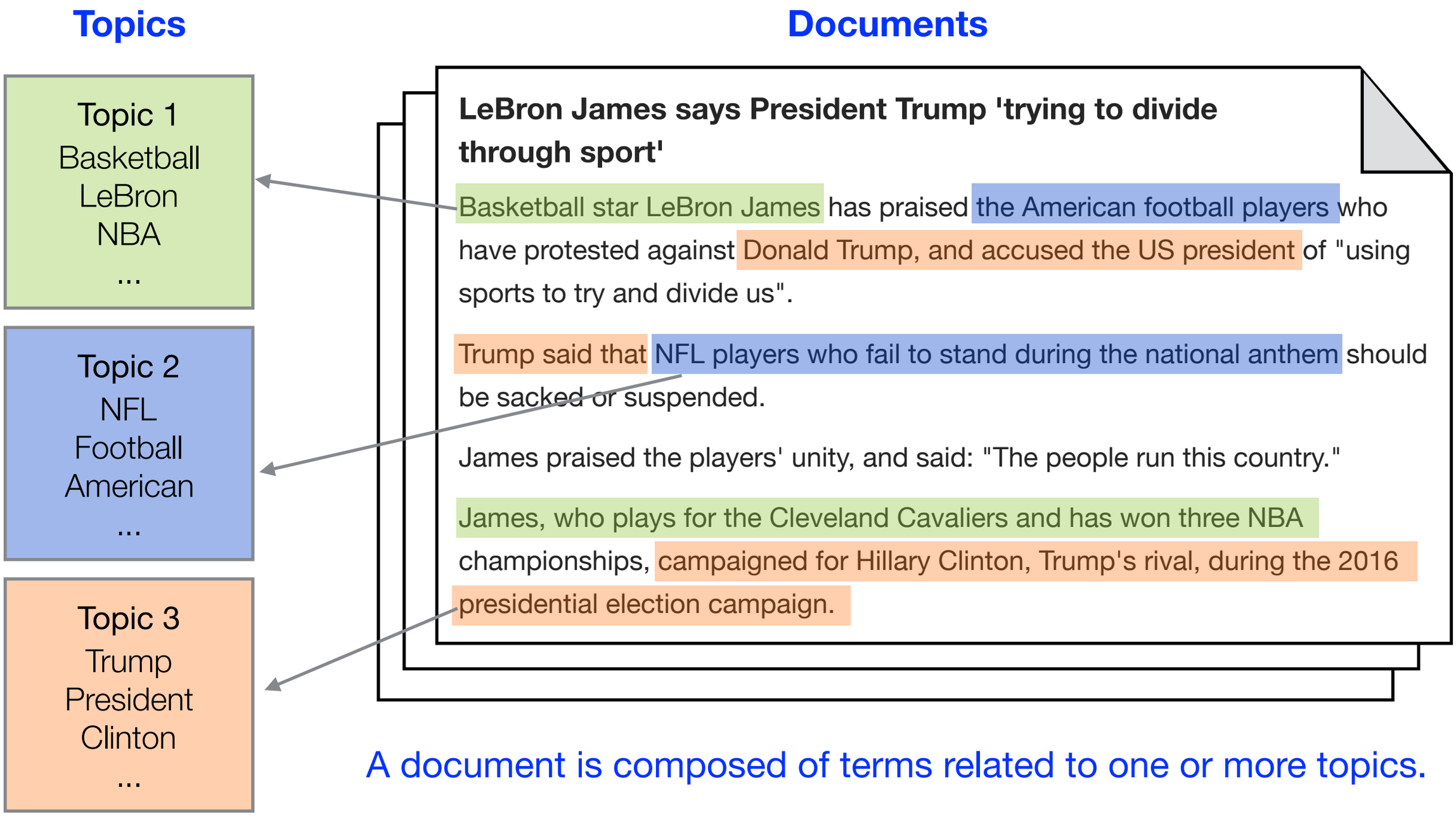

```
pip install scikit-learn
```

```
conda install scikit-learn
```

http://scikit-learn.org/stable

Introduction to Topic Modelling

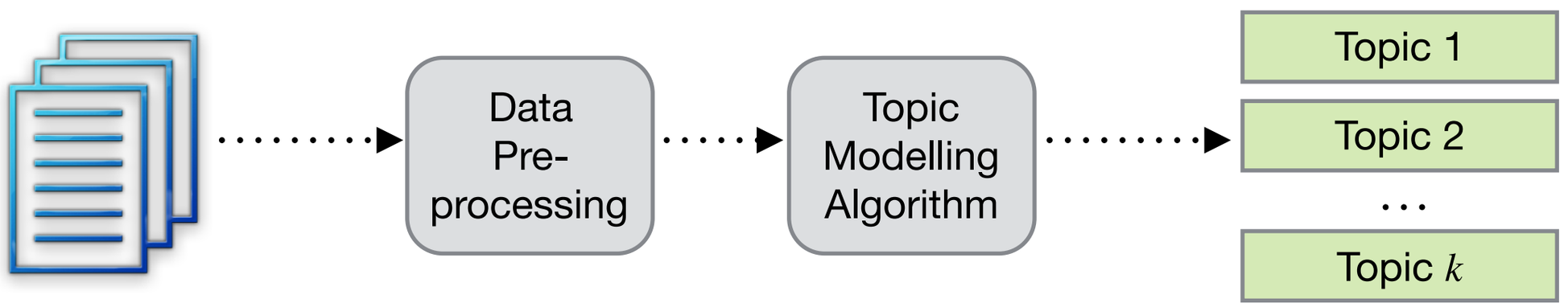
Topic modelling aims to automatically discover the hidden thematic structure in a large corpus of text documents.



A document is composed of terms related to one or more topics.

Introduction to Topic Modelling

- Topic modelling is an unsupervised text mining approach.
- **Input:** A corpus of unstructured text documents (e.g. news articles, tweets, speeches etc). No prior annotation or training set is typically required.



- **Output:** A set of k topics, each of which is represented by:
  1. A descriptor, based on the top-ranked terms for the topic.
  2. Associations for documents relative to the topic.

Introduction to Topic Modelling

Top Terms for Topic 1

Top Terms for Topic 2

Top Terms for Topic 3

Top Terms for Topic 4

Introduction to Topic Modelling

In the output of topic modelling, a single document can potentially be associated with multiple topics…

Election 2015: Parties row over GP out-of-hours cover

3 hours ago | Election 2015

Figures suggest almost 600 fewer GP surgeries in England open at evenings and weekends than before 2010, Labour has claimed.

Health spokesman Andy Burnham said the coalition had created queues outside practices and diverted people to A&E.

Tory Health Secretary Jeremy Hunt said Labour's numbers were wrong and that out-of-hours cover was being extended.

The Lib Dems also said Labour's figures - obtained through a **parliamentary question** - were out of date.

Mr Burnham announced the analysis as his party unveiled a new poster, which reworks the Conservatives' "Labour isn't working" image of 1979 by depicting a huge queue outside a waiting room with the title: "The doctor can't see you now."

Politics or Health?

Chelsea FC reports a record £18m in annual profit

13 November 2014 | Business

Chelsea Football Club has reported a record profit of £18.4m ($29m) for the year to June 2014 - despite last season's lack of silverware.

The London team has only once before turned a profit in the 10 years since it was acquired by Russian billionaire Roman Abramovich.

The club said a new TV broadcasting deal, as well as the sale of players such as Juan Mata, had boosted profits.

Chelsea are currently unbeaten at the top of the Premier League.

Business of Sport

Premier League to share £1bn

Adidas unveils Europe factory plan

Fans to protest over ticket prices

Controversial Gatlin gets Nike deal

Business or Sport?

Application: News Media

We can use topic modelling to uncover the dominant stories and subjects in a corpus of news articles.


EU referendum: Cameron claims leaving EU could make cutting immigration harder - Politics live

David Cameron doing a Q&A on Europe with employees at the BAE Systems factory in Preston. Photograph: Peter Byrne/PA

Donald Trump hits delegate count needed for Republican nomination – as it happened

Republican presidential candidate Donald Trump speaks in Albuquerque, New Mexico on 24 May 2016. Photograph: Brennan Linsley/AP

Topic 1

Rank	Term
1	eu
2	brexit
3	uk
4	britain
5	referendum

Article Headline	Weight
Archbishop accuses Farage of racism and 'accentuating fear'	0.20
Cameron names referendum date as Gove declares for Brexit	0.20
Cameron: EU referendum is a 'once in a generation' decision	0.18
Remain camp will win EU referendum by a 'substantial margin'	0.18
EU referendum: Cameron claims leaving EU could make cutting...	0.18

Topic 2

Rank	Term
1	trump
2	clinton
3	republican
4	donald
5	campaign

Document Title	Weight
Donald Trump: money raised by Hillary Clinton is 'blood money'	0.27
Second US presidential debate – as it happened	0.27
Donald Trump hits delegate count needed for Republican nomination	0.26
Trump campaign reportedly vetting Christie, Gingrich as potential...	0.26
Trump: 'Had I been president, Capt Khan would be alive today'	0.26

Application: Social Media

Topic modelling applied to 4,170,382 tweets from 1,200 prominent Twitter accounts, posted over 12 months. Topics can be identified based on either individual tweets, or at the user profile level.

Topic 1

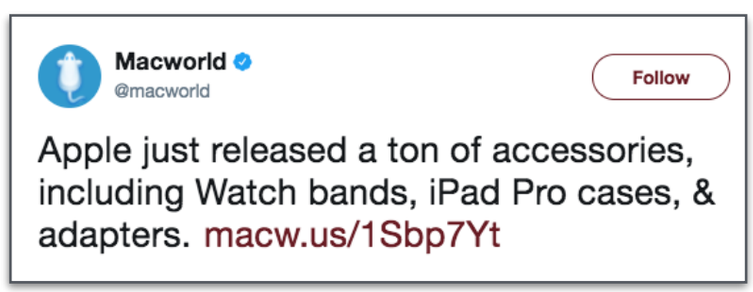
Rank	Term
1	space
2	#yearinspace
3	pluto
4	earth
5	nasa
6	mars
7	mission
8	launch
9	#journeytomars
10	science

Topic 2

Rank	Term
1	#health
2	cancer
3	study
4	risk
5	patients
6	care
7	diabetes
8	#zika
9	drug
10	disease

Topic 3

Rank	Term
1	apple
2	iphone
3	#ios
4	ipad
5	mac
6	app
7	watch
8	apps
9	os
10	tv



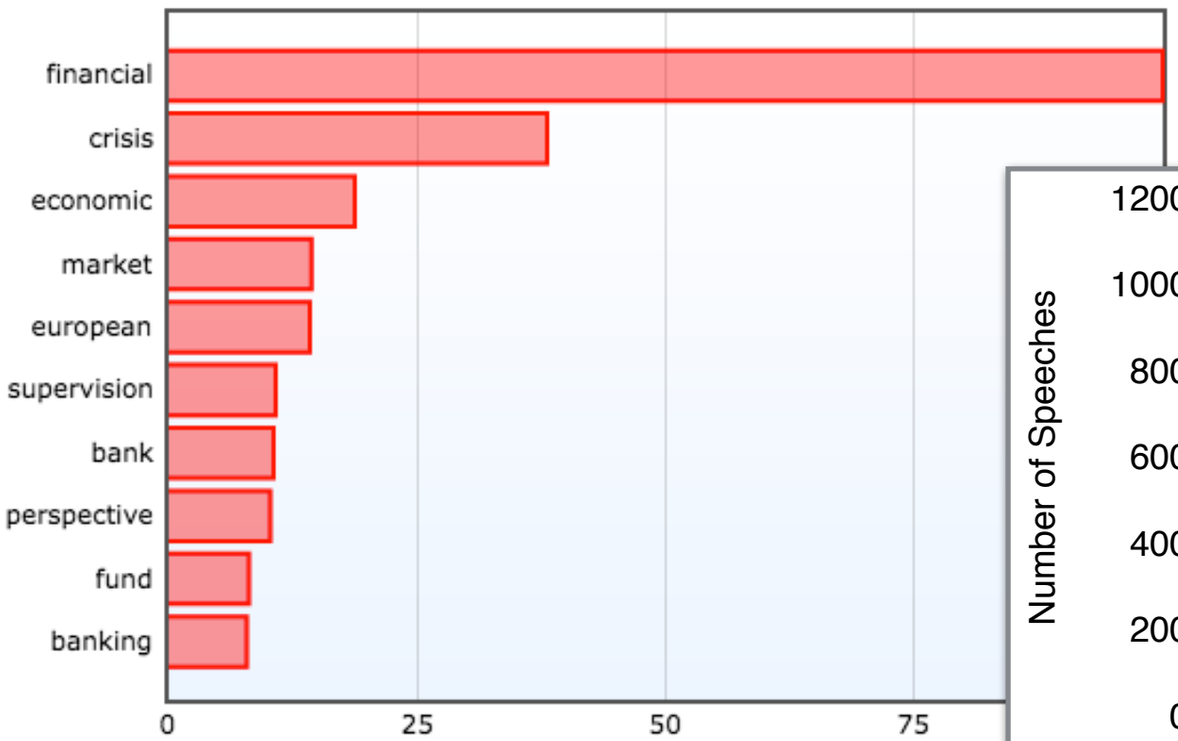
Application: Political Speeches

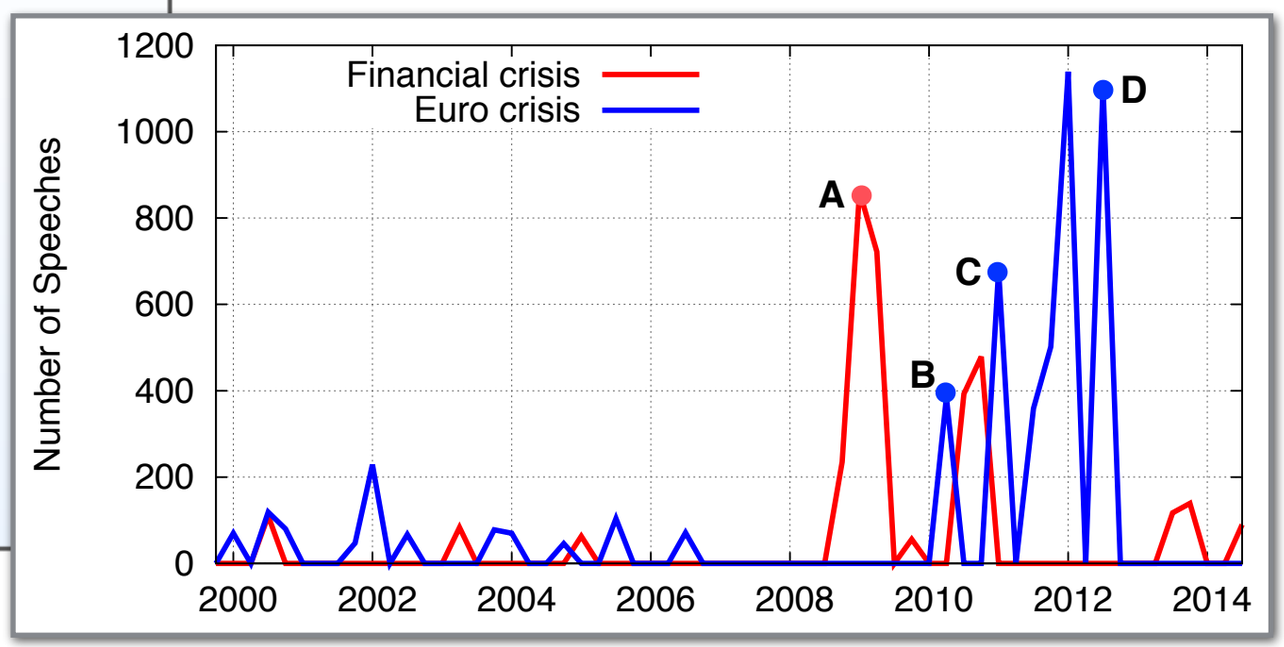
Analysis of 400k European Parliament speeches from 1999-2014 to uncover agenda and priorities of MEPs (Greene & Cross, 2017).

