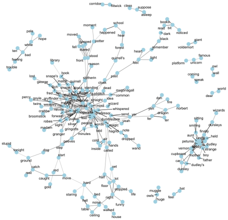
Establishing semantic similarity in text

Alta de Waal

Department of Statistics, University of Pretoria
Centre for Artificial Intelligence (CAIR)

February 21, 2019

image	sound	text
		

Unstructured Text

Challenges

- No meta data available (or at least optional)
- Observations are not labelled
- Veracity and uncertainty
- High dimensional

Unstructured Text

Challenges

- No meta data available (or at least optional)
- Observations are not labelled
- Veracity and uncertainty
- High dimensional

Opportunities

- Larger volumes than labelled text
- Current
- An underlying semantic pattern do exist
- Manifest hypothesis holds true

Manifold hypothesis

Definition

High dimensional spaces tend to lie in the vicinity of underlying lower dimensional spaces (manifolds)

Meaningful Text Analysis

- Descriptive statistics
- Transformations to understand content (and context)
- Retrieve semantically similar documents

1. Preprocess

- Tokenisation
- Vectorisation

Output: Documents represented as a stream of vectors

2. Transformation

- Hidden structure
- Relationship between words
- Compact document representation

Output: Transformed vector space

3. Similarity Queries

- Convert query document to the transformed vector space
- Index query document
- Calculate distance (similarity) between query document

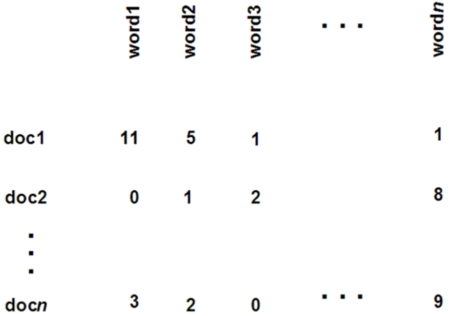
Output: Distance measure

Reading Text Data

1. Data Cleaning and Text Preprocessing
   - Decoding (special characters, emoticons)
   - Remove HTML Markup
   - Split on white space
   - Remove
     - Punctuation, numbers and stopwords
     - Corpus specific stopwords
     - Words occuring $< 1, 2$ in corpus
2. Structure of Text Data
   - Basic unit - word
   - Grouping of words - document
   - Grouping of documents - corpus
3. Descriptive Statistics
   - Total number of words and documents
   - Vocabulary

Bag of words vectorisation

- Count representation of natural text in documents
- First get vocabulary (unique words) from the text
- Count the number of times each word appears in a document



	word1	word2	word3	. . .	word*n*
doc1	11	5	1		1
doc2	0	1	2		8
⋮					
doc*n*	3	2	0	. . .	9

Tweet 1: Julie loves me more than Linda loves me
Tweet 2: Jane likes me more than Julie loves me

Table: Word Count

word	tweet 1	tweet 2
me	2	2
julie	1	1
likes	0	1
loves	2	1
jane	0	1
linda	1	0
than	1	1
more	1	1
Total	8	8

Objectives

- Bring out hidden structure in the corpus
- Discover relationships between words
- Use the relationships to describe the documents in a more semantic way
- Make the document representation more compact

Types of transformation

- Principal Component Analysis
- Term Frequency Inverse Document Frequency (TF-IDF)
- Topic Models
- Word and Sentence Embeddings

Topic Models

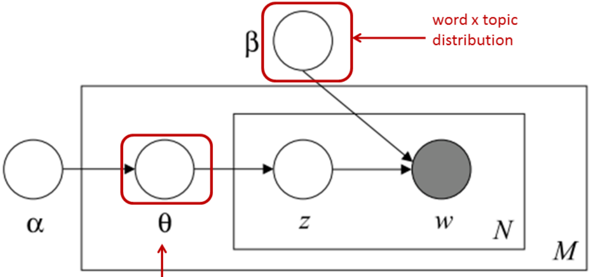
- Algorithms that aim to discover and annotate large archives of documents with thematic information.
- Statistical methods that analyse the words of the original texts to
  - discover the themes that run through them,
  - how those themes are connected to each other, and
  - how they change over time
- Do not require any prior annotations or labelling of the documents
- Topics emerge from the analysis of the original texts.
- Topic modelling enables us to organize and summarize electronic archives at a scale that would be impossible by human annotation.

Probabilistic Topic Models

- Probabilistic models for uncovering the underlying semantic structure of a document collection based on a hierarchical Bayesian analysis of the original texts (Blei, 2003)
- **Aim:** discover patterns of word-use and connect documents that exhibit similar patterns
- **Idea:** documents are mixtures of topics and a topic is a probability distribution over words

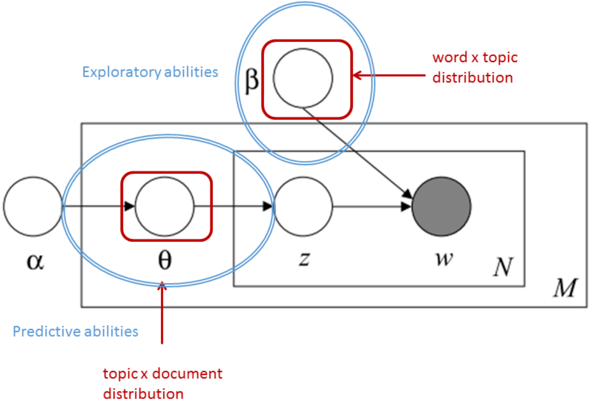
Where clustering and dimensionality reduction meet

- Topic models possess characteristics from both clustering and dimensionality reduction techniques
- A corpus is represented in a lower dimensional form by a set of topics
- Just as with clustering, a document can be associated with a single topic or multiple topics depending on the model.
- Advantage of such methods over clustering, is that “labels” are also produced, in the form of topics.
- They are designed to not only provide data compression, but to also produce topics which are interpretable

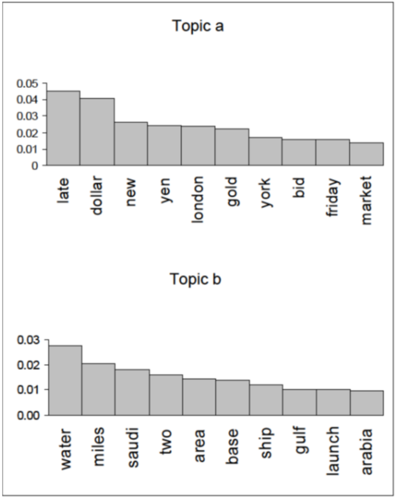


word x topic distribution

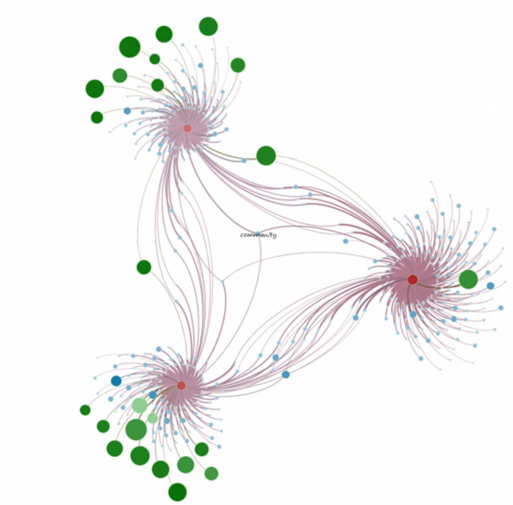
topic x document distribution


Exploratory abilities
β
word x topic
distribution
α
θ
z
w
N
M
Predictive abilities
topic x document
distribution

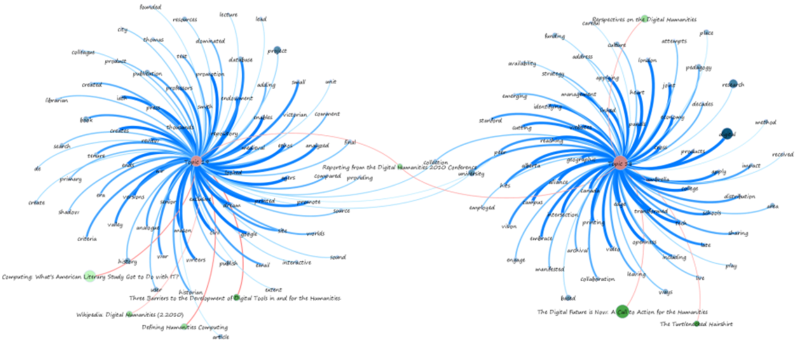
Example of topics



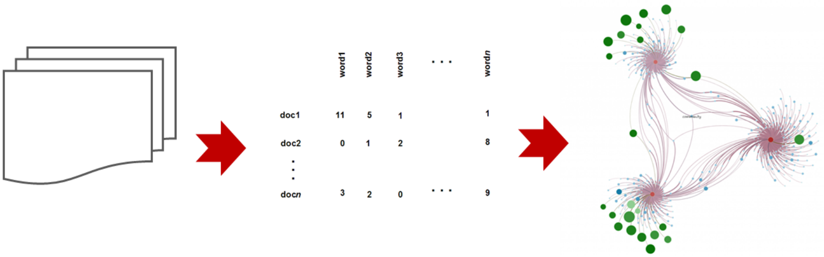
Topic Model - Visualisation 1



Topic Model - Visualisation 2



Process up to now



	word1	word2	word3	. . .	word*n*
doc1	11	5	1		1
doc2	0	1	2		8
⋮					
doc*n*	3	2	0	. . .	9

Natural Language

Vector representation

Manifold representation

Semantic Similarity Interface

Semantic Similarity Interface

Ingredients:

- Query document
- One or more documents to query against (context docs)
- Semantic Structure
- Distance measure

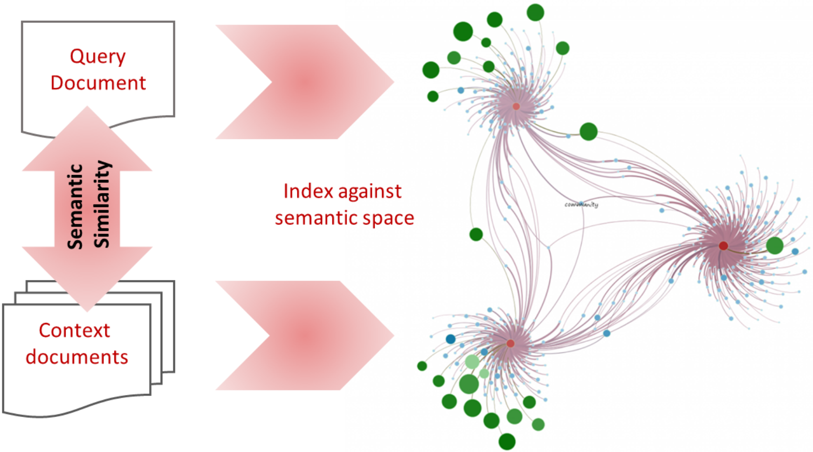
Semantic Similarity Interface

Ingredients:

- Query document
- One or more documents to query against (context docs)
- Semantic Structure
- Distance measure

Steps:

- Vectorise query and context documents
- Index query and context documents
- Calculate distance between query and context documents



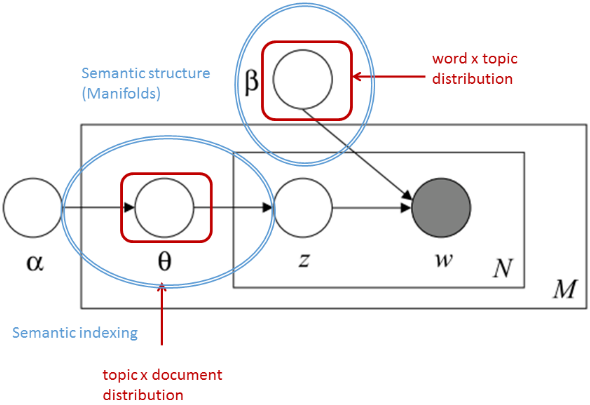
Query Document

Semantic Similarity

Context documents

Index against semantic space

Indexes and Semantic Structure



Semantic structure (Manifolds)

word x topic distribution

Semantic indexing

topic x document distribution

Cosine Similarity

Consider two documents $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$. If we use the bag-of-words representation, then the cosine similarity can be defined as follows:

Definition

Let x_{ij} be the number of times word j occurs in document i.

$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \frac{\boldsymbol{x}_i^T \boldsymbol{x}_{i'}}{||\boldsymbol{x}_i||_2 ||\boldsymbol{x}_{i'}||_2}$

It measures the cosine of the angle between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ when interpreted as vectors. Because $\boldsymbol{x}_i$ is a count vector (non-negative), the cosine similarity is between 0 and 1.

1. Preprocess

- Tokenisation
- Vectorisation

Output: Documents represented as a stream of vectors

2. Transformation

- Hidden structure
- Relationship between words
- Compact document representation

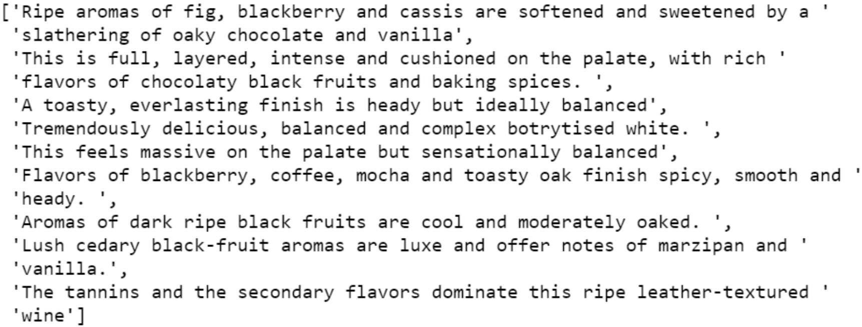
Output: Transformed vector space

3. Similarity Queries

- Convert query document to the transformed vector space
- Index query document
- Calculate distance (similarity) between query document

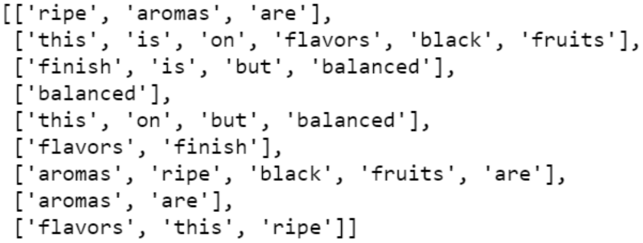
Output: Distance measure

Wine Reviews



```
['Ripe aromas of fig, blackberry and cassis are softened and sweetened by a '
 'slathering of oaky chocolate and vanilla',
 'This is full, layered, intense and cushioned on the palate, with rich '
 'flavors of chocolaty black fruits and baking spices. ',
 'A toasty, everlasting finish is heady but ideally balanced',
 'Tremendously delicious, balanced and complex botrytised white. ',
 'This feels massive on the palate but sensationally balanced',
 'Flavors of blackberry, coffee, mocha and toasty oak finish spicy, smooth and '
 'heady. ',
 'Aromas of dark ripe black fruits are cool and moderately oaked. ',
 'Lush cedary black-fruit aromas are luxe and offer notes of marzipan and '
 'vanilla.',
 'The tannins and the secondary flavors dominate this ripe leather-textured '
 'wine']
```

Wine Reviews

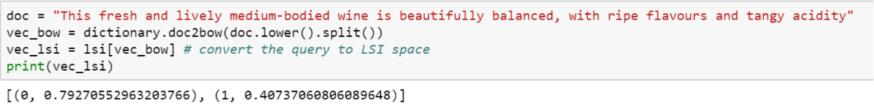


```
[['ripe', 'aromas', 'are'],
 ['this', 'is', 'on', 'flavors', 'black', 'fruits'],
 ['finish', 'is', 'but', 'balanced'],
 ['balanced'],
 ['this', 'on', 'but', 'balanced'],
 ['flavors', 'finish'],
 ['aromas', 'ripe', 'black', 'fruits', 'are'],
 ['aromas', 'are'],
 ['flavors', 'this', 'ripe']]
```

Exploring the semantic structure

topic 1	**topic 2**
are	balanced
aroma	but
ripe	finish
fruits	are
black	aroma

Query - a new (unseen to the model) review



```
doc = "This fresh and lively medium-bodied wine is beautifully balanced, with ripe flavours and tangy acidity"
vec_bow = dictionary.doc2bow(doc.lower().split())
vec_lsi = lsi[vec_bow] # convert the query to LSI space
print(vec_lsi)
```

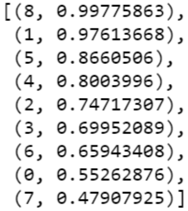
```
[(0, 0.79270552963203766), (1, 0.40737060806089648)]
```

To which review is the query most similar to?

"This fresh and lively medium-bodied wine is beautifully balanced, with ripe flavours and tangy acidity"

To which review is the query most similar to?

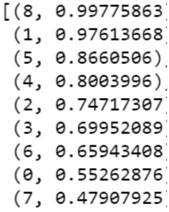
"This fresh and lively medium-bodied wine is beautifully balanced, with ripe flavours and tangy acidity"



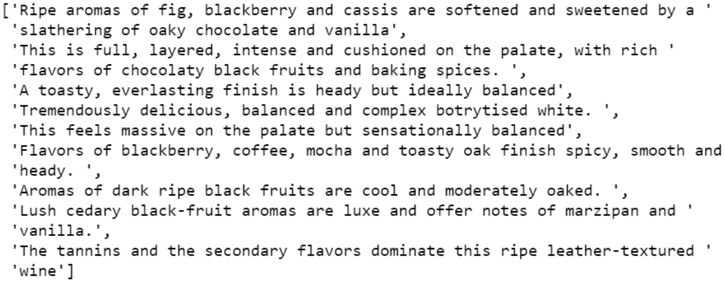
```
[(8, 0.99775863),
 (1, 0.97613668),
 (5, 0.8660506),
 (4, 0.8003996),
 (2, 0.74717307),
 (3, 0.69952089),
 (6, 0.65943408),
 (0, 0.55262876),
 (7, 0.47907925)]
```

To which review is the query most similar to?

“This fresh and lively medium-bodied wine is beautifully balanced, with ripe flavours and tangy acidity”



```
[(8, 0.99775863
 (1, 0.97613668
 (5, 0.8660506)
 (4, 0.8003996)
 (2, 0.74717307
 (3, 0.69952089
 (6, 0.65943408
 (0, 0.55262876
 (7, 0.47907925
```



```
['Ripe aromas of fig, blackberry and cassis are softened and sweetened by a '
 'slathering of oaky chocolate and vanilla',
 'This is full, layered, intense and cushioned on the palate, with rich '
 'flavors of chocolaty black fruits and baking spices. ',
 'A toasty, everlasting finish is heady but ideally balanced',
 'Tremendously delicious, balanced and complex botrytised white. ',
 'This feels massive on the palate but sensationally balanced',
 'Flavors of blackberry, coffee, mocha and toasty oak finish spicy, smooth and
 'heady. ',
 'Aromas of dark ripe black fruits are cool and moderately oaked. ',
 'Lush cedary black-fruit aromas are luxe and offer notes of marzipan and '
 'vanilla.',
 'The tannins and the secondary flavors dominate this ripe leather-textured '
 'wine']
```

Resources

- Visual Cinnamon
- Data Visualization Society
- Significance Magazine
- Twitter
  - Mara Averick (@dataandme)
  - Frederica Fragapane (@fredfragapane)

Music Lyric Analysis

Datacamp Tutorial