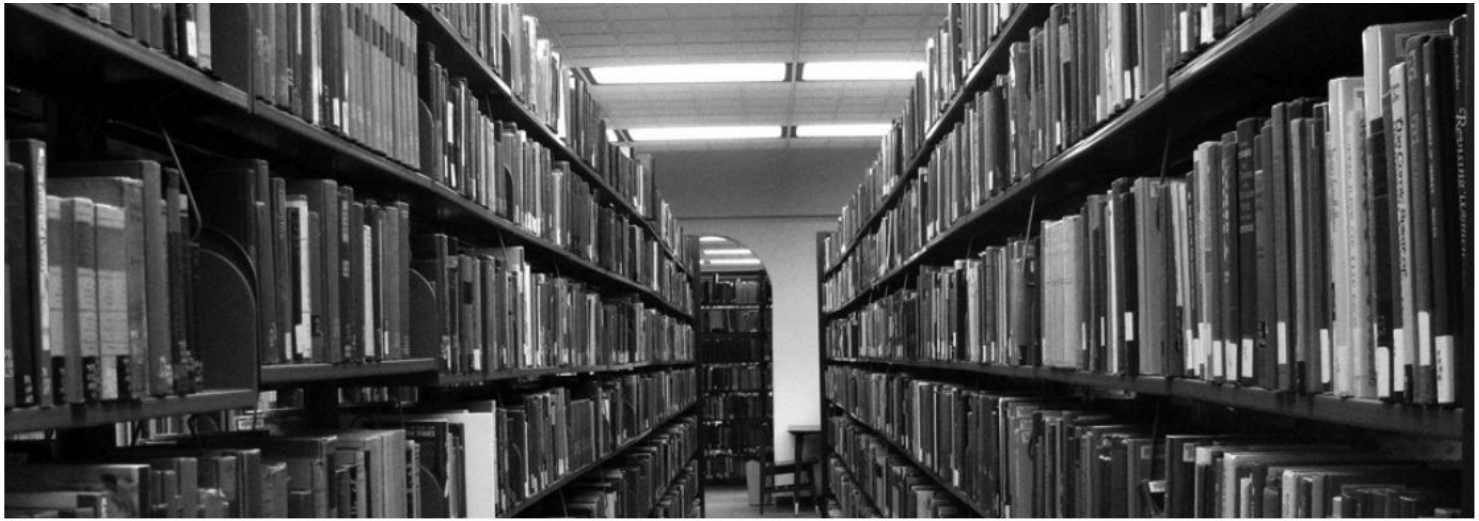
Topic modeling and visualization of news comments

Goal and Motivation



Input: An unorganized collection of documents

Output: An organized collection, and a description of how

Goal: To extract topics from news articles and their comments, and to perform visualizations of the results. Final product will contain the visualizations depicting the topic based insights relating the articles and the comments. Grouping users based on the topics they relate to. Change of sentiment due to comments, type of authors and the complexity of threads.

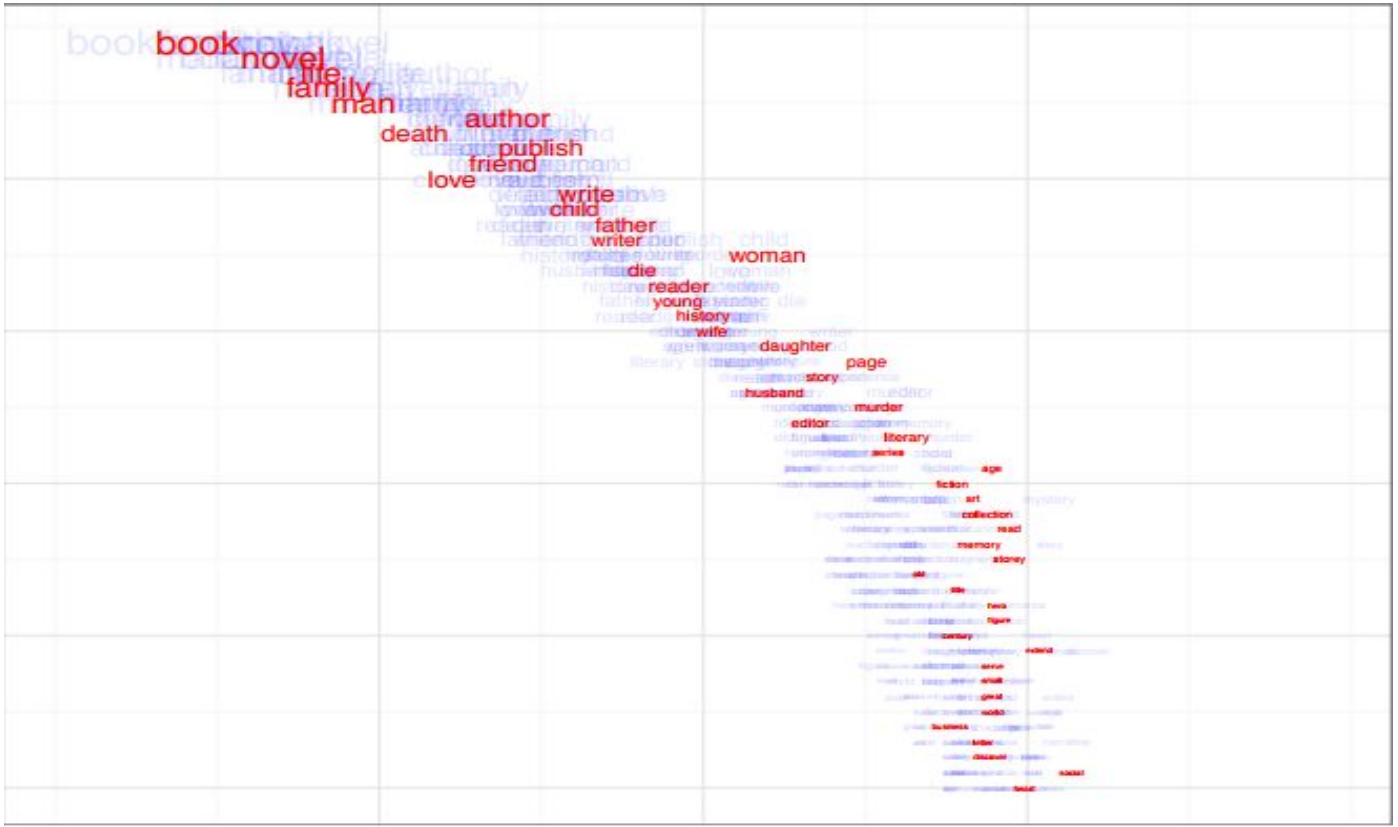
Motivation for topic modeling:

- Identify trending topics in the news and how they evolve
- Help people find documents that they are interested in ✏ Learn about how the documents are implicitly organized
- Explore user sentiments toward the articles
- Understand behavior of commentators (engagement, sentiments, etc)
- Create an automatic tool for annotating and summarizing documents and generating outputs for the above purposes

Challenges

- How do we select and revise?
  - Which model should be chosen for a problem?
  - Where does our model go right? Where does it go wrong?
  - Model evaluation
- Natural language processing is ambiguous
  - Variations in words: am, are, is, be
  - Operate operating operates operation operative operatives operational
- