University of Amsterdam
Economics & Business

Data Science Minor - 2020/2021 - Sem. 1, Period 3

Text Retrieval - Text Mining
Week 2 – Semantic Representations

Julien ROSSI

Week 2

After this lecture, you will:

- Understand the challenges of **semantic representation**
- Know about 2 techniques of **topic modeling**:
  - SVD
  - LDA

Previously in Text Representations

- **Lexical Representations**
- Vocabulary = list of unique token / word / term / lemma / n-grams appearing in the corpus
- 1 dimension = 1 item of the vocabulary

- Raw Counts or TF-IDF
- Preprocessing (lemmatizing, stemming, stopping, …)

- In a lexical representation, a text is seen as an assembly of words

Semantic Representations

- **Semantic = deals with meaning**

- **Text** as a gathering of **topics** (“science”, “business”, “sport”, …)
- These **topics** are responsible for the **terms** that appear
  - Each topic has a list of **terms**
  - Topic “business” → “revenue”, “operations”, etc…
  - Topic “sport” → “goal”, “league”, “ranking”, etc…
  - Think about “terms more likely to appear when talking about the topic”

Semantic Representations

TOPIC

- Weighted list of terms (word / n-gram / stem …)
- Each term has a weight
  - Important terms with high weight
  - Less important terms with low weight

Semantic Representations

DOCUMENT

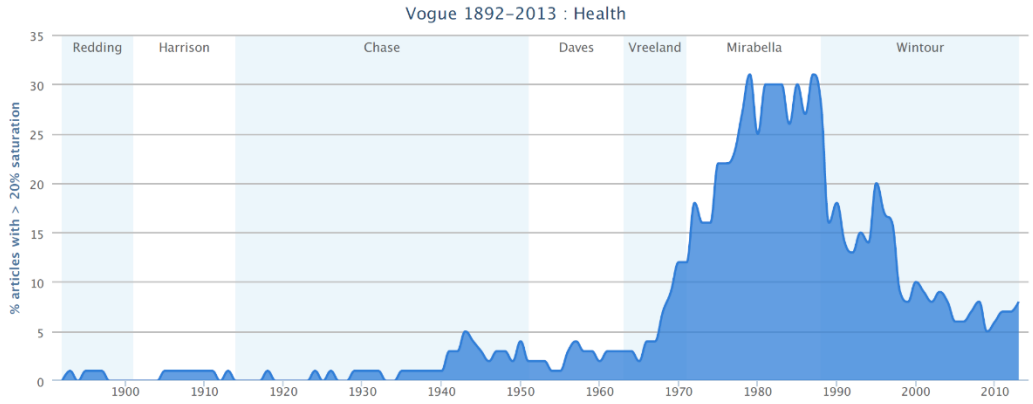
- Weighted list of topics

- Vector representation:
  - 1 dimension = 1 topic
  - Coefficient = weight of topic in document

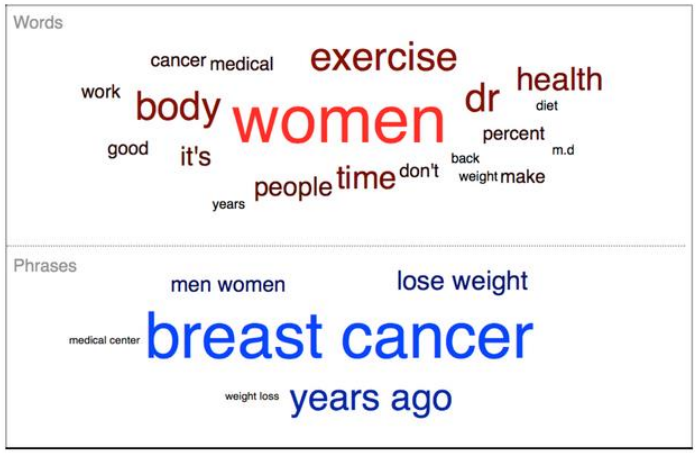
In Practice

"How did **Vogue Magazine** talk about **Health** ?"

- How many articles?



- Using which words?



http://library.ifla.org/930/1/119-leonard-en.pdf

Semantic Representations

Challenge

- Discover the topics from text

- We will see 2 techniques:
  - Singular Value Decomposition (SVD)
  - Latent Dirichlet Allocation (LDA)

SVD

- Discover the topics from text

- Factorization of the Term-Document Matrix
- Based on Linear Algebra

- 3 matrices
  - U → terms / topics
  - ∑ → topics
  - V → documents / topics

- Term-Document Matrix = U * ∑ * V

Factorization

Term-Document Matrix

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

U

ship	–0.44	–0.30	0.57	0.58	0.25
boat	–0.13	–0.33	–0.59	0.00	0.73
ocean	–0.48	–0.51	–0.37	0.00	–0.61
wood	–0.70	0.35	0.15	–0.58	0.16
tree	–0.26	0.65	–0.41	0.58	–0.09

∑

2.16	0.00	0.00	0.00	0.00
0.00	**1.59**	0.00	0.00	0.00
0.00	0.00	**1.28**	0.00	0.00
0.00	0.00	0.00	**1.00**	0.00
0.00	0.00	0.00	0.00	**0.39**

V

d_1	d_2	d_3	d_4	d_5	d_6
–0.75	–0.28	–0.20	–0.45	–0.33	–0.12
–0.29	–0.53	–0.19	0.63	0.22	0.41
0.28	–0.75	0.45	–0.20	0.12	–0.33
0.00	0.00	0.58	0.00	–0.58	0.58
–0.53	0.29	0.63	0.19	0.41	–0.22

University of Amsterdam
Economics & Business

Dimension Reduction 2 dims

U’

ship	–0.44	–0.30
boat	–0.13	–0.33
ocean	–0.48	–0.51
wood	–0.70	0.35
tree	–0.26	0.65

Approximate Term-Document Matrix

	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	–0.08
boat	0.36	0.36	0.16	–0.20	–0.02	–0.18
ocean	1.01	0.72	0.36	–0.04	0.16	–0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	–0.39	–0.08	0.90	0.41	0.49

Σ’

2.16	0.00
0.00	**1.59**

V’

d_1	d_2	d_3	d_4	d_5	d_6
–0.75	–0.28	–0.20	–0.45	–0.33	–0.12
–0.29	–0.53	–0.19	0.63	0.22	0.41

Dimension Reduction

The term-document matrix we have could correspond to the texts:

D2: I cruised the ocean on a boat.
D3: I sail on a ship

(hypothesis: cruise / sail are stopwords)

$\cos(d_2, d_3) = 0.95$

Topic-Document Matrix

V

d_1	d_2	d_3	d_4	d_5	d_6
–0.75	–0.28	–0.20	–0.45	–0.33	–0.12
–0.29	–0.53	–0.19	0.63	0.22	0.41

$\cos(d_2, d_3) = 0.0$

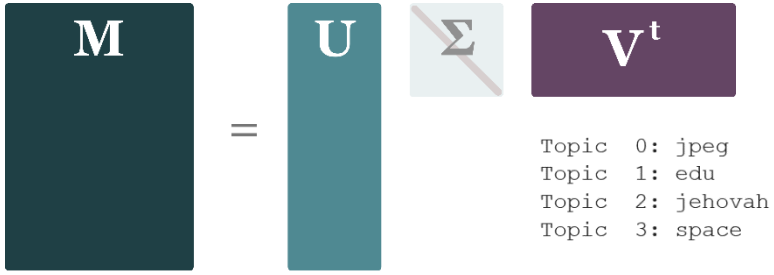
Term-Document Matrix

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

Semantic Representations

SVD

- See the notebook “SVD”




```
Topic  0: jpeg          image        file         gif          edu
Topic  1: edu           graphics     pub          data         ftp
Topic  2: jehovah       elohim       god          lord         jesus
Topic  3: space         earth        planet       venus        spacecraft
```

```
Document at index: 1540
Document Vector: [0.0653272  0.0855462  0.04933061 0.32438576]
Document Topics:
    Computer_01     :  0.07
    Computer_02     :  0.09
    Religion        :  0.05
    Space           :  0.32

Joe,

        your description sounds like one of the  gravity probe  spacecraft
ideas.
```

LDA

- Discover the topics from text

- Generative Model
- The Bag of Word is generated by random process

- Process:
  - 1 document = sum of weighted topics
  - 1 topic = probability distribution over dictionary

LDA

- 2 Topics:
  - “Sports” with words:
    - “football” P=0.4
    - “ronaldo” P=0.2
    - “goal” P=0.2
    - “world cup” P=0.2
  - “Business”
    - “revenue” P=0.6
    - “tax” P=0.3
    - “benefit” P=0.1

LDA

- 1 Document
  - 0.8 “Sport” + 0.2 “Business”

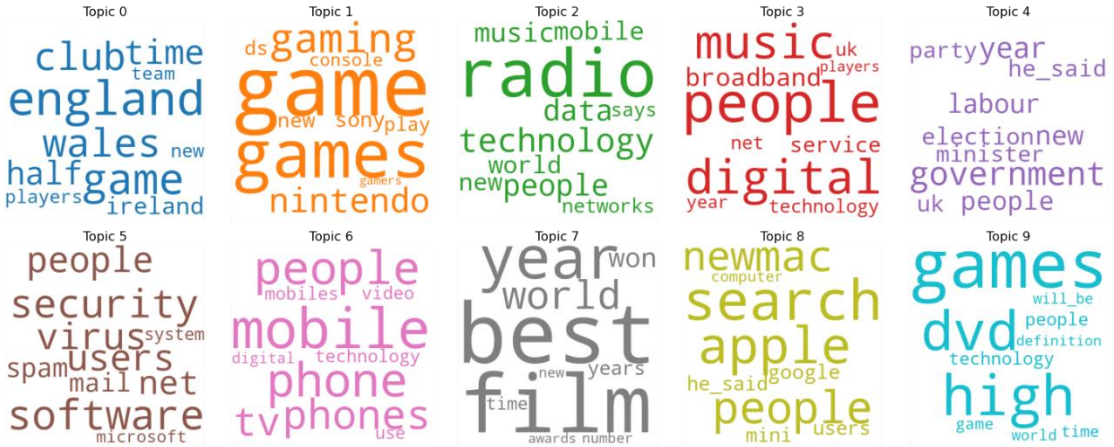
- Bag of Words:
  - 80% of the time we draw from “Sport”
  - 20% of the time we draw from “Business”

