42578 Advanced Business Analytics

Text Analytics

Francisco Pereira [camara@dtu.dk] and Stanislav Borysov

Text data

- **Language is one of the primary means of communicating**

- **Huge in size**
  - Google processes 5.13B queries/day (2013)
  - Twitter receives 340M tweets/day (2012)
  - Facebook has 2.5 PB of user data + 15 TB/day (4/2009)
  - eBay has 6.5 PB of user data + 50 TB/day (5/2009)
  - …

- **80% data is unstructured (IBM, 2010)**

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

What is “Text Mining”?

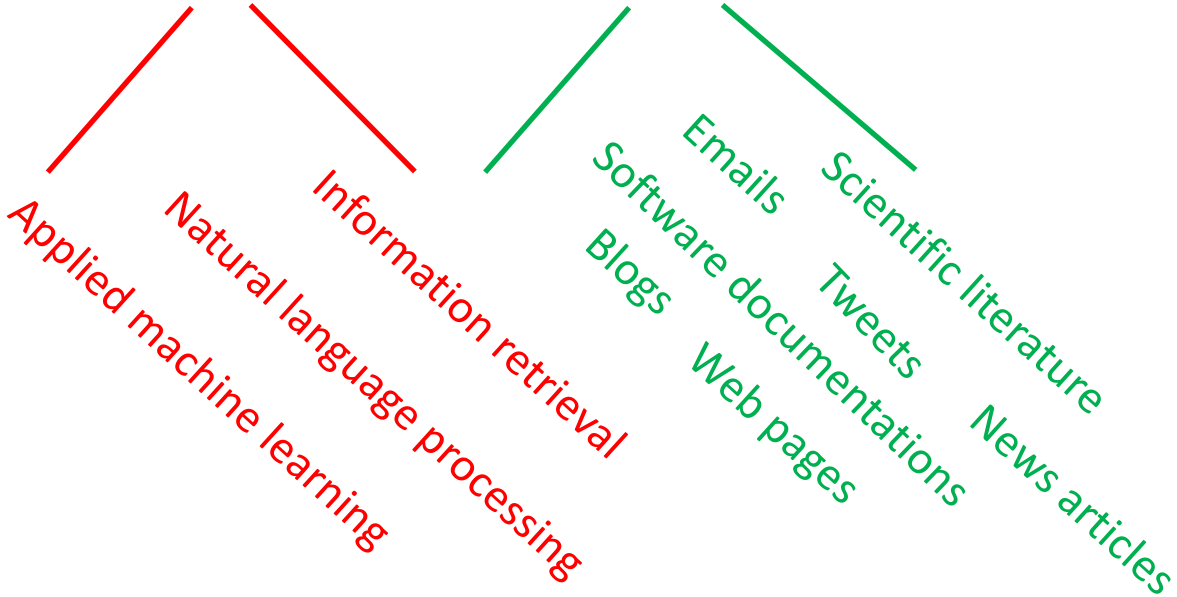
- “Text mining, also referred to as **text data mining**, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text.” - wikipedia

- “Another way to view text data mining is as a process of **exploratory** data analysis that leads to **heretofore unknown** information, or to answers for questions for which the answer is not currently known.” - Hearst, 1999

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

How to perform text mining?

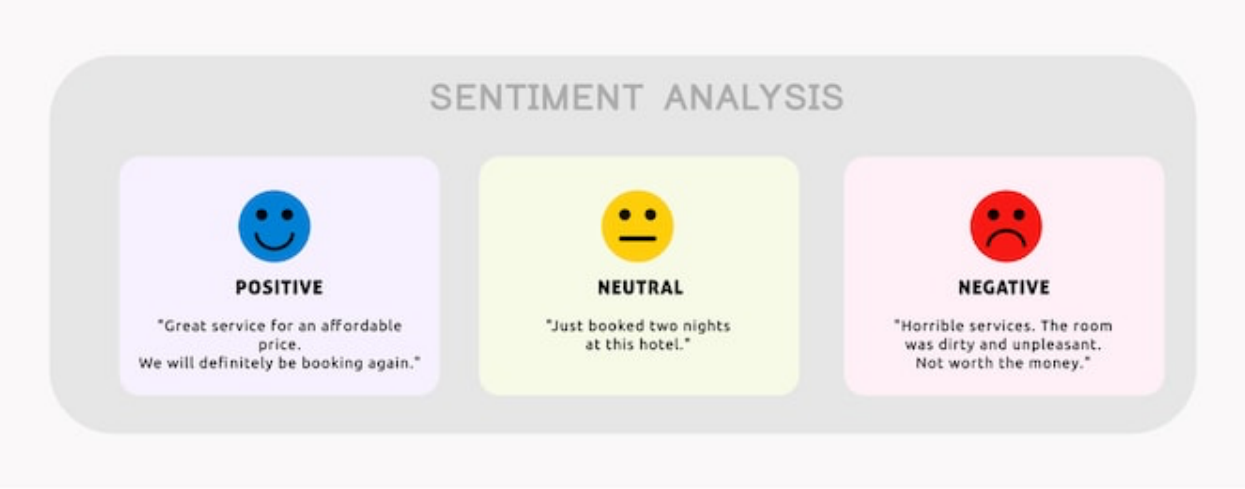
- Text Mining = Data Mining + Text Data



Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

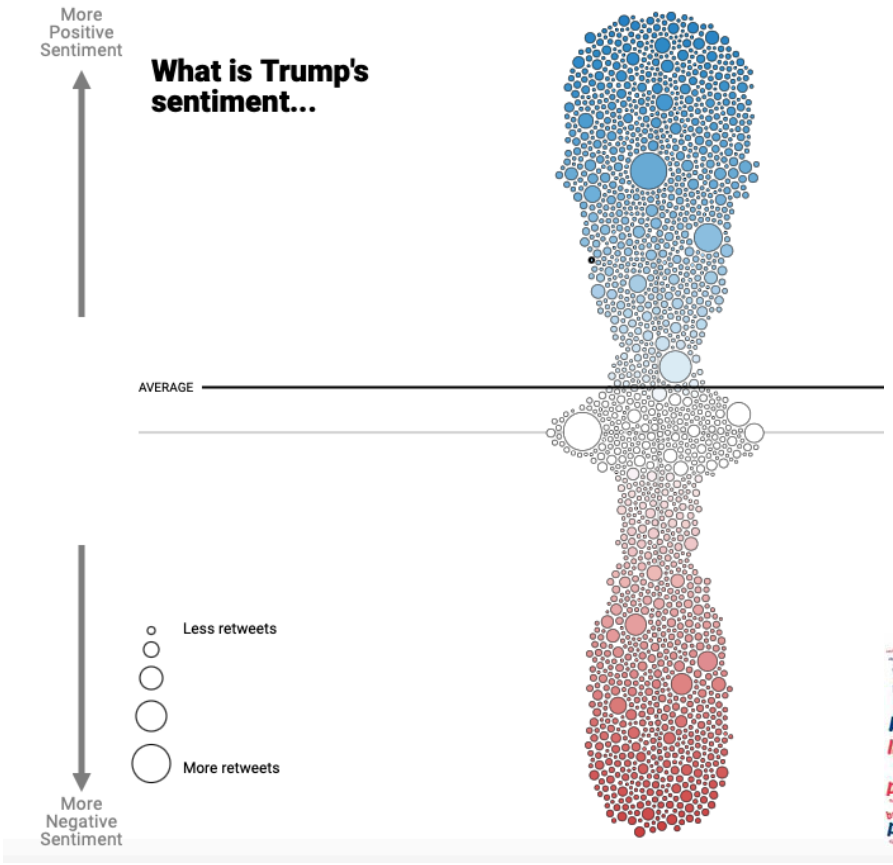
- Sentiment analysis

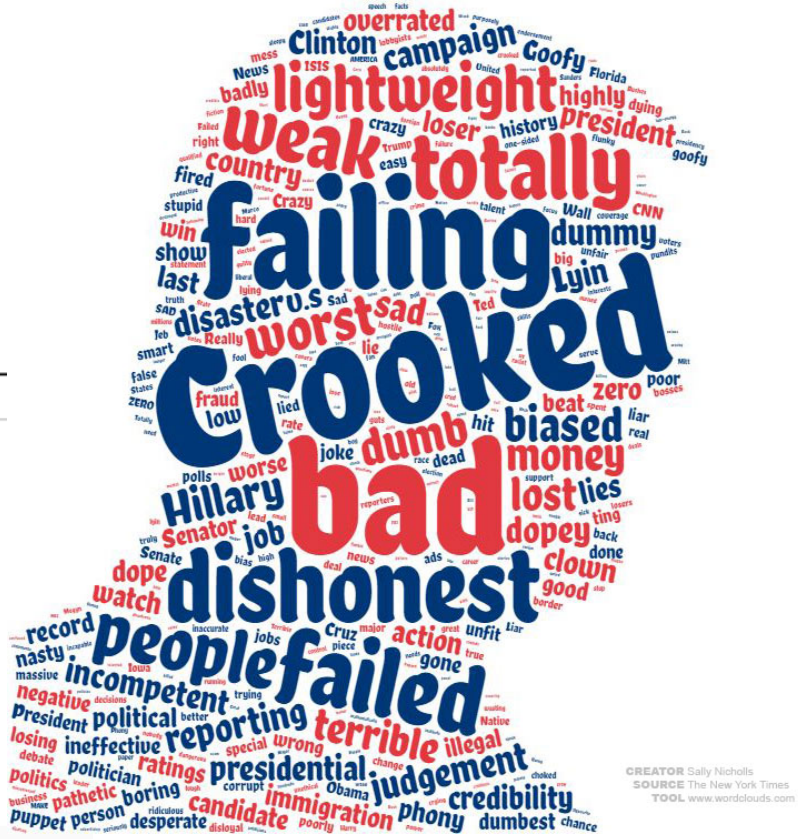


Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

- Sentiment analysis

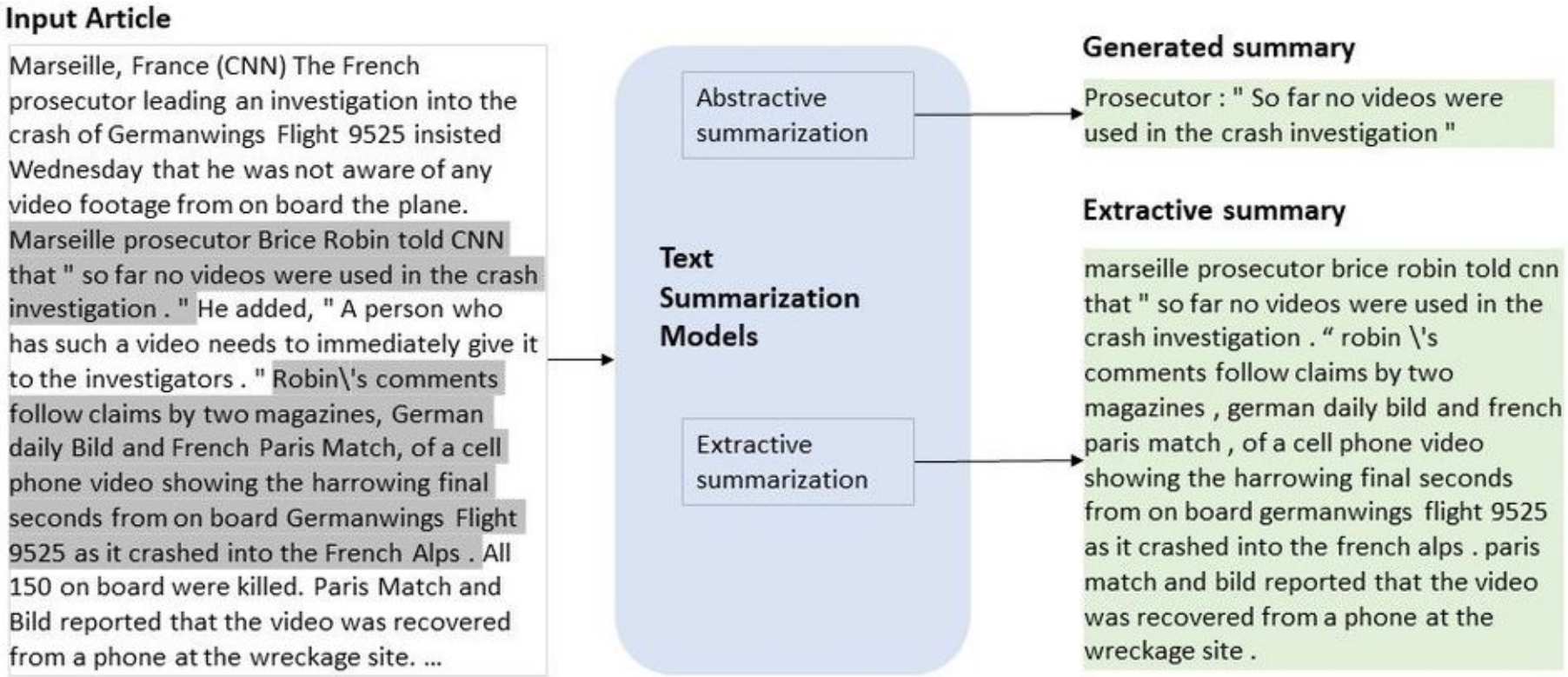




Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

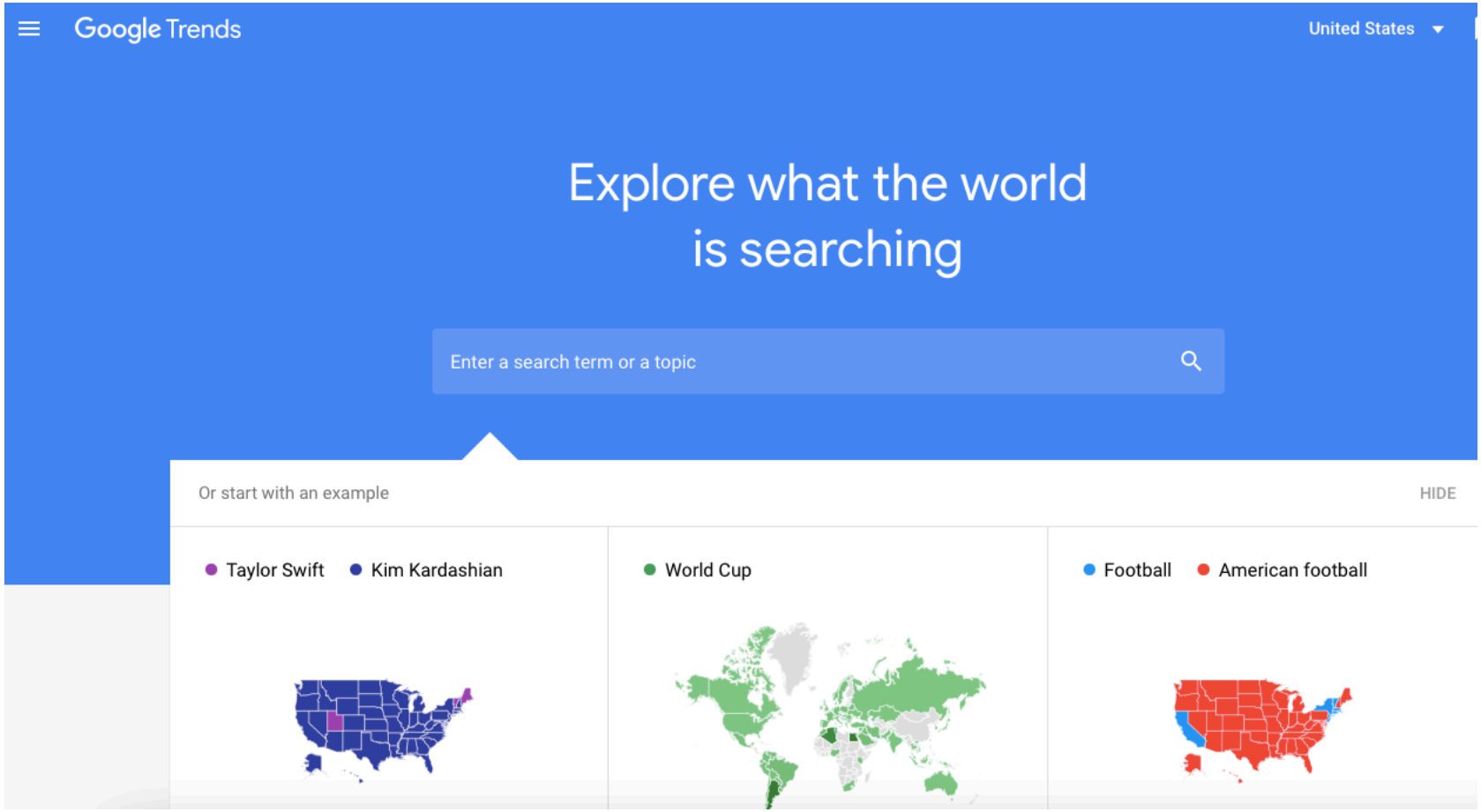
- Document summarization



Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

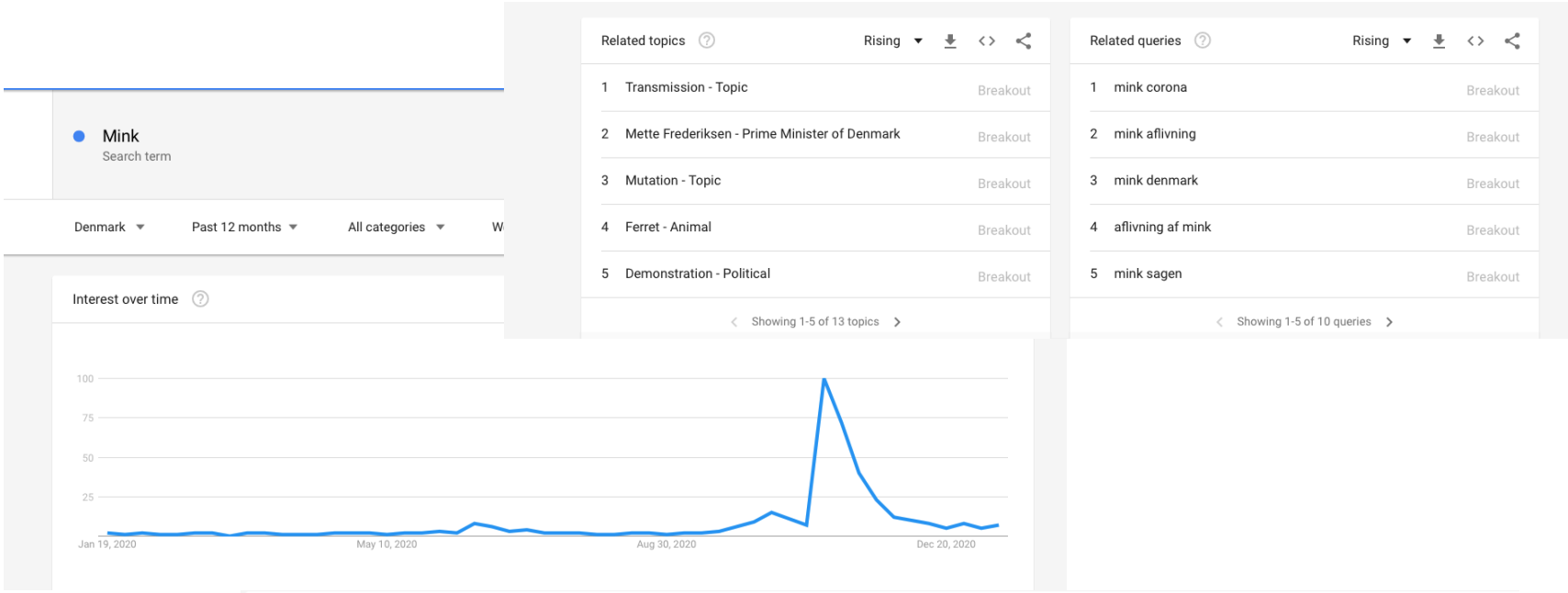
Text mining around us

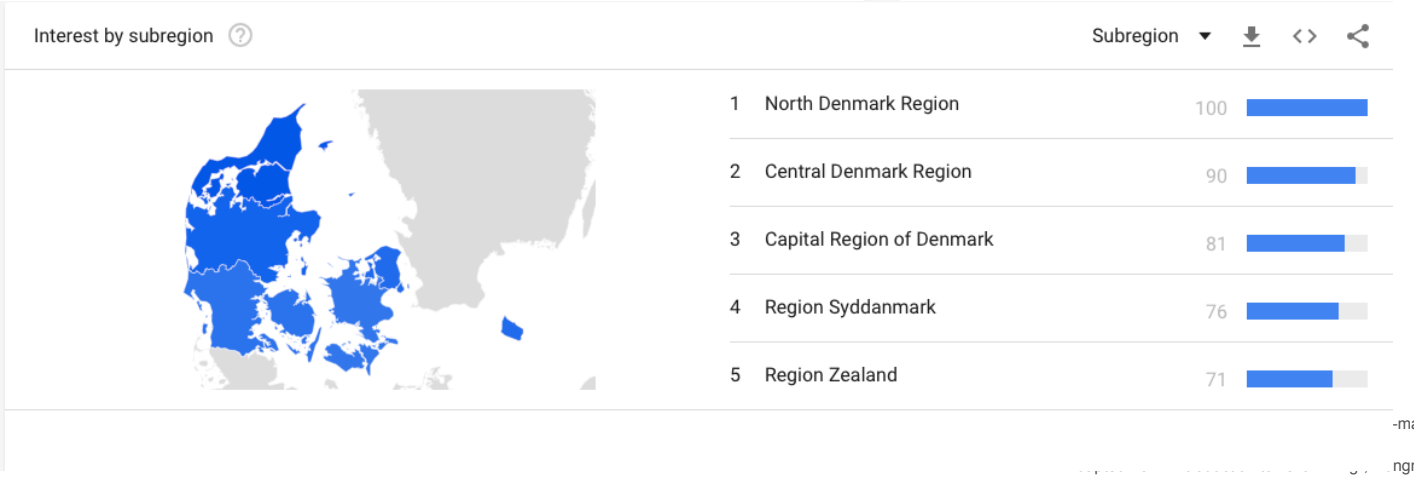
- Trends



Text mining around us

- Trends





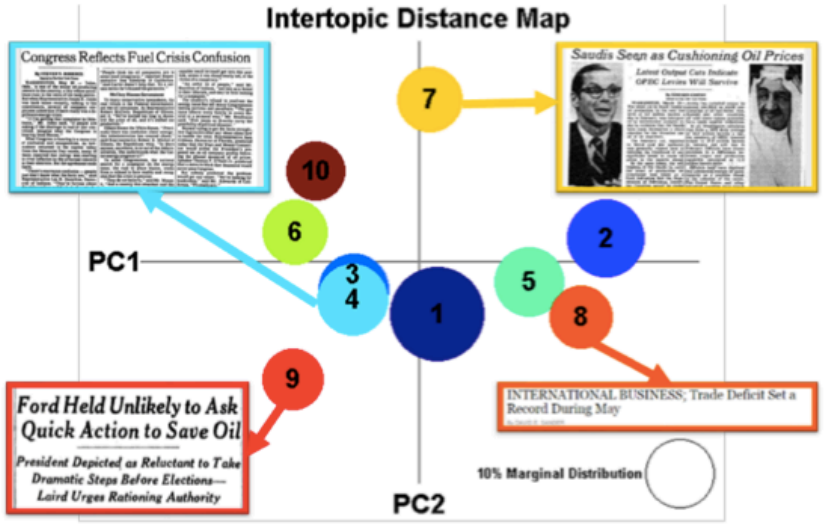
-machine-16abe0eb4181

ngning Wang (CS@UVa)

Text mining around us

- Text analytics in financial services

#	Word Distribution for Topics	Topic Interpretation	Color
1	1.6% company, 0.9% quarter, 0.5% profits, 0.4% industry	Corporate Finance	
2	1.2% dollar, 0.9% futures, 0.8% trade, 0.4% commodity	Commodity and US Currency	
3	1.2% economic, 0.9% Russia, 0.6% Mexico, 0.5% debt	World Economy	
4	2.2% energy, 1.1% tax, 1.0% gas, 0.6% bill, 0.5% congress	US Energy Policy	
5	1.9% economy, 1.8% rates, 1.7% growth, 1.6% interest	Emerging Economies	
6	1.2% Iraq, 1.1% Saudi, 0.7% Iran, 0.6% war, 0.3% military	Middle East Conflict	
7	3.1% OPEC, 1.8% production, 1.8% crude, 1.0% output	OPEC Production	
8	2.5% stocks, 2.0% market, 1.8% dow, 1.2% shares	Stock Market	
9	0.6% president, 0.3% America, 0.2% public, 0.2% election	US Elections and Politics	
10	1.0% countries, 0.7% world, 0.5% arab, 0.5% OPEC	World-OAPEC Relations	

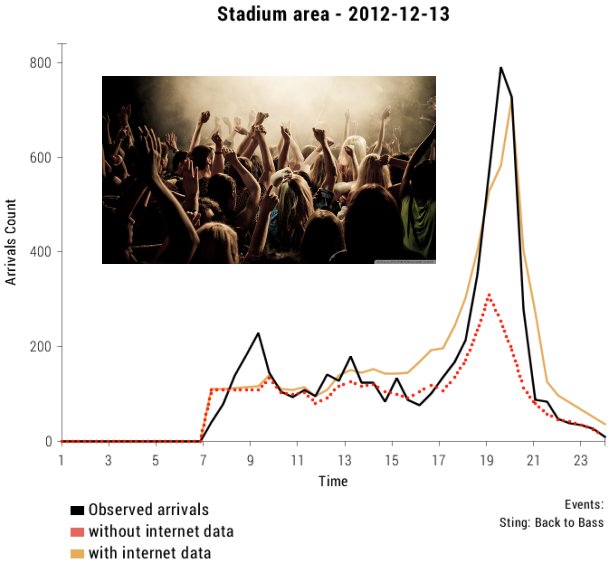


https://plot.ly/~suman.dna/4.embed

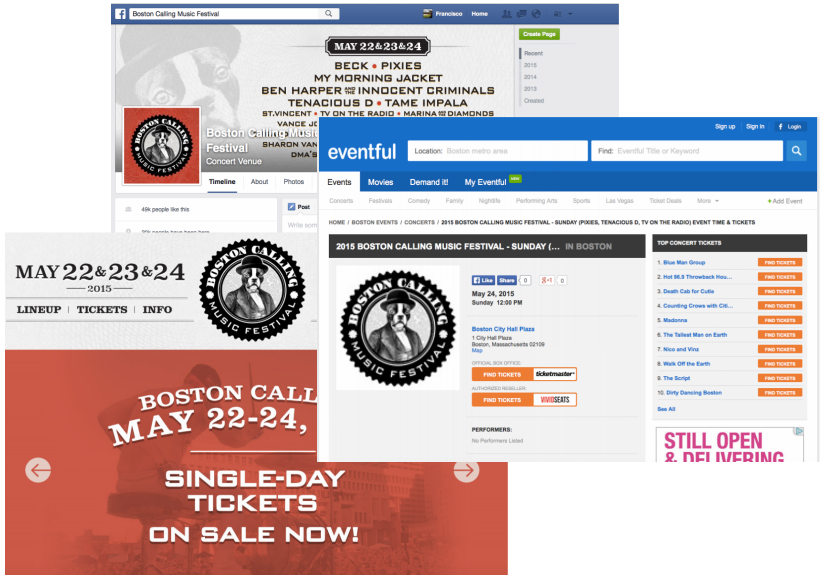
Adapted from "Introduction to Text Mining", Hongning Wang (CS@UVa)

Text mining around us

- Text analytics in transportation



+



Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Text mining around us

Text Mining has numerous applications in any industry	**Government** Detect fraudulent activity. Spot emerging trends and public concerns.	**Finance** Retention of current customer base using call center transcriptions or transcribed audio. Identification of potentially fraudulent activities.	**Insurance** Identify fraudulent claims. Track competitive intelligence. Brand management
Retail Identify the most profitable customers and the underlying reasons for their loyalty. Brand management	**Manufacturing** Reduce time to detect root cause of product issues. Identify trends in market segments.	**Telecommunications** Help prevent churn and suggest up-sell/cross-sell opportunities for individual customers.	**Life Sciences** Identify adverse events. Recommend appropriate research materials.

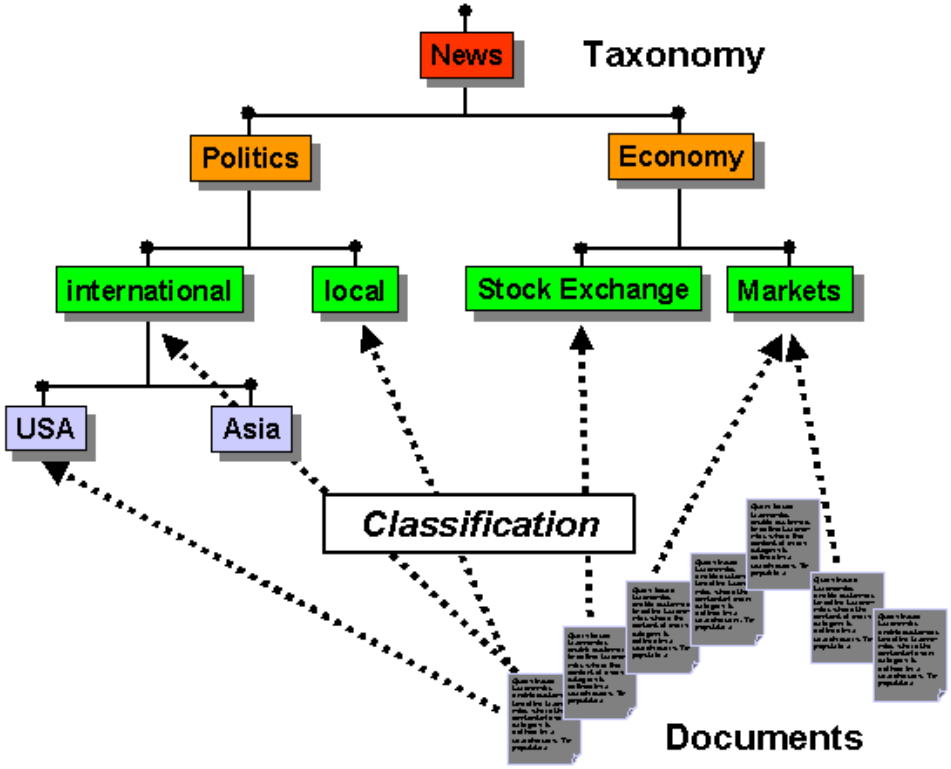
MS4252 Big Data Analytics, City University of Hong Kong

Some common text mining tasks

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Some common text mining tasks

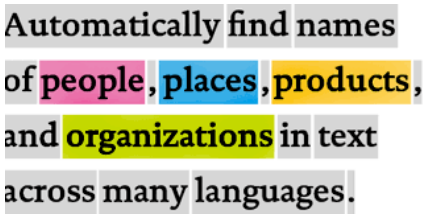
- Document categorization
  - Adding labels to a text corpus



Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Some common text mining tasks

- Named Entity Recognition

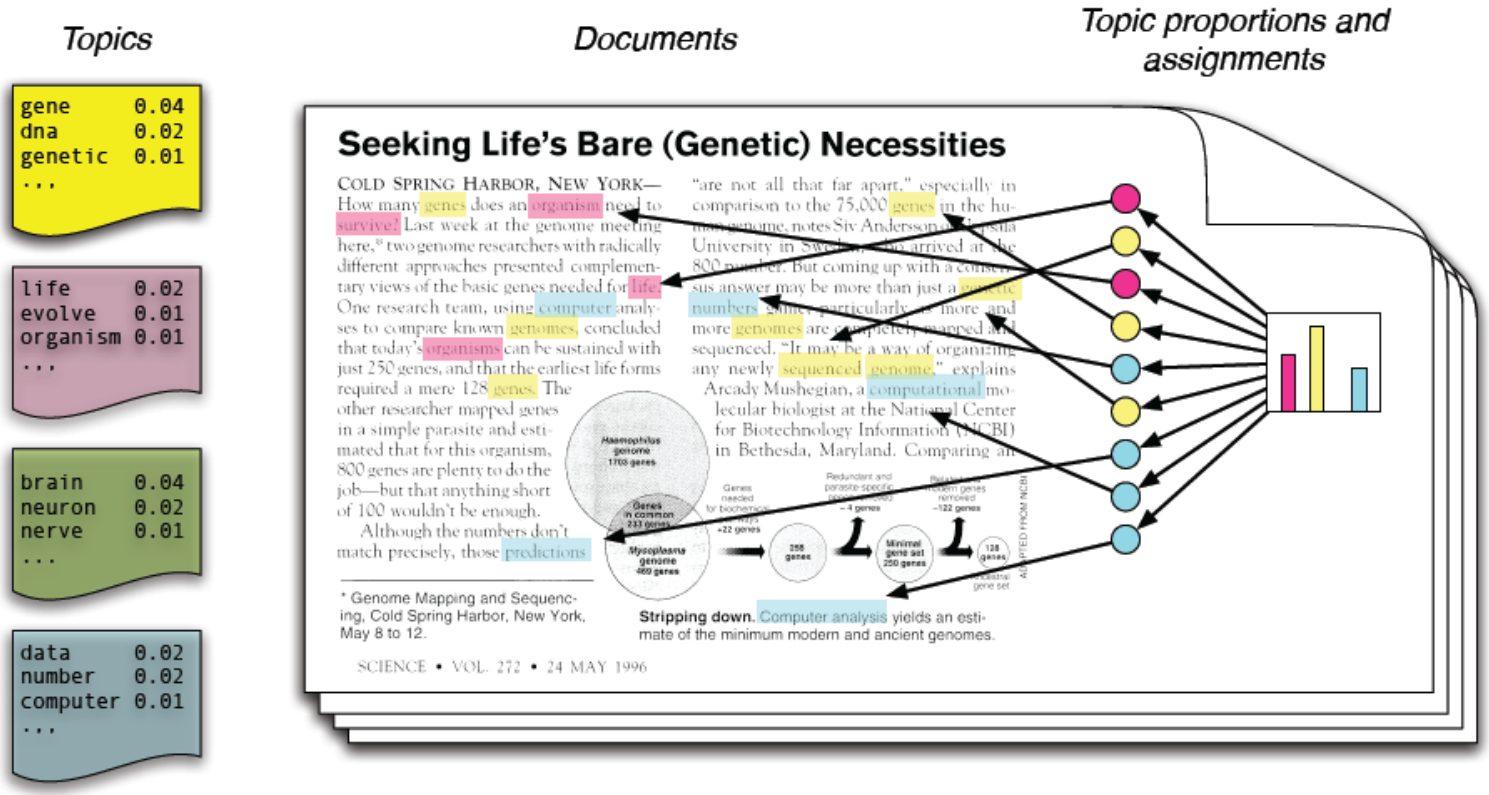


https://imanage.com/blog/named-entity-recognition-ravn-part-1/
https://wordlift.io/blog/en/entity/named-entity-recognition/

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Some common text mining tasks

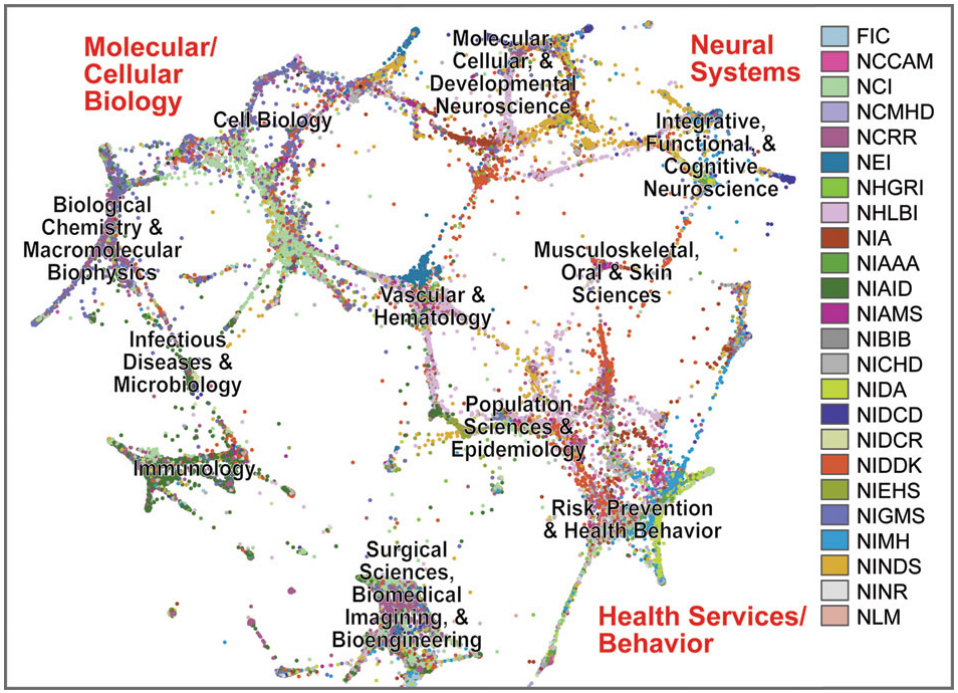
- Topic modeling
  - Identifying common themes/topics in a text corpus