Seán Kelly (PPE). – *(GA)* Mr President, I was happy to give my support to this report, but I believe that the earlier we can put a single market in place, the better it will be for all of us, especially to escape from the economic recession and to create jobs and growth in the European Union.

I find it somewhat bemusing to hear that this addresses the concerns of European citizens and businesses. It certainly addresses the concerns of businesses but, unfortunately, where most citizens are concerned, they are probably not even aware of the single market and what it means. That certainly is a challenge for us.

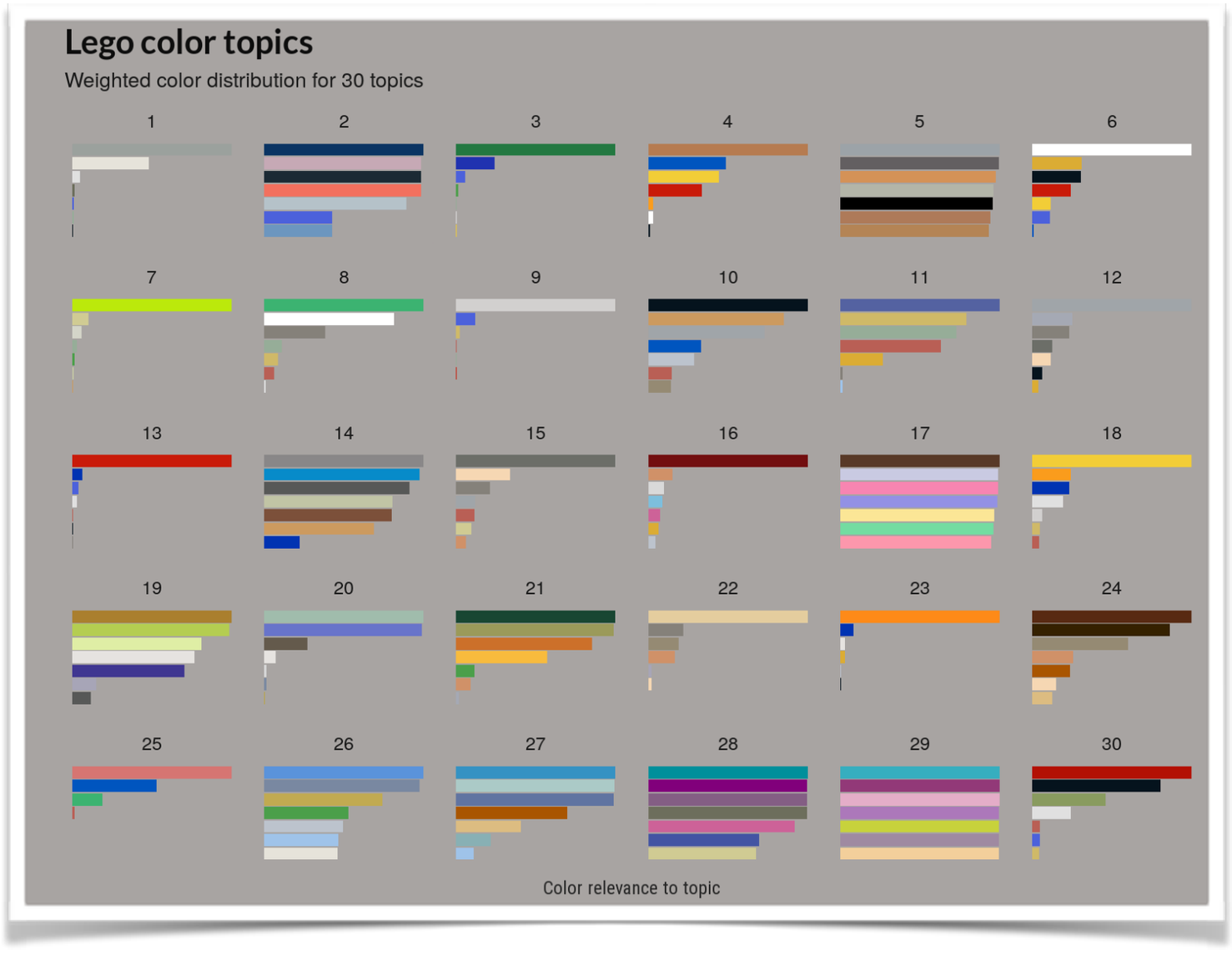
Having said that, the proposals are eminently sensible and certainly they are worth completing as soon as we can. I would just like to emphasise two of them. One is the vast potential that is there in the use of the internet and e-commerce. Too many businesses do not trade outside their own Member State, never mind the European Union: there is a challenge for us. The other is the recognition of qualifications across the European Union.





Other Applications

Topic models have also been applied to discover the underlying patterns across a range of different non-textual datasets.



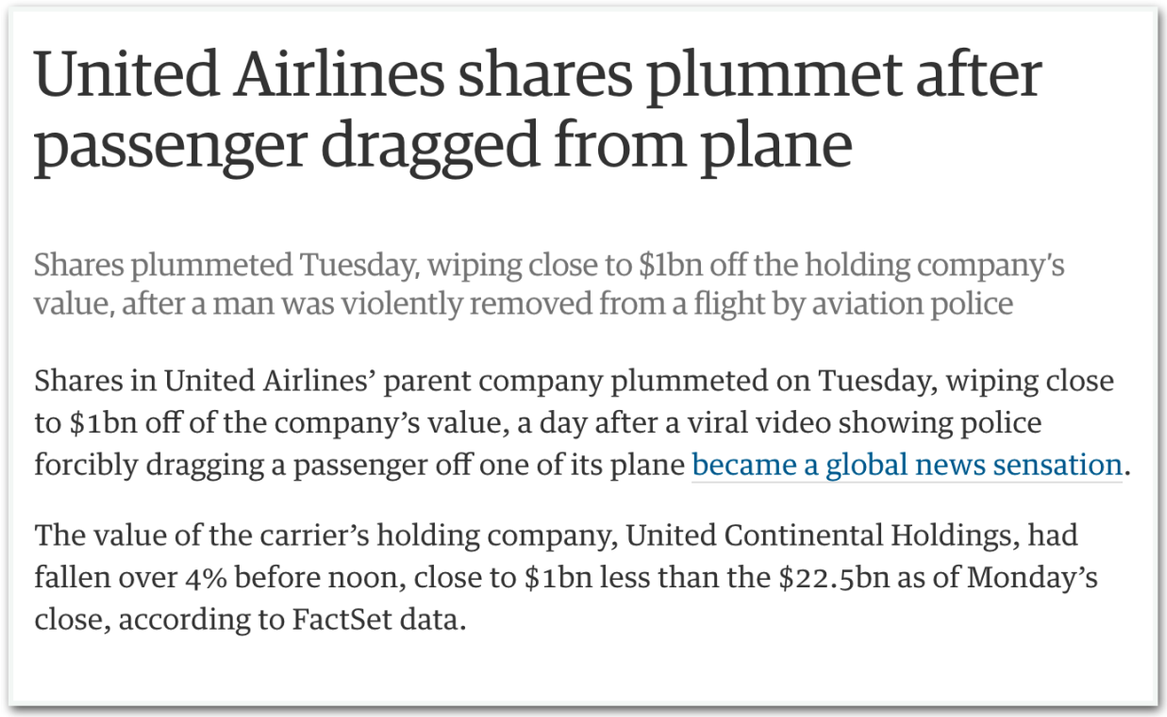
LEGO colour themes as topic models

https://nateaff.com/2017/09/11/lego-topic-models

Working with Text Data

Working with Text Data

Most text data arrives in an unstructured form without any pre-defined organisation or format, beyond natural language. The vocabulary, formatting, and quality of the text can vary significantly.

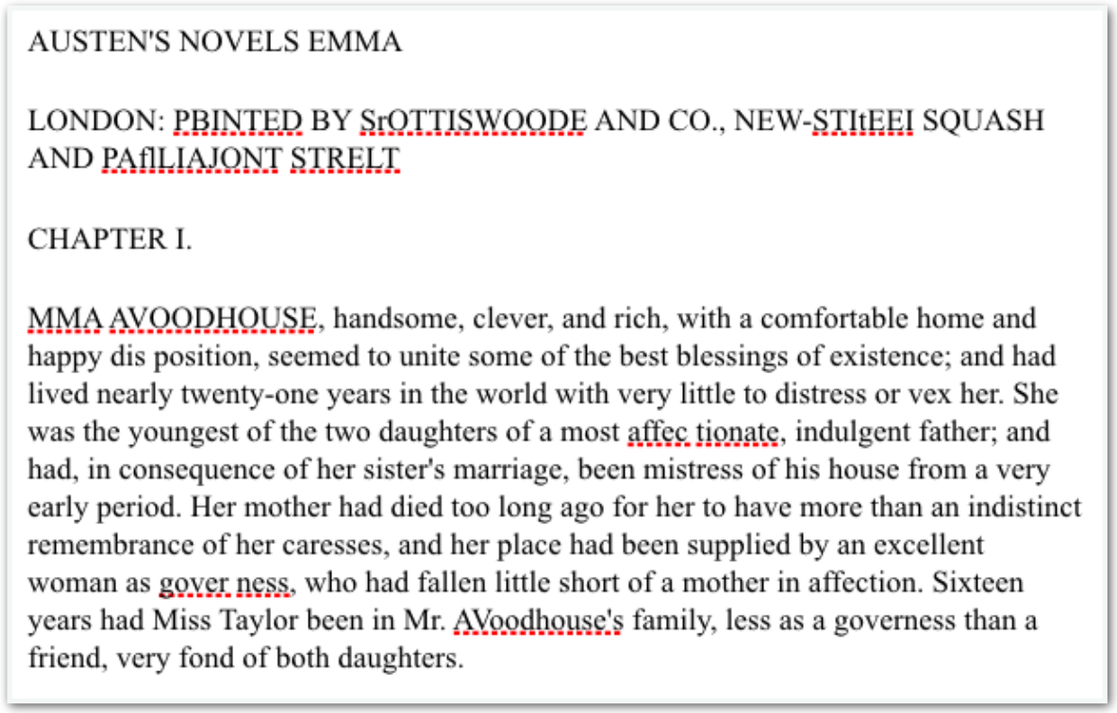


United Airlines shares plummet after passenger dragged from plane

Shares plummeted Tuesday, wiping close to $1bn off the holding company's value, after a man was violently removed from a flight by aviation police

Shares in United Airlines' parent company plummeted on Tuesday, wiping close to $1bn off of the company's value, a day after a viral video showing police forcibly dragging a passenger off one of its plane became a global news sensation.

The value of the carrier's holding company, United Continental Holdings, had fallen over 4% before noon, close to $1bn less than the $22.5bn as of Monday's close, according to FactSet data.



AUSTEN'S NOVELS EMMA

LONDON: PBINTED BY SrOTTISWOODE AND CO., NEW-STItEEI SQUASH AND PAflLIAJONT STRELT

CHAPTER I.

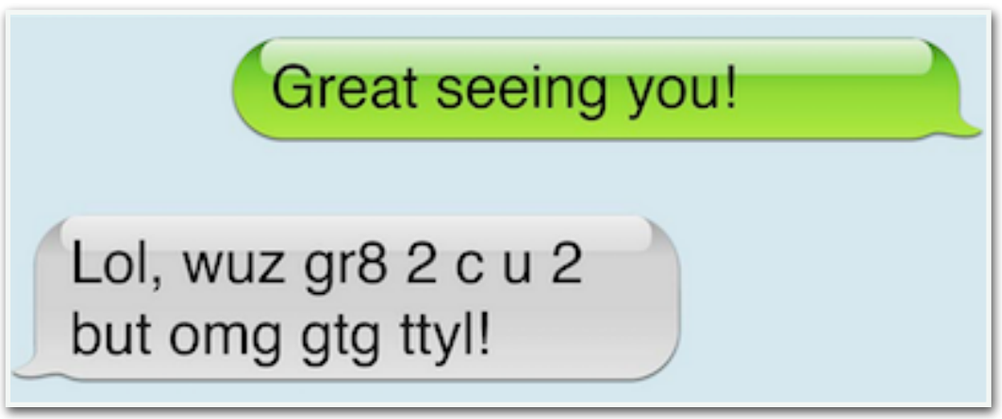
MMA AVOODHOUSE, handsome, clever, and rich, with a comfortable home and happy dis position, seemed to unite some of the best blessings of existence; and had lived nearly twenty-one years in the world with very little to distress or vex her. She was the youngest of the two daughters of a most affec tionate, indulgent father; and had, in consequence of her sister's marriage, been mistress of his house from a very early period. Her mother had died too long ago for her to have more than an indistinct remembrance of her caresses, and her place had been supplied by an excellent woman as gover ness, who had fallen little short of a mother in affection. Sixteen years had Miss Taylor been in Mr. AVoodhouse's family, less as a governess than a friend, very fond of both daughters.

Ai woz lyin on teh stairz n @vorvolak steppd on meh! She sez she noes seez meh buh ai woz dere! 😱😣

Translated from Indonesian by bing

Wrong translation?

Could not translate Tweet



Great seeing you!

Lol, wuz gr8 2 c u 2 but omg gtg ttyl!

Text Preprocessing

- Documents are textual, not numeric. The first step in analysing unstructured documents is tokenisation: split raw text into individual tokens, each corresponding to a single term.
- For English we typically split a text document based on whitespace. Punctuation symbols are often used to split too:

```
text = "Apple reveals new iPhone model"
text.split()
```

```
['Apple', 'reveals', 'new', 'iPhone', 'model']
```

- Splitting by whitespace will not work for some languages: e.g. Chinese, Japanese, Korean; German compound nouns.
- For some types of text content, certain characters can have a special significance:

Bag-of-Words Representation

- How can we go from tokens to numeric features?
- Bag-of-Words Model: Each document is represented by a vector in a m-dimensional coordinate space, where m is number of unique terms across all documents (the corpus vocabulary).

Document 1:
Forecasts cut as IMF issues warning

Document 2:
IMF and WBG meet to discuss economy

Document 3:
WBG issues 2016 growth warning

Example:
When we tokenise our corpus of 3 documents, we have a vocabulary of 14 distinct terms

```
vocab = set()
for doc in corpus:
     tokens = tokenize(doc)
     for tok in tokens:
          vocab.add(tok)
print(vocab)
```

```
{'2016', 'Forecasts', 'IMF', 'WBG', 'and',
'as', 'cut', 'discuss', 'economy', 'growth',
'issues', 'meet', 'to', 'warning'}
```

Bag-of-Words Representation

- Each document can be represented as a term vector, with an entry indicating the number of time a term appears in the document:

Document 1:
Forecasts cut as IMF issues warning

2016	Forecasts	IMF	WBG	and	as	cut	discuss	economy	growth	issues	meet	to	warning
0	1	1	0	0	1	1	0	0	0	1	0	0	1

- By transforming all documents in this way, and stacking them in rows, we create a full document-term matrix:

Document 2:
IMF and WBG meet to discuss economy

Document 3:
2016: WBG issues 2016 growth warning

2016	Forecasts	IMF	WBG	and	as	cut	discuss	economy	growth	issues	meet	to	warning
0	1	1	0	0	1	1	0	0	0	1	0	0	1
0	0	1	1	1	0	0	1	1	0	0	1	1	0
2	0	0	1	0	0	0	0	0	1	1	0	0	1

3 Documents x 14 Terms

Bag-of-Words in Scikit-learn

- Scikit-learn includes functionality to easily transform a collection of strings containing documents into a document-term matrix.

  Our input, documents, is a list of strings. Each string is a separate document.

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
A = vectorizer.fit_transform(documents)
```

  Our output, **A**, is a sparse NumPy 2D array with rows corresponding to documents and columns corresponding to terms.

- Once the matrix has been created, we can access the list of all terms and an associated dictionary (`vocabulary_`) which maps each unique term to a corresponding column in the matrix.

```
terms = vectorizer.get_feature_names()
len(terms)
```

```
3288
```

How many terms in the vocabulary?

```
vocab = vectorizer.vocabulary_
vocab["world"]
```

```
3246
```

Which column corresponds to a term?

Further Text Preprocessing

- The number of terms used to represent documents is often reduced by applying a number of simple preprocessing techniques before building a document-term matrix:
  - Minimum term length: Exclude terms of length < 2
  - Case conversion: Converting all terms to lowercase.
  - Stop-word filtering: Remove terms that appear on a pre-defined "blacklist" of terms that are highly frequent and do not convey useful information (e.g. `and, the, while`)
  - Minimum frequency filtering: Remove all terms that appear in very few documents.
  - Maximum frequency filtering: Remove all terms that appear in a very large number of documents.
  - Stemming: Process by which endings are removed from terms in order to remove things like tense or plurals:
    e.g. `compute, computing, computer = comput`

Further Text Preprocessing

- Further preprocessing steps can be applied directly using the CountVectorizer class by passing appropriate parameters - e.g.:

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(
    stop_words=custom_list,
    min_df=20,
    max_df=1000,
    lowercase=False,
    ngram_range=2)
A = vectorizer.fit_transform(documents)
```

Parameter	Explanation
`stop_words=custom_list`	Pass in a custom list containing terms to filter.
`min_df=20`	Filter those terms that appear in < 20 documents.
`max_df=1000`	Filter those terms that appear in > 1000 documents.
`lowercase=False`	Do not convert text to lowercase. Default is True.
`ngram_range=2`	Include phrases of length 2, instead of just single words.

Term Weighting

- As well as including or excluding terms, we can improve the usefulness of the document-term matrix by giving higher weights to more "important" terms.
- TF-IDF: Common approach for weighting the score for a term in a document. Consists of two parts:
  - Term Frequency (TF): Number of times a given term appears in a single document.
  - Inverse Document Frequency (IDF): Function of total number of distinct documents containing a term. Effect is to penalise common terms that appear in almost every document.

$$w(t, D) = tf(t, d) \times (log(\frac{n}{df(t)}) + 1)$$

n = total number of documents

- Example: the term "cat" appears in a given document 3 times and appears 50 times overall in a corpus of 1000 documents:

$$w(\text{cat}, D) = 3 \times (log(\frac{1000}{50}) + 1) = 11.987$$

Term Weighting in Scikit-learn

- A similar vectorisation approach can be used in Scikit-learn to produce a TF-IDF normalised document-term matrix:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
A = vectorizer.fit_transform(documents)
```

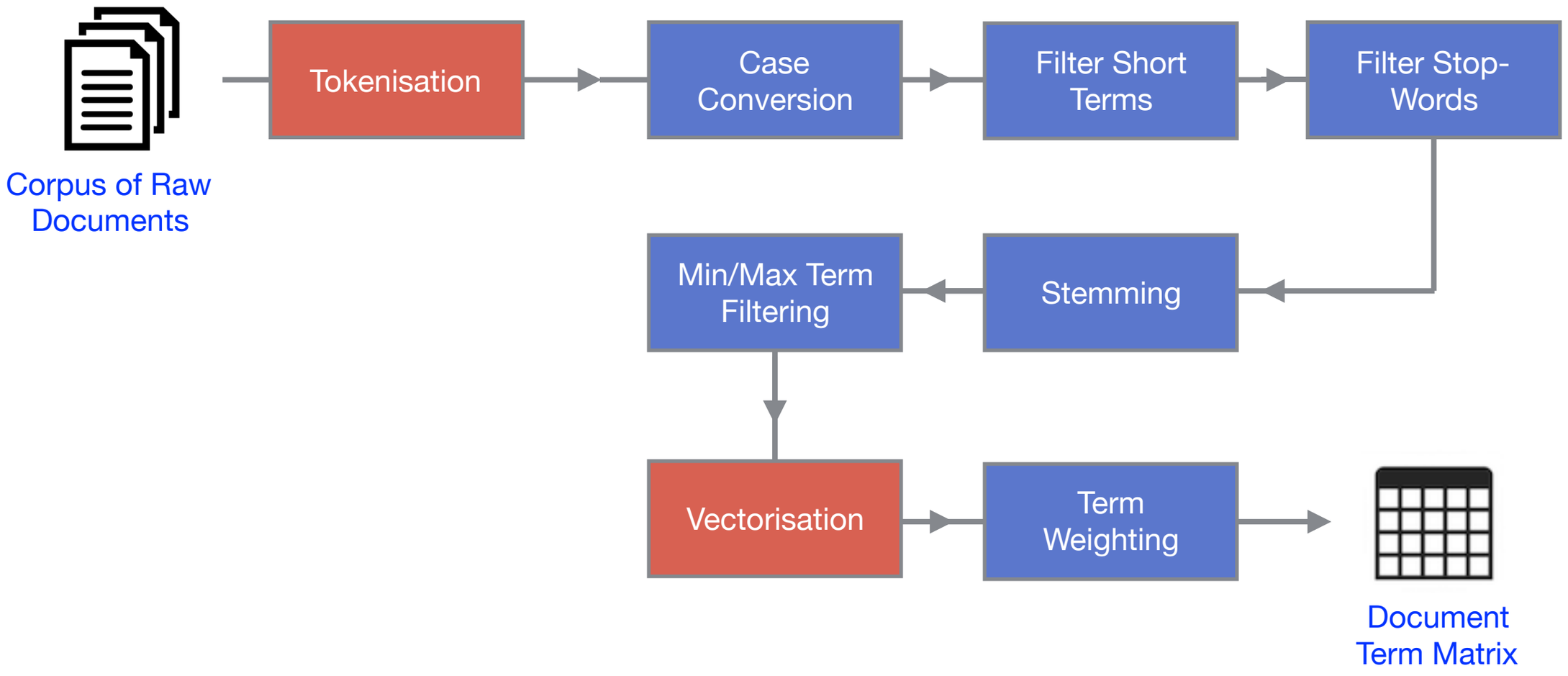
The output, **A**, is a sparse NumPy array where the entries are all TF-IDF normalised.

- Again we can perform additional preprocessing steps by passing the appropriate parameter values to TfidfVectorizer:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(
    stop_words=custom_list,
    min_df=20,
    max_df=1000,
    lowercase=False,
    ngram_range=2)
A = vectorizer.fit_transform(documents)
```

Text Preprocessing Pipeline

- Typical text preprocessing steps for a document corpus...



- Note: Stemming is not included in scikit-learn. See NLTK package.
- Once we have our document-term matrix, we are ready to apply machine learning algorithms to explore the data.

Topic Modelling

Topic Modelling Algorithms

Various different methods for topic modelling have been proposed. Two general approaches are popular:

1. Probabilistic approaches

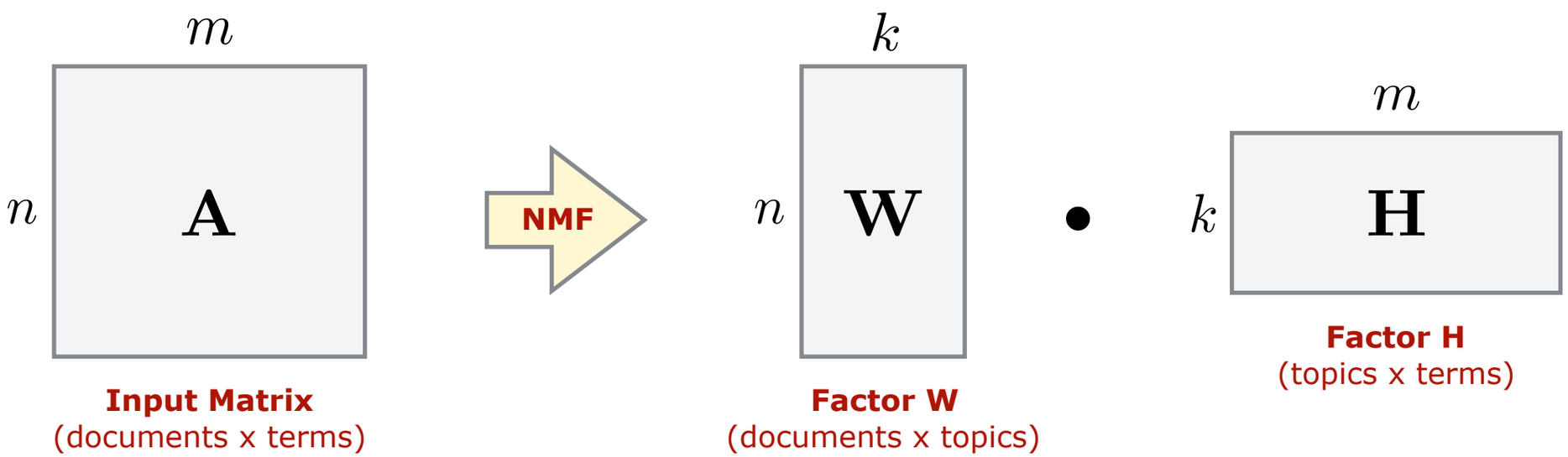
- View each document as a mixture of a small number of topics.
- Certain words appearing in documents will have a high probability for one topic but not other topics.
- Words and documents receive probability scores for each topic.
- e.g. Latent Dirichlet Allocation (LDA) (Blei et al, 2003).

2. Matrix factorisation approaches

- Apply methods from linear algebra to decompose a single matrix (e.g. document-term matrix) into a set of smaller matrices.
- For text data, we can interpret these matrices as a topic model.
- e.g. Non-negative Matrix Factorisation (NMF) (Lee & Seung, 1999).

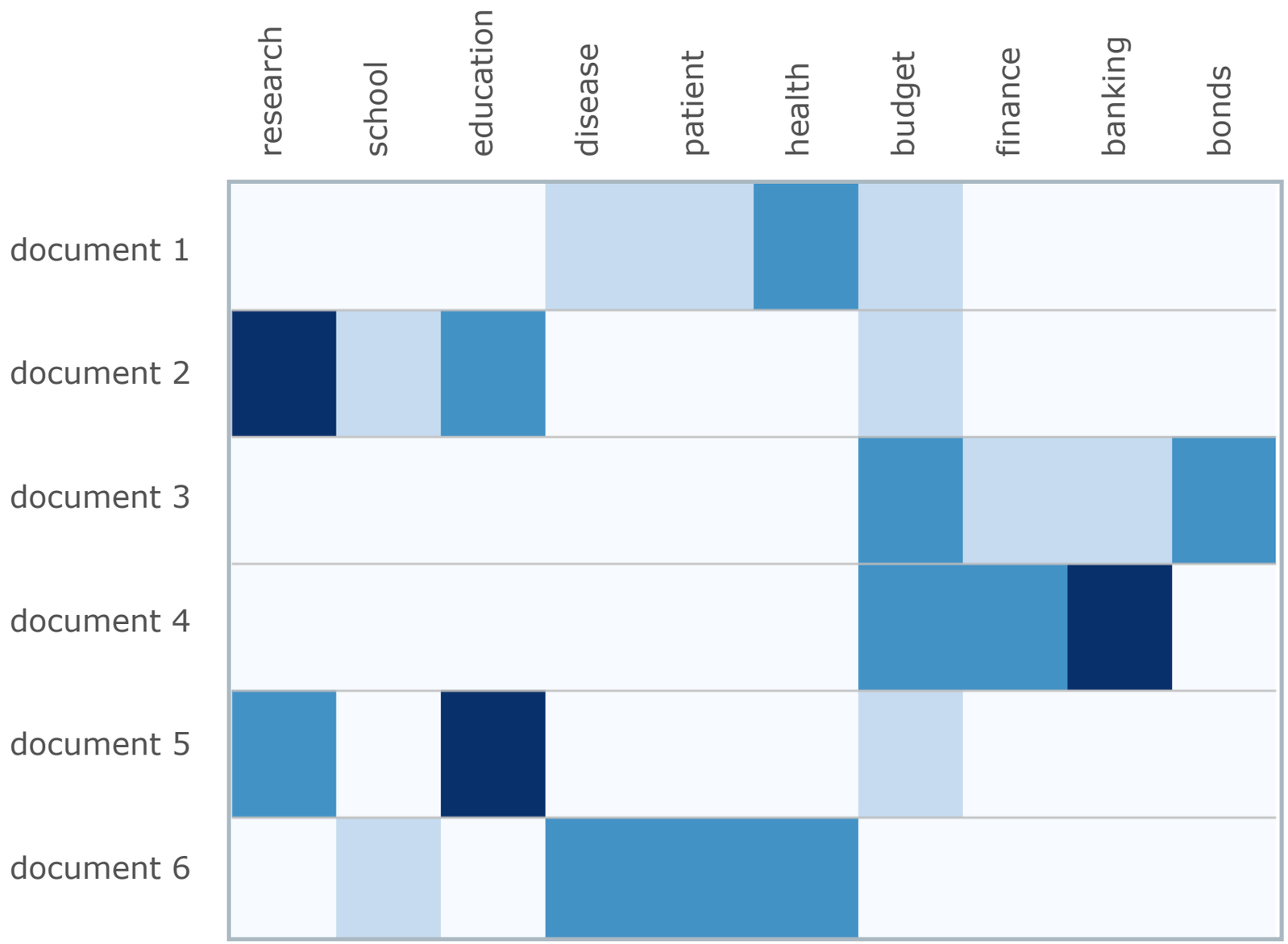
Non-negative Matrix Factorisation

- Non-negative Matrix Factorisation (NMF): Family of linear algebra algorithms for identifying the latent structure in data represented as a non-negative matrix (Lee & Seung, 1999).
- NMF can be applied for topic modeling, where the input is a document-term matrix, typically TF-IDF normalised.
- **Input:** Document-term matrix **A**; Number of topics k.
- **Output:** Two k-dimensional factors **W** and **H** approximating **A**.