- 5-word long BoW
  - Football: 1, Ronaldo: 1, goal: 2
  - Business: revenue: 1
  - “*Football star Ronaldo's revenue grows as he scores goal after goal.*”

LDA

- Given a **collection of documents**
- Given a **number of topics**

- Infer the distribution of topics in documents
- Infer the distribution of words in topics



LDA

- Discover Distribution
  - Of topics in documents
  - Of words in topics

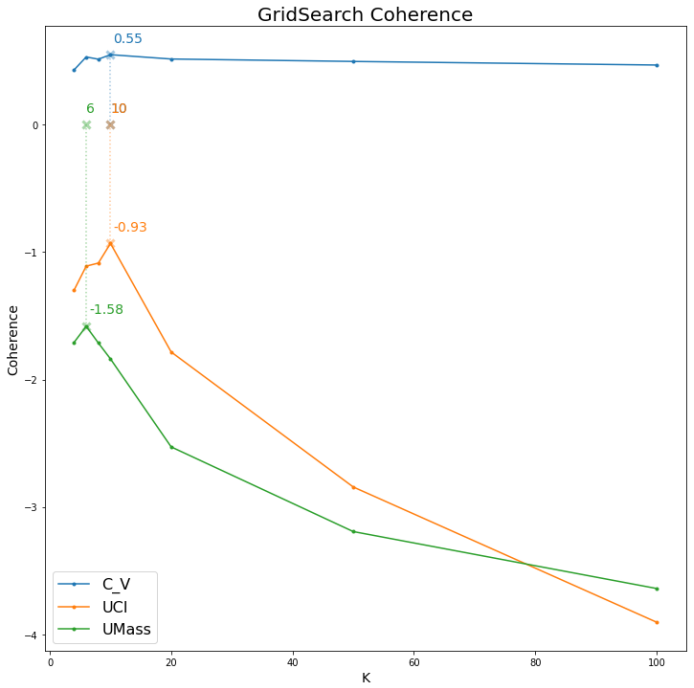
- Machine Learning task
  - Python implementation in gensim or sklearn
  - Details of the learning task are out of scope

LDA - Evaluation

- Intrinsic Evaluation = evaluation of the topic / word distributions

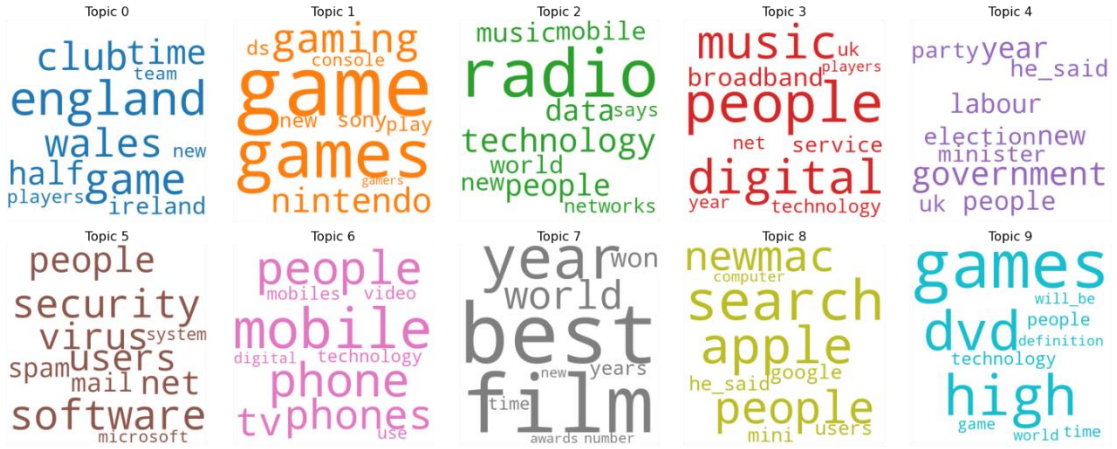
- Coherence metrics
  - Umass, CV, UCI

- Given a collection of documents
  - Try multiple values of K
  - Select highest coherence



LDA

- See the notebook “LDA”




```
Topic # 3 (Internet             ): 0.135
Topic # 4 (Politics             ): 0.620
Topic # 7 (Movies               ): 0.042
Topic # 8 (Apple                ): 0.137
Topic # 9 (Video Technology     ): 0.057
----------------------------------------
Sum                               0.992
```