Adapted from "Introduction to Text Mining", Hongning Wang (CS@UVa)

Some common text mining tasks

- Text clustering
  - Grouping common themes/topics in a text corpus

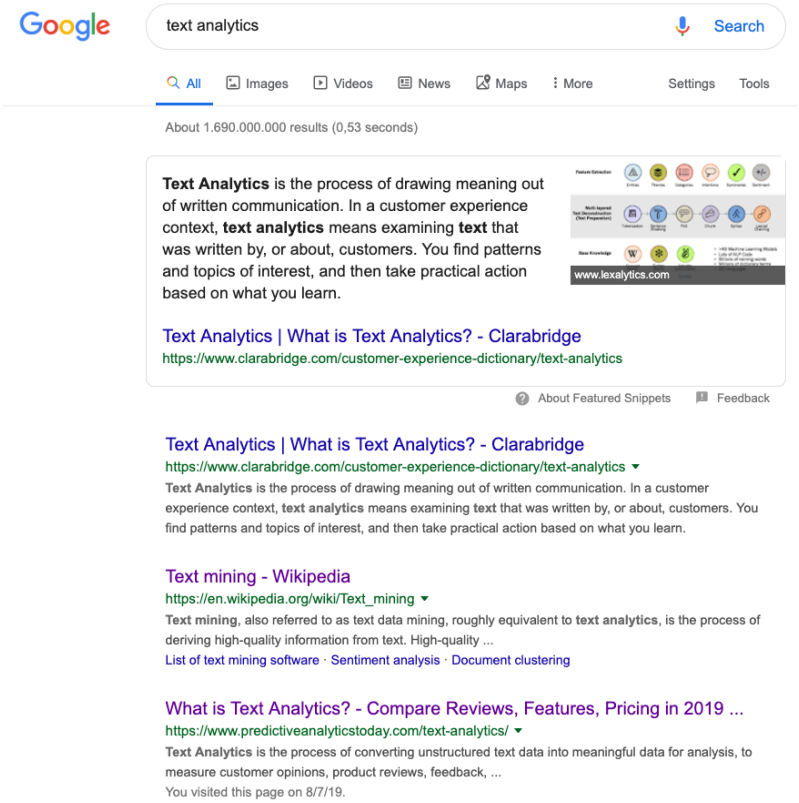


https://imanage.com/blog/named-entity-recognition-ravn-part-1/
https://wordlift.io/blog/en/entity/named-entity-recognition/

Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Some common text mining tasks

- Document retrieval
  - Given a query q rank documents by relevance (similarity)

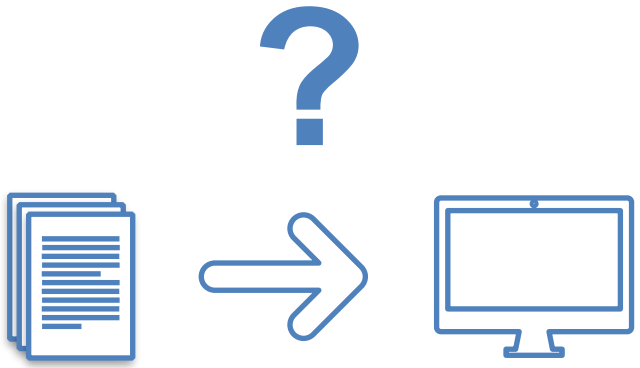


Adapted from “Introduction to Text Mining”, Hongning Wang (CS@UVa)

Challenges in text mining

- **Data is “free text”**
  - Data is not well-organised
    - Semi-structured or unstructured
  - Natural language text contains ambiguities on many levels
    - Lexical, syntactic, semantic, and pragmatic
  - Learning techniques for processing text typically need annotated training examples
    - Expensive to acquire at scale

- **Where to start?**



Example: Stock price prediction

- **model DJI_b(t) = f(news_headlines(t))**

- **Data row for t = 2016-06-24:**
  - DJI_b = 0 // decreased or stayed the same; “1” - increased
  - news_headline_1 = ‘David Cameron to Resign as PM After EU Referendum.’
  - …

Tokenisation

Document → List of tokens

‘David Cameron to Resign as PM After EU Referendum.’

['David', 'Cameron', 'to', 'Resign', 'as', 'PM', 'After', 'EU', 'Referendum', '.']

Tokenization

Document → List of tokens

‘David Cameron to Resign as PM After EU Referendum.’

1-gram ['David', 'Cameron', 'to', 'Resign', 'as', 'PM', 'After', 'EU', 'Referendum', '.']

2-gram [('David', 'Cameron'), ('Cameron', 'to'), ('to', 'Resign'), ('Resign', 'as'), ('as', 'PM'), ('PM', 'After'), ('After', 'EU'), ('EU', 'Referendum'), ('Referendum', '.')]

...

n-gram

Representing words

1-gram

Words → Vectors

Document	**Categorical variables**	**Representation**
['David',	0: ‘David'	
'Cameron',	1: ’Cameron’	
'to',	2: ‘to’	
'Resign',	3: ‘Resign’	
'as',	4: ’as’	
'PM',	5: ’PM’	
'After',	6: ‘After’	
'EU',	7: ’EU’	
'Referendum',	8: ‘Referendum'	
‘.']	9: ’.’	

Representing words - one-hot encoding

1-gram

Words → Vectors

Document

['David',
'Cameron',
'to',
'Resign',
'as',
'PM',
'After',
'EU',
'Referendum',
'.']

Categorical variables

0: 'David'
1: 'Cameron'
2: 'to'
3: 'Resign'
4: 'as'
5: 'PM'
6: 'After'
7: 'EU'
8: 'Referendum'
9: '.'

vocabulary

One-hot encoding

'David' = (**1**, 0, 0, 0, 0, 0, 0, 0, 0, 0)
'Cameron' = (0, **1**, 0, 0, 0, 0, 0, 0, 0, 0)
'to' = (0, 0, **1**, 0, 0, 0, 0, 0, 0, 0)
'Resign' = (0, 0, 0, **1**, 0, 0, 0, 0, 0, 0)
'as' = (0, 0, 0, 0, **1**, 0, 0, 0, 0, 0)
'PM' = (0, 0, 0, 0, 0, **1**, 0, 0, 0, 0)
'After' = (0, 0, 0, 0, 0, 0, **1**, 0, 0, 0)
'EU' = (0, 0, 0, 0, 0, 0, 0, **1**, 0, 0)
'Referendum' = (0, 0, 0, 0, 0, 0, 0, 0, **1**, 0)
'.' = (0, 0, 0, 0, 0, 0, 0, 0, 0, **1**)

Representing words - word embeddings

1-gram

Words → Vectors

Document

['David',
'Cameron',
'to',
'Resign',
'as',
'PM',
'After',
'EU',
'Referendum',
'.']

Categorical variables

0: 'David'
1: 'Cameron'
2: 'to'
3: 'Resign'
4: 'as'
5: 'PM'
6: 'After'
7: 'EU'
8: 'Referendum'
9: '.'

vocabulary

Embedding representation

'David' = (.5, .1, .2, .1)
'Cameron' = (.3, .1, -.7, .2)
'to' = (-.9, .1, .3, .3)
'Resign' = (0, .9, .2, 1)
'as' = (1, .3, -.6, -.8)
'PM' = (.8, .2, .3, -.2)
'After' = (.9, .1, .3, .6)
'EU' = (.9, .1, .3, -.6)
'Referendum' = (.3, -.6, -.8, .1)
'.' = (.1, -.2, -.3, .7)

Word embeddings

2-gram

Words → Vectors

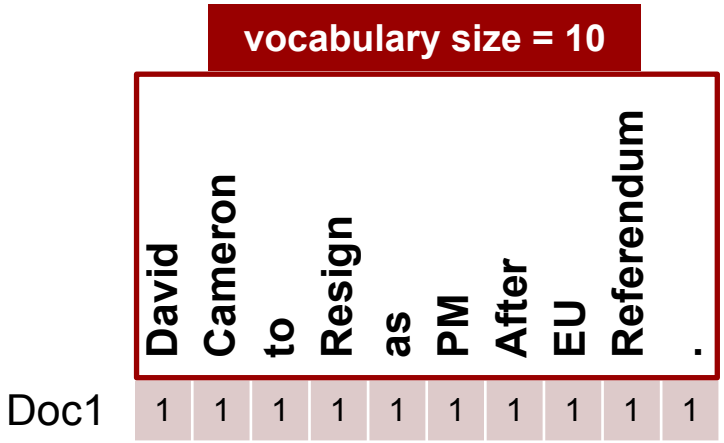
Document	**Categorical variables**	**New representation**
[('David', 'Cameron'),	0: ('David', 'Cameron')	('David', 'Cameron') = (**1**, 0, …)
('Cameron', 'to'),	1: ('Cameron', 'to')	('Cameron', 'to') = (0, **1**, …)
…]	…	…

Document vectorisation: Bag-of-Words (BoW)

- Document = sum of vectorised words
- **Grammar and word ordering is ignored**

Document vectorisation: Bag-of-Words (BoW)

- Document = sum of vectorised words
- Doc1 = 'David Cameron to Resign as PM After EU Referendum.'



Document vectorisation: Bag-of-Words (BoW)

- Document = sum of vectorised words

- Doc1 = 'David Cameron to Resign as PM after EU Referendum.'
- Doc2 = ‘Prime Minister resigns after European Union referendum. Cameron leaves a bitter feeling after himself’
- Doc3 = 'Giant panda twins born at a zoo in European Union.’

vocabulary size = 24

	David	Cameron	to	Resign	as	PM	after	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	…
Doc1	1	1	1	1	1	1	1	1	1	1														
Doc2							2			1	1	1	1	1	1	1						1		
Doc3										1				1	1		1	1	1	1	1	1	1	…

Document vectorisation: Bag-of-Words (BoW)

- Document = sum of vectorised words

- Doc1 = 'David Cameron to Resign as PM after EU Referendum.'
- Doc2 = ‘Prime Minister resigns after European Union referendum. Cameron leaves a bitter feeling after himself’
- Doc3 = 'Giant panda twins born at a zoo in European Union.’

vocabulary size = 24

	David	Cameron	to	Resign	as	PM	after	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	…
Doc1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	…
Doc2	0	0	0	0	0	0	2	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	0	…
Doc3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	…

document-term matrix

Document vectorisation: Bag-of-Words (BoW)

These vectors can now be used for

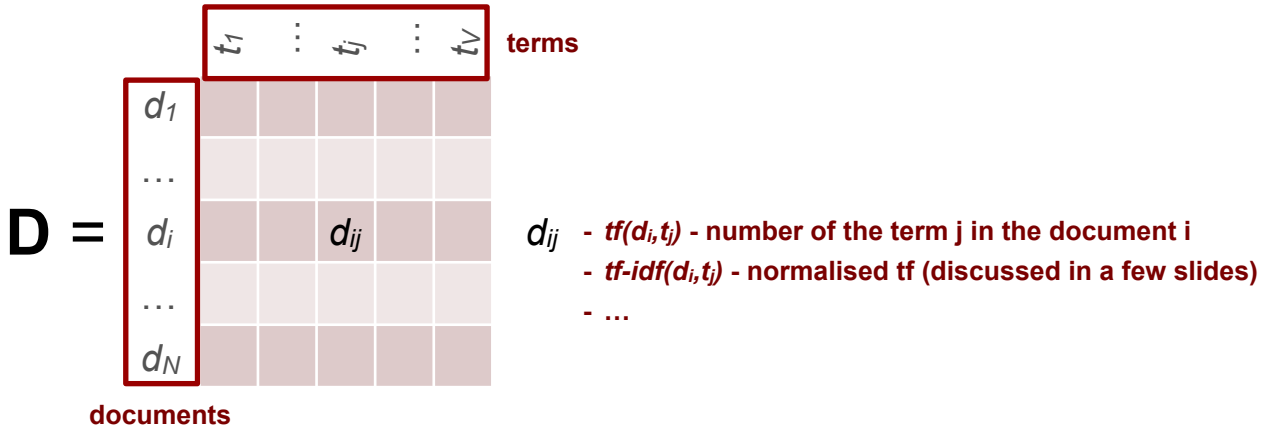
- a ML model for DJI_b(t) = f(news_headlines(t))
- Clustering
- Information retrieval
- …

vocabulary size = 24

	David	Cameron	to	Resign	as	PM	after	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	…
Doc1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	…
Doc2	0	0	0	0	0	0	2	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	0	…
Doc3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	…

document-term matrix

Document-term matrix (D)



- N - number of documents
- V - vocabulary size - quickly gets to thousands! - For English ~ 10^6
- Usually super-sparse

Document retrieval

- Query document $\boldsymbol{q}$
- Rank documents $\boldsymbol{d_i}$ by similarity to $\boldsymbol{q}$

- **How to do it?**

Document retrieval

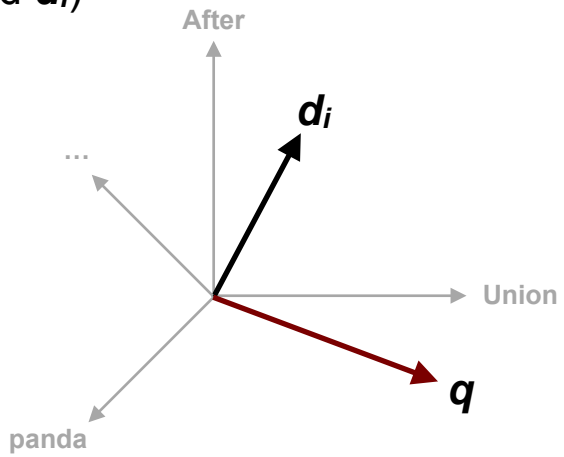
- Query document $\boldsymbol{q} = (q_1, \ldots, q_V)$
- Rank documents $\boldsymbol{d_i} = (d_{i1}, \ldots, d_{iV})$ by similarity to $\boldsymbol{q}$

- **Boolean similarity**
  - bool($\boldsymbol{q}$, $\boldsymbol{d_i}$) = # of terms that both present in $\boldsymbol{q}$ and $\boldsymbol{d_i}$

- **Cosine similarity**
  - CosSim($\boldsymbol{q}$, $\boldsymbol{d_i}$) = 1 - cos(angle between $\boldsymbol{q}$ and $\boldsymbol{d_i}$)

$$= 1 - \frac{\sum_{j=1}^{V} q_j d_{ij}}{\sqrt{\sum_{j=1}^{V} q_j^2}\sqrt{\sum_{j=1}^{V} d_{ij}^2}}$$

- …



Document retrieval

- Query document **Q**
- Rank documents **D** by similarity to **Q**

- Doc1 = 'David Cameron to Resign as PM After EU Referendum.'
- Doc2 = 'Prime Minister Resigns After European Union referendum.'
- Doc3 = 'Giant panda twins born at a zoo in European Union.'

Q = 'Union After panda.'

	David	Cameron	to	Resign	as	PM	After	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	in	Boolean similarity	Cosine similarity
Doc1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.31
Doc2	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	**3**	**0.53**
Doc3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	**3**	0.45

Term frequency - inverse document frequency (tf-idf)

- Common words which are present in almost all documents are not helpful for differentiating
- tf-idf penalises frequent words and increase importance of rare words

- **d_{ij} are weighted: *tf-idf = tf * idf***
  - *tf(t, d)* = number of term *t* in a document *d* (what we had before)
  - *idf(t)* = log[*N* / *df(t)*]
    - *N* is the total number of documents
    - *df(t)* is a number of documents containing term *t*

Term frequency - inverse document frequency (tf-idf)

- Common words which are present in almost all documents are not helpful for differentiating
- tf-idf penalises frequent words and increase importance of rare words

- **d_{ij} are weighted: *tf-idf = tf * idf***
  - *tf(t, d)* = number of term *t* in a document *d* (what we had before)
  - *idf(t)* = log[*N* / *df(t)*]
    - *N* is the total number of documents
    - *df(t)* is a number of documents containing term *t*

- Normalisation of documents (rows of **D**)
  - Document length (sum of d_{ij} over *j*)
  - Cosine (square root of sum of d^2_{ij} over *j*)
  - Max number of term appears in all documents
  - …

Document retrieval: tf-idf

- Query document **Q**
- Rank documents **D** by similarity to **Q**

- Doc1 = 'David Cameron to Resign as PM After EU Referendum.'
- Doc2 = 'Prime Minister Resigns After European Union referendum.'
- Doc3 = 'Giant panda twins born at a zoo in European Union.'

Q = 'Union After panda.'

	David	Cameron	to	Resign	as	PM	After	EU	Referendum	.	Prime	Minister	Resigns	European	Union	referendum	Giant	panda	twins	born	at	a	zoo	in	Boolean similarity	Cosine similarity	Cosine similarity (tf-idf matrix)
Doc1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.31	0.19
Doc2	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	**3**	**0.53**	0.38
Doc3	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	**3**	0.45	**0.39**

Term frequency - inverse document frequency (tf-idf)

- There are many modifications of the tf-idf

Term frequency (if $\mathrm{tf}_{t,d} > 0$)		Document frequency		Document Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log \mathrm{tf}_{t,d}$	t (idf)	$\log N/\mathrm{df}_t$	c (cosine)	$1/\sqrt{\sum_i w_i^2}$
a (augmented)	$0.5 + \frac{0.5 \cdot \mathrm{tf}_{t,d}}{\max_t \mathrm{tf}_{t,d}}$	p (prob idf)	$\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$ (see [MRS08])
b (boolean)	1	(idf smooth)	$\log N/(\mathrm{df}_t + 1)$	b (byte size)	$1/\mathrm{CharLength}^\alpha, \alpha < 1$
L (log mean)	$\frac{1+\log \mathrm{tf}_{t,d}}{1+\mathrm{mean}_{t \in d} \log \mathrm{tf}_{t,d}}$				
d (double log)	$1 + \log(1 + \log \mathrm{tf}_{t,d})$				
(BM25)	$\frac{k \cdot \mathrm{tf}_{t,d}}{k + b \cdot (L-1) + \mathrm{tf}_{t,d}}$	(BM25)	$\log \frac{N-\mathrm{df}_t+.5}{\mathrm{df}_t+.5}$		
(three val.)	$\min\{\mathrm{tf}_{t,d}, 2\}$				
(log1p)	$\log(1 + \mathrm{tf}_{t,d})$				
(sqrt)	$\sqrt{\mathrm{tf}_{t,d}}$			(manhattan)	$1/\sum_i w_i$

Erich Schubert, "Advanced Topics in Text Mining", Heidelberg U (2017)

Text Preprocessing (“Normalisation”)

David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

Text Preprocessing (“Normalisation”)

David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

stopwords

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

Text Preprocessing (“Normalisation”)

David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

stopwords

punctuation

Text Preprocessing ("Normalisation")

David Cameron to Resign as PM After EU Referendum . Prime Minister Resigns European Union referendum Giant panda twins born at a zoo in

stopwords

punctuation

abbreviations

60 Most Important Abbreviations

1. **LOL**: laugh out loud
2. **OMG**: Oh my God
3. **ILY**: I love you
4. **LMAO**: laughing my a** off
5. **TTYN**: Talk to you never
6. **FBO**: Facebook official
7. **TTYS**: Talk to you soon
8. **HMB**: Hit me back
9. **SFW**: Safe work work
10. **PTFO**: Passed the f** out
11. **ASL**: Age/Sex/Location
12. **AFAIK**: As far as I know
13. **IMHO:** In my humble opinion
14. **IRL**: In rela life
15. **ISO:** In search of
16. **J/K:** Just Kidding
17. **L8R:** Later
18. **POV:** Point of view
19. **RBTL:** Read between the lines
20. **RT**: Real time
21. **BTW**: By the way
22. **CTN**: Can't talk now
23. **CYE**: Check your email
24. **dl**: Download
25. **ETA**: Estimated time of arrival
26. **FWIW**: For what it's worth
27. **FYI**: For your information
28. **GG**: Good game
29. **GJ**: Good job
30. **GL**: Good luck
31. **gr8**: Great
32. **GTG**: Got to go
33. **GMV**: Got my vote
34. **HTH**: Hope this helps
35. **OT**: Off topic
36. **PC**: Personel computer
37. **pls**: Please
38. **POS**: Parent over shoulder
39. **ppl**: People
40. **Txt**: Text
41. **BRB:** Be Right Back
42. **B4N:** Bye for Now
43. **BCNU:** Be seeing you
44. **BFF:** Best Friends Forever
45. **CYA:** Cover You're A**
46. **TY**: Thank you
47. **w/e**: Whatever
48. **W8**: Wait
49. **XOXO**: Hugs and kisses
50. **Y**: Why
51. **YOLO**: You only live once
52. **YNt**: Why not
53. **YW**: You're welcome
54. **ZZZ**: Sleeping
55. **MMB**: Message me back
56. **msg**: Message
57. **NC**: No comment
58. **noob**: Newbie
59. **GMV**: Got my vote
60. **HTH**: Hope this helps

English Study Here

www.englishstudyhere.com

Text Preprocessing (“Normalisation”)

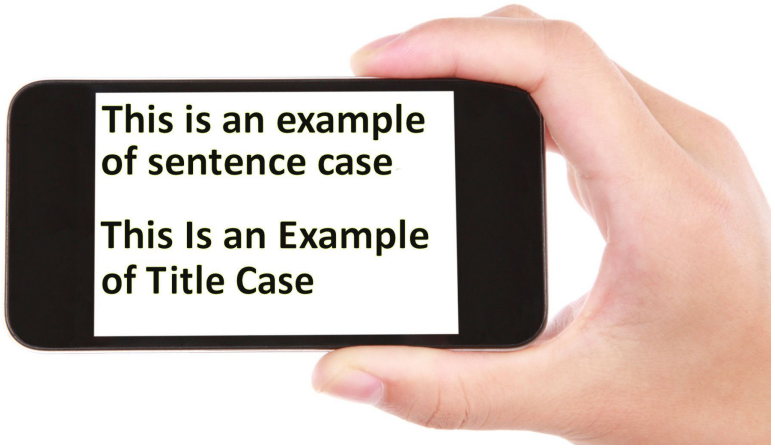
David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

stopwords

punctuation

abbreviations

capitalisation



Text Preprocessing (“Normalisation”)

David Cameron to Resign as PM After EU Referendum . Prime Minister Resigns European Union referendum Giant panda twins born at a zoo in

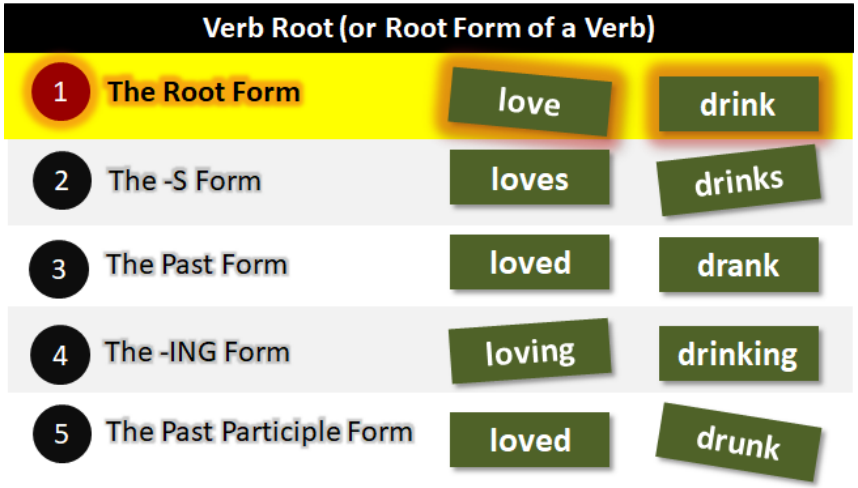
stopwords

punctuation

abbreviations

capitalisation

verb tenses



Text Preprocessing ("Normalisation")

David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

stopwords

punctuation

abbreviations

capitalisation

verb tenses

proper nouns

Text Preprocessing (“Normalisation”)

David Cameron to Resign as PM After EU Referendum . Prime Minister Resigns European Union referendum Giant panda twins born at a zoo in

stopwords

punctuation

abbreviations

capitalisation

verb tenses

proper nouns

Lemmatization

Stemming

adjustable → adjust
formality → formaliti
formaliti → formal
airliner → airlin ⚠

Lemmatization

was → (to) be
better → good
meeting → meeting

Text Preprocessing (“Normalisation”)

David
Cameron
to
Resign
as
PM
After
EU
Referendum
.
Prime
Minister
Resigns
European
Union
referendum
Giant
panda
twins
born
at
a
zoo
in

stopwords

punctuation

abbreviations

capitalisation

verb tenses

proper nouns

Lemmatization

Numbers, synonyms, anaphora, …

Text Preprocessing (“Normalisation”)

- Most common text preprocessing steps (for the BoW representation)
  - remove punctuation
  - remove / replace numbers
  - lowercase
  - remove stopwords
  - stemming / lemmatising

- **Ad hoc!**

Playtime!