Example: NMF Topic Modelling

Apply NMF topic modelling to a small document-term matrix **A** representing a corpus of 6 documents, to generate $k=3$ topics…



6 Documents x 10 Terms

Example: NMF Topic Modelling

Factor **W**
Weights for 6 documents
relative to 3 topics

	Topic 1	Topic 2	Topic 3
document 1	0.0	1.0	1.0
document 2	0.0	0.0	1.0
document 3	0.7	0.0	0.0
document 4	0.7	0.0	0.0
document 5	0.0	0.0	1.0
document 6	0.0	1.0	1.0

6 Rows x 3 Columns

Factor **H**
Weights for 10 terms
relative to 3 topics

	Topic 1	Topic 2	Topic 3
research	0.0	0.0	1.0
school	0.0	0.1	0.1
education	0.0	0.0	1.0
disease	0.0	0.6	0.0
patient	0.0	0.7	0.0
health	0.0	1.0	0.0
budget	0.3	0.1	0.2
finance	0.6	0.0	0.0
banking	0.7	0.0	0.0
bonds	0.3	0.0	0.0

10 Rows x 3 Columns

Applying NMF in Scikit-learn

- Scikit-learn includes a fast implementation of NMF.
- By default, the values in factors **W** and **H** are given random initial values. The key required input parameter is the number of topics (components) k:

```
from sklearn import decomposition
model = decomposition.NMF(n_components=k)
W = model.fit_transform( A )
H = model.components_
```

Apply NMF to document-term matrix **A**, extract the resulting factors **W** and **H**

- When using random initialisation, the results can be different every time NMF is applied to the same data. More reliable results can be obtained if you initialise with SVD (Belford et al, 2018).

```
from sklearn import decomposition
model = decomposition.NMF(n_components=k, init="nndsvd")
W = model.fit_transform( A )
H = model.components_
```

Applying NMF in Scikit-learn

- The **H** factor contains term weights relative to each of the k topics. Each row corresponds to a topic, and each column corresponds to a unique term in the corpus vocabulary.
- Sorting the values in each row gives us a ranking of terms - the descriptor of each topic.

```
import numpy as np
top_indices = np.argsort( H[topic_index,:] )[::-1]
top_terms = []
for term_index in top_indices[0:top]:
    top_terms.append( terms[term_index] )
```

For each topic, sort the row indices in reverse, then get the terms for the top indices.

Repeat for all topics to get the full set of descriptors:

```
Topic 01: eu, brexit, uk, britain, referendum, leave, vote, european, cameron, labour
Topic 02: trump, clinton, republican, donald, campaign, president, hillary, cruz, sanders, election
Topic 03: film, films, movie, star, hollywood, director, actor, story, drama, women
Topic 04: league, season, leicester, goal, premier, united, city, liverpool, game, ball
Topic 05: bank, banks, banking, financial, rbs, customers, shares, deutsche, barclays, lloyds
Topic 06: health, nhs, care, patients, mental, doctors, hospital, people, services, junior
Topic 07: album, music, band, song, pop, songs, rock, love, sound, bowie
Topic 08: internet, facebook, online, people, twitter, media, users, google, company, amazon
```

Applying NMF in Scikit-learn

- The **W** factor contains document membership weights across the k topics. Each row corresponds to a different document, and each column corresponds to a topic.
- Sorting the values gives us a ranking of the most relevant documents for each topic.

```
top_indices = np.argsort( W[:,topic_index] )[::-1]
top_documents = []
for doc_index in top_indices[0:top]:
    top_documents.append( documents[doc_index] )
```

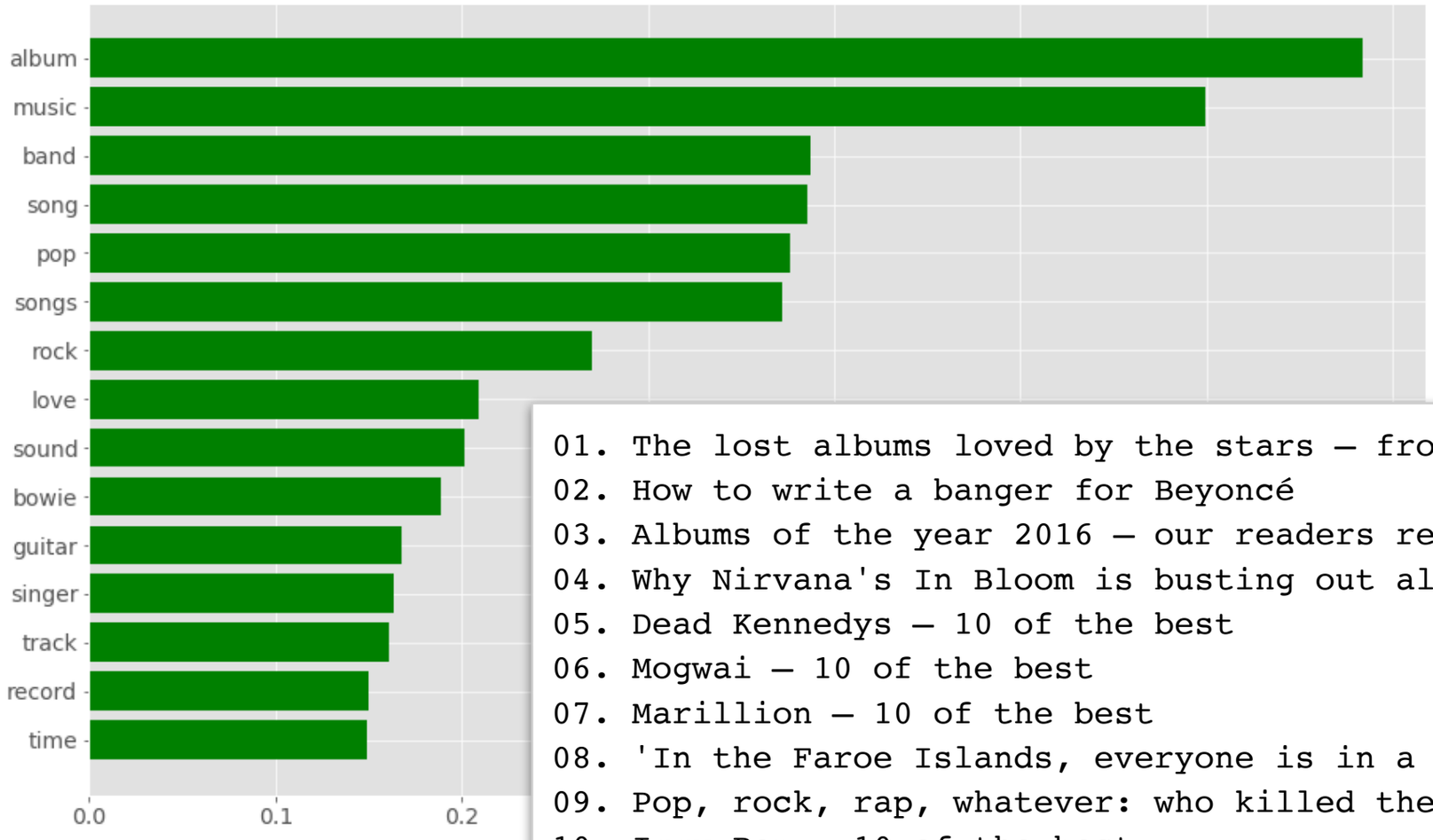
For each topic, sort column indices in reverse, then get the documents for the top indices.

The top documents for a topic might be summarised using titles or snippets:

```
01. Donald Trump: money raised by Hillary Clinton is 'blood money'
02. Second US presidential debate – as it happened
03. Trump campaign reportedly vetting Christie, Gingrich as potential running mates
04. Donald Trump hits delegate count needed for Republican nomination
05. Trump: 'Had I been president, Capt Khan would be alive today'
06. Clinton seizes on Trump tweets for day of campaigning in Florida
07. Melania Trump defends husband's 'boy talk' in CNN interview
08. Hillary Clinton: 'I'm sick of the Sanders campaign's lies'
09. Donald Trump at the White House: Obama reports 'excellent conversation'
10. Donald Trump: Hillary Clinton has 'no right to be running'
```

Applying NMF in Scikit-learn

```
Topic 01: eu, brexit, uk, britain, referendum, leave, vote, european, cameron, labour
Topic 02: trump, clinton, republican, donald, campaign, president, hillary, cruz, sanders, election
Topic 03: film, films, movie, star, hollywood, director, actor, story, drama, women
Topic 04: league, season, leicester, goal, premier, united, city, liverpool, game, ball
Topic 05: bank, banks, banking, financial, rbs, customers, shares, deutsche, barclays, lloyds
Topic 06: health, nhs, care, patients, mental, doctors, hospital, people, services, junior
Topic 07: album, music, band, song, pop, songs, rock, love, sound, bowie
Topic 08: internet, facebook, online, people, twitter, media, users, google, company, amazon
```




```
01. The lost albums loved by the stars — from ecstatic gospel to Italian prog
02. How to write a banger for Beyoncé
03. Albums of the year 2016 — our readers respond
04. Why Nirvana's In Bloom is busting out all over
05. Dead Kennedys — 10 of the best
06. Mogwai — 10 of the best
07. Marillion — 10 of the best
08. 'In the Faroe Islands, everyone is in a band'
09. Pop, rock, rap, whatever: who killed the music genre?
10. Iggy Pop — 10 of the best
```

Parameter Selection

- The key parameter selection decision for topic modelling involves choosing the number of topics k.
- Common approach: Measure and compare the topic coherence of models generated for different values of k.
- Topic coherence: The extent to which the top terms representing a topic (i.e. the topic descriptor) are semantically related, relative to some "background corpus".
- A variety of different measures exist for measuring coherence e.g. NPMI, UMass, TC-W2V etc. (O'Callaghan et al, 2015).

Rank	Term
1	port
2	sea
3	maritime
4	naval
5	vessel

"High coherence topic"

Rank	Term
1	agriculture
2	farmer
3	beef
4	food
5	dairy

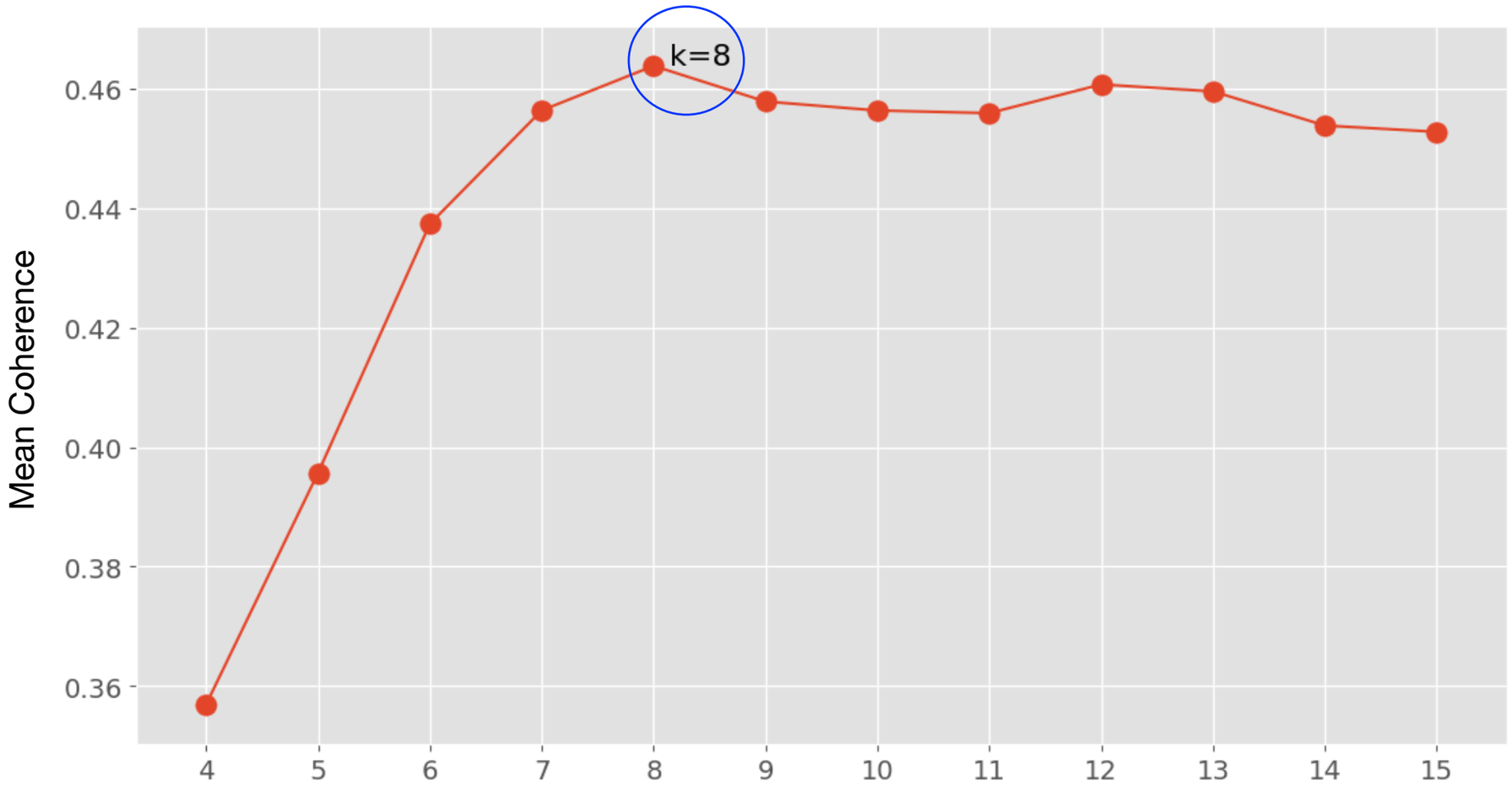
"High coherence topic"

Rank	Term
1	farmer
2	naval
3	dairy
4	maritime
5	nuclear

"Low coherence topic"

Parameter Selection

- **Typical approach for parameter selection:**
  1. Apply NMF for a "sensible" range $k=[kmin,kmax]$.
  2. Calculate mean coherence of the topics produced for each k, relative to the overall corpus or a related background corpus.
  3. Select the value of k giving the highest mean coherence.



Practical Issues

- **Preprocessing**
  - Stop-word filtering often has a major impact.
  - TF-IDF often leads to more useful topics than raw frequencies.
- **Initialisation**
  - Random initialisation of both NMF and LDA can lead to unstable results, particularly for larger datasets.
- **Scalability**
  - NMF typically more scalable than LDA, but running times can increase considerably as number of topics k increases.
- **Parameter Selection**
  - In many cases, there can be several "good" values of k.
  - Choice of coherence measure can produce different results.
- **Interpretation**
  - Topic models reflect the structure of the data available. Best used carefully as an exploratory tool to aid human interpretation.

Any Questions?

derek.greene@ucd.ie

@derekgreene

https://github.com/derekgreene/topic-model-tutorial

References

- Pedregosa, F., et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12. Oct (2011): 2825-2830.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.
- Blei, D. M. (2012). Probabilistic topic models. Communications of the ACM, 55(4), 77-84.
- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by Non-negative Matrix Factorization. Nature, 401(6755), 788.
- Belford, M., Mac Namee, B., & Greene, D. Stability of Topic Modeling via Matrix Factorization. Expert Systems with Applications, 2018.
- O'Callaghan, D., Greene, D., Carthy, J., & Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. Expert Systems with Applications, 42(13), 5645-5657.
- Greene, D., & Cross, J. P. (2017). Exploring the Political Agenda of the European Parliament Using a Dynamic Topic Modeling Approach. Political Analysis, 25(1), 77-94.
- Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.