- **Text Analysis - Part 1 - Spam Classification.ipynb**
  Do Sections 1-3

Recap

- **Text preprocessing steps:**

1. Tokenisation
2. Normalisation (remove punctuation, stopwords, stemming, lowercase, etc)
3. Vectorisation (tokenisation, converting documents to vectors)

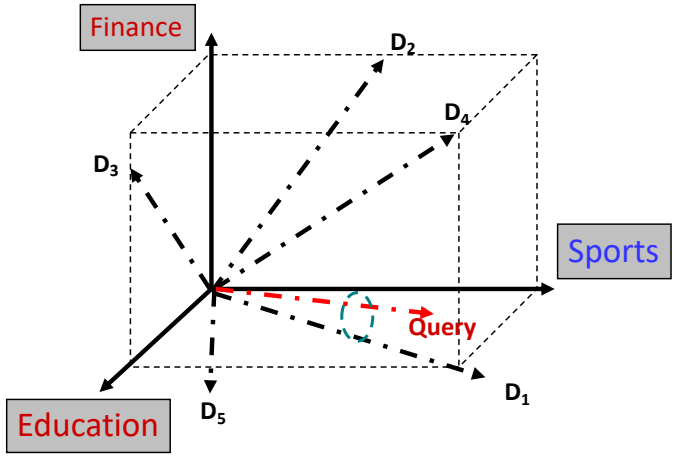
Topic Modeling

- D is high dimensional (News example: 73608 documents x 43509 terms)
- Computationally expensive
- Difficult to deal with synonyms and related words
  (“car” vs “automobile” vs “vehicle”)

- **Dimensionality reduction can help!**

Topic Modeling: General Idea

- Topic modelling is about decomposing a corpus of documents D into sets of concepts (“topics”)
- Then each document D_i can be (re-)represented by linear combinations of such topics

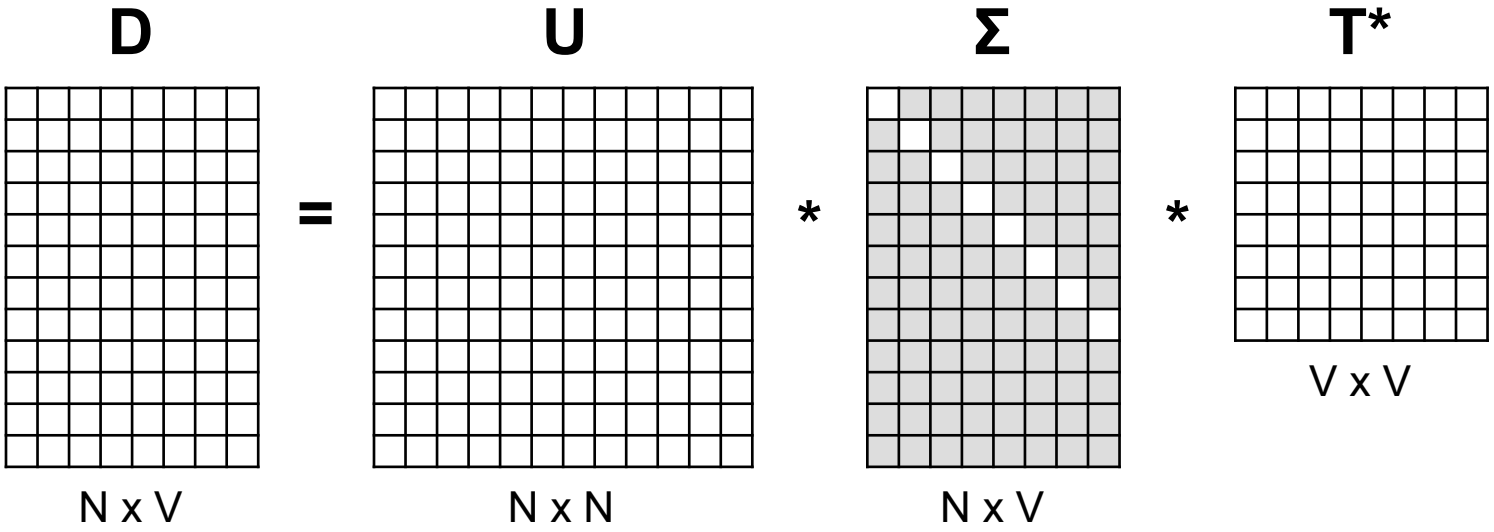


Topic modelling via matrix decomposition: Latent Semantic Indexing (Analysis) - LSI (LSA)

- Is an indexing and retrieval method that uses Singular Vector Decomposition (SVD) to identify patterns in the relationships between the terms and documents.
- It is based on the principle that words that are used in the same contexts tend to **have similar meanings**.
- A key feature of LSI: its ability to extract the conceptual content of a body of text by establishing associations between those **terms that occur in similar contexts**.
- Could trace back its history to factor analysis applications in mid 1960s, but it started gaining the popularity in late 80s to early 90s. Nowadays, LSI is being used in many applications on a daily basis.

Latent Semantic Indexing (Analysis) - LSI (LSA)

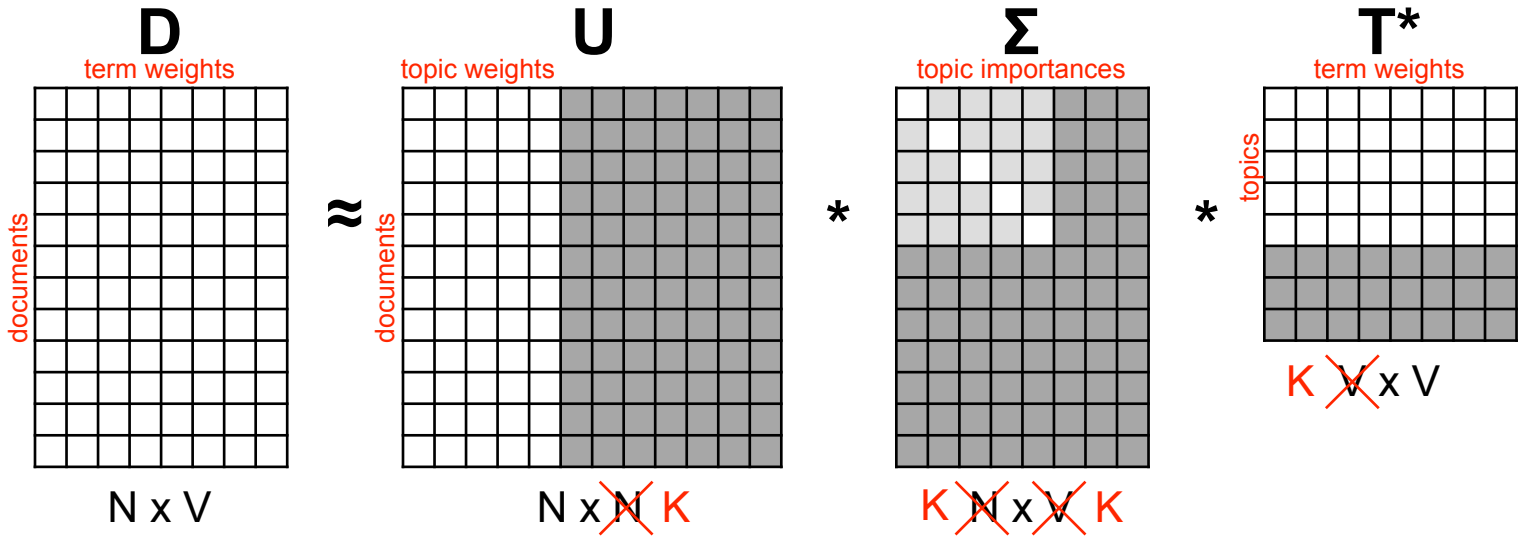
- It is a Singular Value Decomposition (generalisation of eigenvalue decomposition) of the document-term matrix **D=UΣT***



- **D** is a document-term matrix
- **U** is a document-topic map (“topic distribution”)
- **Σ** is an ordered diagonal matrix of singular values (“topic importance”)
- **T*** is a term-topic map (“term distribution”)

Latent Semantic Indexing (Analysis) - LSI (LSA)

- It is a Singular Value Decomposition (generalisation of eigenvalue decomposition) of the document-term matrix **D=UΣT***

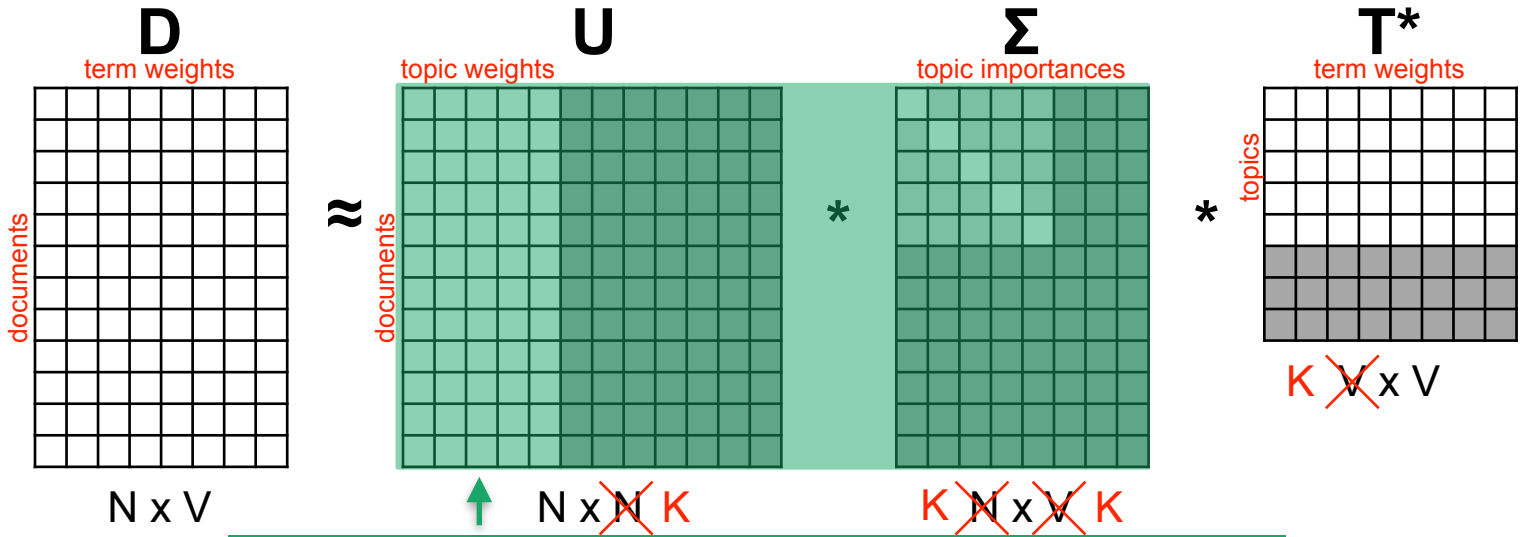


- **D** is a document-term matrix
- **U** is a document-topic map (“topic distribution”)
- **Σ** is an ordered diagonal matrix of singular values (“topic importance”)
- **T*** is a term-topic map (“term distribution”)

Truncate the matrices to K topics (keep enough…)

Latent Semantic Indexing (Analysis) - LSI (LSA)

- It is a Singular Value Decomposition (generalisation of eigenvalue decomposition) of the document-term matrix **D=UΣT***



- **D** is a do
- **U** is a do
- **Σ** is an o
- **T** is a te

…nce")

Instead of D, U*Σ can now be used for

- a ML model for DJI_b(t)
- Clustering
- Information retrieval

…

Truncate the matrices to K topics (keep enough…)

LSI results for the news titles: Topics (k=25)

T* term-topic map:

0: 0.963*”number" + 0.107*"year" + 0.063*"people" + 0.055*"million" + 0.050*"world" + 0.048*"say" + 0.044*"killed" + 0.041*"new" + 0.039*"amp" + 0.037*” china"

1: 0.489*"say" + 0.214*"new" + -0.211*"number" + 0.199*"government" + 0.188*"world" + 0.162*"country" + 0.154*"israel" + 0.146*"china" + 0.145*"state" + 0.141*"police"

2: -0.805*"say" + 0.304*"new" + 0.282*"world" + 0.164*"china" + 0.144*"government" + 0.139*"country" + 0.122*"year" + 0.089*"state" + 0.077*"people" + 0.075*"war"

3: 0.489*"world" + -0.402*"police" + 0.377*"china" + -0.282*"israel" + 0.205*"say" + -0.159*"israeli" + 0.147*"korea" + -0.137*"palestinian" + 0.134*"north" + -0.127*"attack"

4: -0.397*"israel" + -0.371*"korea" + -0.341*"north" + 0.310*"police" + 0.286*"new" + 0.196*"say" + 0.164*"world" + -0.157*"state" + -0.154*"south" + 0.153*"year"

5: -0.465*"world" + 0.399*"china" + -0.342*"israel" + 0.338*"korea" + 0.312*"north" + 0.299*"new" + 0.161*"south" + 0.157*"year" + -0.134*"israeli" + -0.130*"palestinian"

6: 0.540*"new" + -0.397*"police" + 0.332*"year" + -0.315*"world" + 0.274*"israel" + -0.229*"china" + -0.213*"people" + -0.168*"korea" + -0.158*"north" + 0.098*"palestinian"

7: 0.851*"year" + -0.409*"new" + 0.108*"police" + -0.107*"government" + 0.088*"old" + -0.080*"number" + -0.073*"people" + 0.069*"prison" + -0.067*"amp" + 0.058*"woman"

8: -0.682*"government" + 0.391*"police" + 0.364*"new" + -0.192*"country" + -0.166*"people" + 0.153*"world" + 0.150*"korea" + 0.135*"north" + 0.125*"woman" + -0.113*"china"

9: -0.763*"china" + 0.331*"korea" + 0.308*"north" + 0.229*"government" + -0.199*"israel" + 0.129*"world" + 0.107*"south" + -0.098*"chinese" + 0.097*"people" + 0.082*”korean"

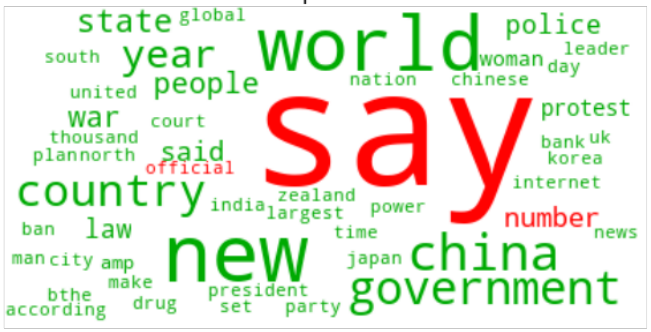
…

24: 0.794*"people" + -0.332*"government" + -0.289*"police" + 0.230*"killed" + 0.100*"right" + -0.099*"uk" + 0.087*"year" + -0.075*"world" + 0.073*"human" + -0.070*"nuclear"

LSI results for the news titles: Topics (k=25)

Topic #1

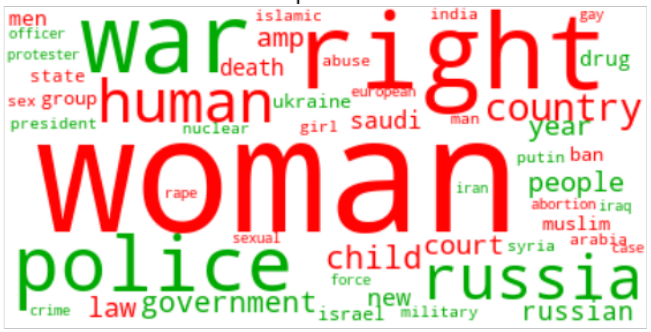
Topic #2



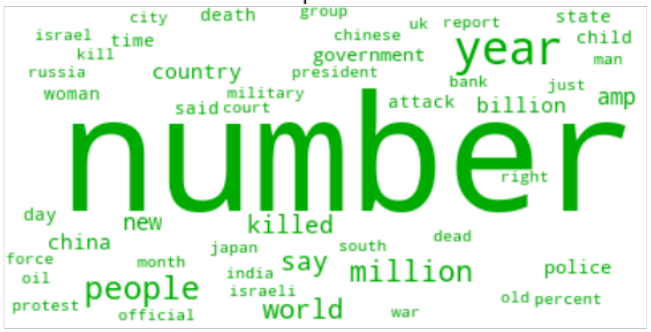
Topic #5

Topic #4

Topic #12



Topic #0



LSI results for the news titles: Documents by topics

- “North Korea seen to fire submarine-launched ballistic missile: South Korea”
- ['north', 'korea', 'seen', 'submarinelaunched', 'ballistic', 'missile', 'south', ‘korea']
- Document-topic assignment U*Σ is: **5:** -1.71, **8:** -1.24, **3:** 0.51, **10:** 0.44, …

Topic #5

Topic #8

Topic #3

Topic #10

LSI results for the news titles: Documents by topics

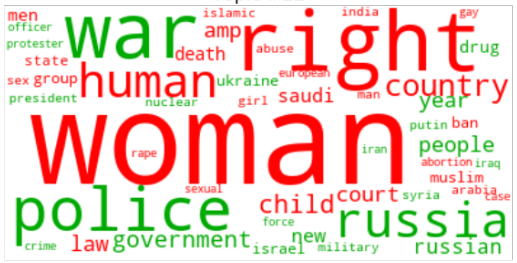
- “North Korea has threatened to conduct a nuclear test in response to a United Nations move towards a probe into the country's human rights violations”
- ['north', 'korea', 'threatened', 'conduct', 'nuclear', 'test', 'response', 'united', 'nation', 'probe', 'country', 'human', 'right', 'violation']
- Document-topic assignment U*Σ is: **5:** -1.11, **15:** 0.79, **8:** -0.74, **12:** -0.74, **1:** 0.74, **18:** -0.69, …

Topic #5

Topic #15

Topic #8

Topic #12



Topic #1

Topic #18

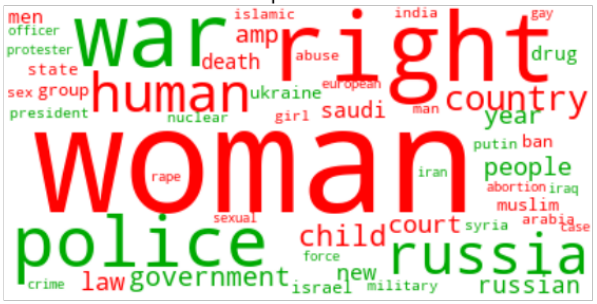
LSI results for the news titles: Documents by topics

- “Argentine Court rules: Orang Utans are "non-human-persons" with human rights and therefore need to be released from zoo”
- ['argentine', 'court', 'rule', 'orang', 'utans', 'nonhumanpersons', 'human', 'right', 'need', 'released', ‘zoo']
- Document-topic assignment U*Σ is: **16:** 0.91, **18:** -0.67, **12:** -0.66, **15:** 0.33, …

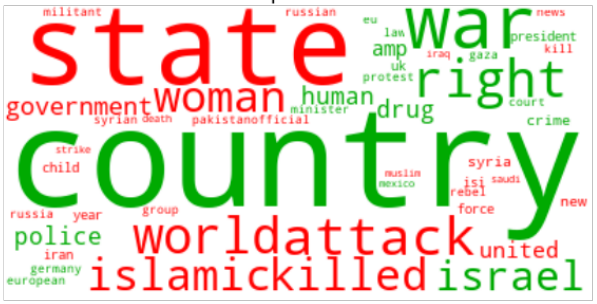
Topic #16

Topic #18

Topic #12



Topic #15



LSI beyond Text

- Recommender systems (next week)
- Image processing (“Eigenfaces”)
- ...

LSI problems

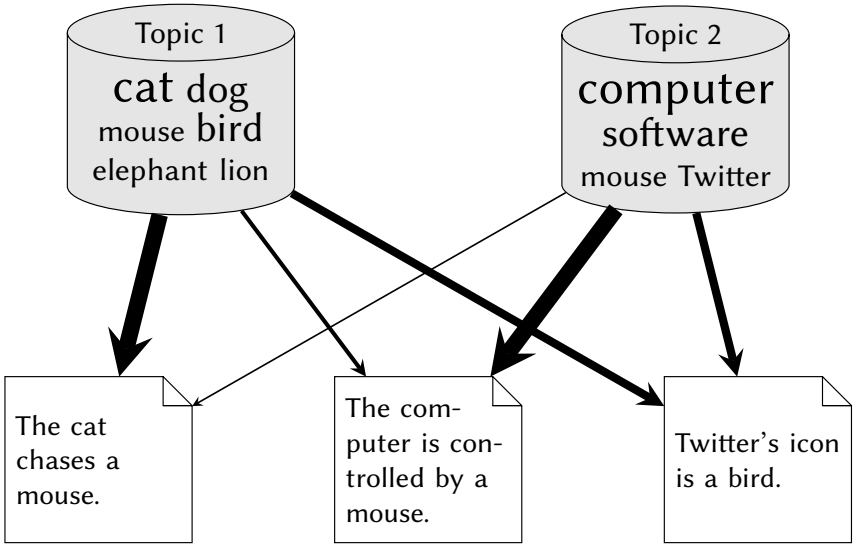
- Computationally expensive: Complexity $O(\min\{NV^2, N^2V\})$
  - Monte-Carlo sampling, approximations…
- Difficult to interpret the decomposition results (e.g., negative values)

- **Another approach: Probabilistic modelling**

Probabilistic topic modeling

- **U** is a document-topic map (“topic distribution”)
- **T** is a term-topic map (“term distribution”)

- **Come up with a probabilistic model that generates documents**
  U ~ P(topics | documents)
  T ~ P(terms | topics)



Erich Schubert, “Advanced Topics in Text Mining”, Heidelberg U (2017)

Latent Dirichlet Allocation (LDA)

- This is better formulated as a “generative story”:

Let’s consider each observed document i $(i = 1, 2, ..., M)$ as a vector, $\mathbf{w}_i$, with its sequence of words[1]:

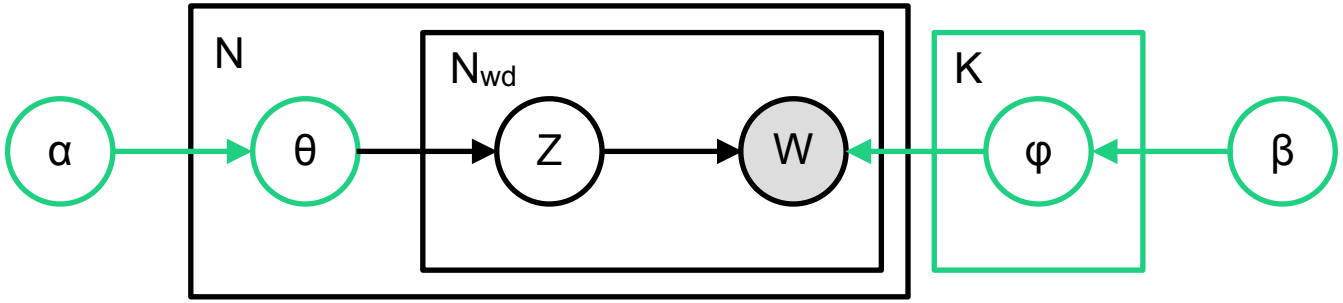
1. For each document i, there is a vector of K topics
2. For each topic k, there is a vector of C words (dictionary)
3. For each of the words, j, in each document i $(j = 1, 2, ..., ||\mathbf{w}_i||)$, we have
    1. A topic assignment, $z_{i,j}$ $(z_{i,j} \in \{1, 2, ..., K\})$
    2. A word, $w_{i,j}$

[1] Only for simplicity. BoW formulation would have more complicated steps 3.1 and 3.2.

Latent Dirichlet Allocation (LDA)

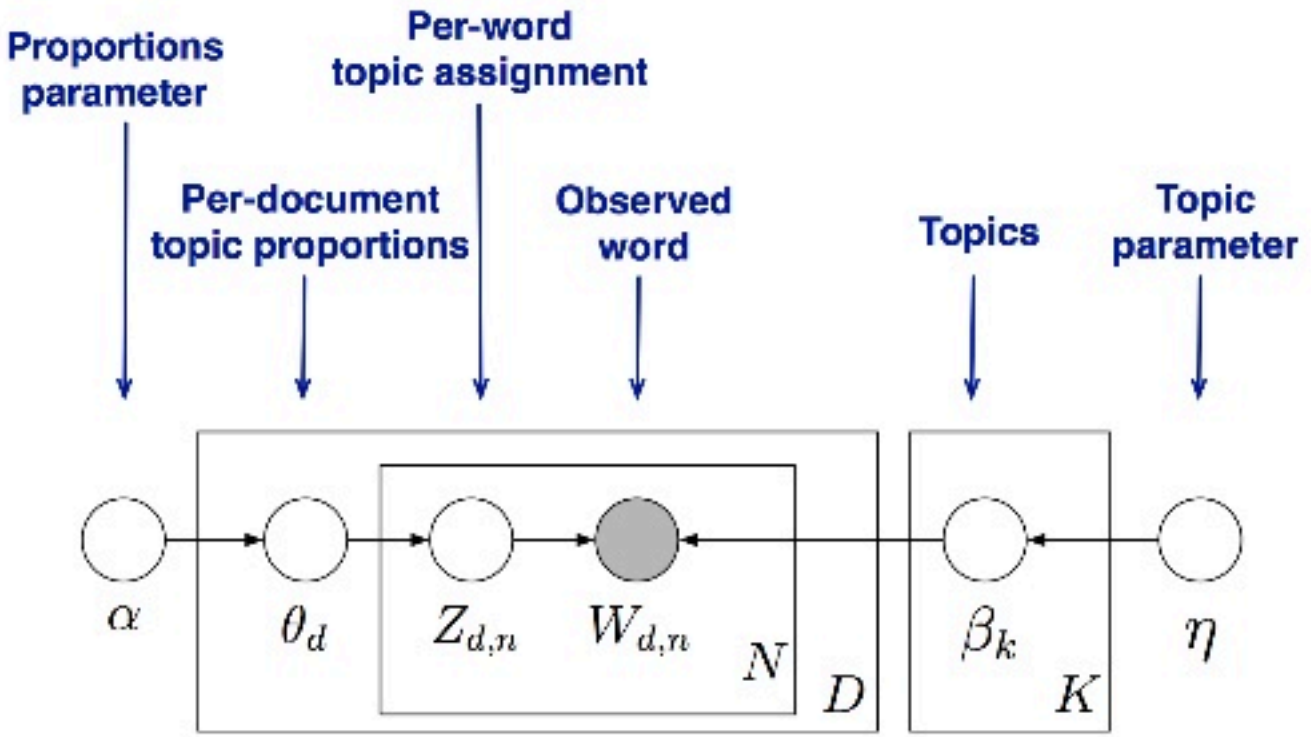
- LDA is a Bayesian version of LSA. In particular, it uses Dirichlet priors for the document-topic and word-topic distributions, lending itself to better generalization.

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) P(W_{j,t} \mid \varphi_{Z_{j,t}}),$$



1. Choose $\theta_i \sim \mathrm{Dir}(\alpha)$, where $i \in \{1, \ldots, M\}$ and $\mathrm{Dir}(\alpha)$ is a Dirichlet distribution with a symmetric parameter α which typically is sparse ($\alpha < 1$)
2. Choose $\varphi_k \sim \mathrm{Dir}(\beta)$, where $k \in \{1, \ldots, K\}$ and β typically is sparse
3. For each of the word positions i, j, where $i \in \{1, \ldots, M\}$, and $j \in \{1, \ldots, N_i\}$
   (a) Choose a topic $z_{i,j} \sim \mathrm{Multinomial}(\theta_i)$.
   (b) Choose a word $w_{i,j} \sim \mathrm{Multinomial}(\varphi_{z_{i,j}})$.

LDA: Graphical Model



$$p(\beta, \theta, z, w|\alpha, \eta) = \prod_{i=1}^{K} p(\beta_i|\eta) \prod_{d=1}^{D} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right)$$

Source: Blei, ICML 2012 tutorial

5/15

Latent Dirichlet Allocation (LDA)

```
//Latent Dirichlet Allocation
//Model Based Machine Learning, DTU
//2017-2018


data {
  int<lower=2> K;               // num topics
  int<lower=2> V;               // num words
  int<lower=1> M;               // num docs
  int<lower=1> N;               // total word instances
  int<lower=1,upper=V> w[N];    // word n
  int<lower=1,upper=M> doc[N];  // doc ID for word n
  vector<lower=0>[K] alpha;     // topic prior
  vector<lower=0>[V] beta;      // word prior
}
parameters {
  simplex[K] theta[M];   // topic dist for doc m
  simplex[V] phi[K];     // word dist for topic k
}
model {
  for (m in 1:M)
    theta[m] ~ dirichlet(alpha);  // prior
  for (k in 1:K)
    phi[k] ~ dirichlet(beta);     // prior
  for (n in 1:N) {
    real gamma[K];
    for (k in 1:K)
      gamma[k] <- log(theta[doc[n],k]) + log(phi[k,w[n]]);
    increment_log_prob(log_sum_exp(gamma));  // likelihood
  }
}
```

LDA results for the news titles: Topics (k=25)

0: 0.058*"number" + 0.048*"bus" + 0.036*"city" + 0.030*"hit" + 0.027*"water" + 0.022*"japan" + 0.021*"time" + 0.020*"talk" + 0.017*"world" + 0.016*”record"

1: 0.133*"war" + 0.051*"russia" + 0.039*"military" + 0.037*"russian" + 0.034*"troop" + 0.032*"force" + 0.025*"crime" + 0.025*"east" + 0.023*"saudi" + 0.018*"arm"

2: 0.065*"minister" + 0.047*"british" + 0.035*"britain" + 0.032*"prime" + 0.030*"wikileaks" + 0.027*"election" + 0.026*"party" + 0.021*"freedom" + 0.018*"warning" + 0.016*"take"

3: 0.047*"want" + 0.034*"dont" + 0.031*"school" + 0.023*"turn" + 0.021*"control" + 0.017*"die" + 0.017*"son" + 0.017*"release" + 0.015*"block" + 0.013*"europe"

4: 0.078*"state" + 0.035*"president" + 0.033*"nation" + 0.031*"united" + 0.030*"bomb" + 0.028*"security" + 0.026*"africa" + 0.025*"threat" + 0.018*"open" + 0.018*"national"

5: 0.095*"police" + 0.055*"protest" + 0.045*"dead" + 0.034*"man" + 0.031*"arrested" + 0.025*"shot" + 0.025*"mexico" + 0.023*"street" + 0.021*"officer" + 0.018*"violence"

6: 0.080*"woman" + 0.070*"child" + 0.032*"girl" + 0.029*"men" + 0.027*"family" + 0.024*"sex" + 0.023*"charge" + 0.020*"abuse" + 0.020*"accused" + 0.019*"trial"

7: 0.094*"gaza" + 0.055*"news" + 0.047*"video" + 0.045*"drug" + 0.027*"medium" + 0.023*"bbc" + 0.022*"tv" + 0.020*"site" + 0.016*"banned" + 0.016*"online"

8: 0.066*"right" + 0.051*"law" + 0.041*"human" + 0.026*"european" + 0.023*"torture" + 0.021*"eu" + 0.019*"vote" + 0.018*"gay" + 0.018*"demand" + 0.018*"parliament"

9: 0.065*"china" + 0.061*"nuclear" + 0.045*"world" + 0.034*"power" + 0.032*"india" + 0.016*"economy" + 0.016*"deal" + 0.015*"look" + 0.014*"zimbabwe" + 0.014*"building"

10: 0.034*"aid" + 0.030*"gaza" + 0.028*"food" + 0.028*"hamas" + 0.027*"head" + 0.024*"journalist" + 0.018*"away" + 0.018*"big" + 0.017*"doe" + 0.016*"legal"

…

LDA results for the news titles: Topics (k=25)

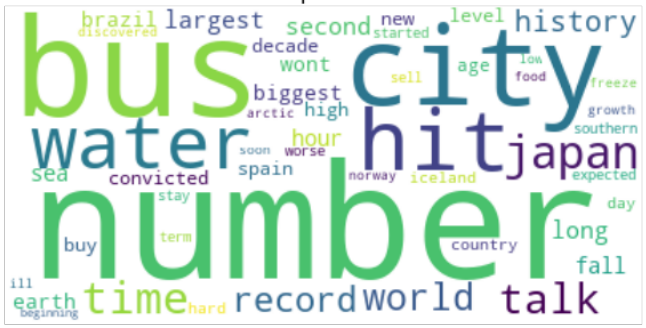
Topic #13

Topic #1

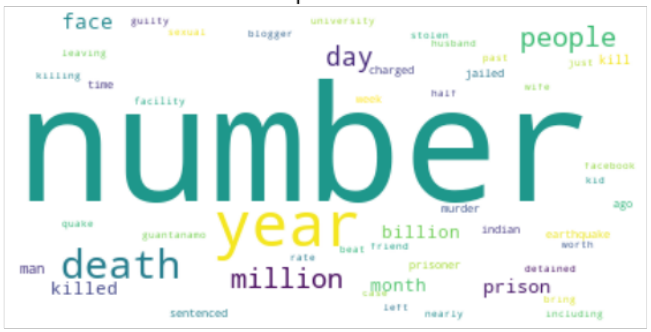
Topic #21

Topic #15

Topic #0



Topic #24



LDA results for the news titles: Documents by topics

- “North Korea has threatened to conduct a nuclear test in response to a United Nations move towards a probe into the country's human rights violations”
- ['north', 'korea', 'threatened', 'conduct', 'nuclear', 'test', 'response', 'united', 'nation', 'probe', 'country', 'human', 'right', 'violation']
- **15:** 0.75, **21:** 0.12, …

Topic #15

Topic #21

LDA results for the news titles: Documents by topics

- “North Korea has threatened to conduct a nuclear test in response to a United Nations move towards a probe into the country's human rights violations”
- ['north', 'korea', 'threatened', 'conduct', 'nuclear', 'test', 'response', 'united', 'nation', 'probe', 'country', 'human', 'right', 'violation']
- **8:** 0.23, **15:** 0.20, **21:** 0.15, **4:** 0.15, **9:** 0.14, …

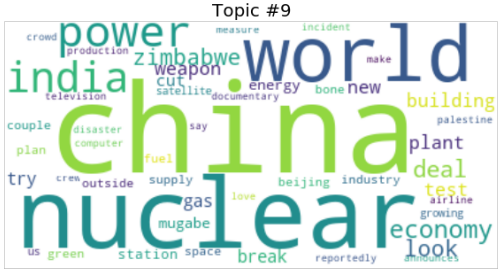
Topic #8

Topic #15

Topic #21

Topic #4

Topic #9



LDA results for the news titles: Documents by topics

- “Argentine Court rules: Orang Utans are "non-human-persons" with human rights and therefore need to be released from zoo”
- ['argentine', 'court', 'rule', 'orang', 'utans', 'nonhumanpersons', 'human', 'right', 'need', 'released', ‘zoo']
- **8:** 0.50, **22:** 0.13, **23:** 0.12, **3:** 0.12, …

Topic #8

Topic #22

Topic #23

Topic #3

Playtime!

- Text Analysis - Part 1 - Spam Classification.ipynb
  - Do section “4. Topic Modeling”

- Text Analysis - Part 2 - Stock prediction.ipynb (optional)
  - Extra. LDA.ipynb (optional)

One more thing…

- State of the art for text data: Deep learning

https://openai.com/blog/better-language-models/

Deep learning for text data

Human-generated input

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Machine completion

https://openai.com/blog/better-language-models/

Deep learning for text data

Human-generated input

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Machine completion

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd…

https://openai.com/blog/better-language-models/

Recommended reading

- Text pre-processing, TF-IDF, Lemmatization, etc. [optional]

https://towardsdatascience.com/preprocessing-text-in-python-923828c4114f

- Latent Semantic Analysis tutorial [optional]
  https://www.engr.uvic.ca/~seng474/svd.pdf

- "*Intuitive Guide to Latent Dirichlet Allocation" [optional]*
  https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158

Who said that text mining is stressful?

I'm 27 and I feel great!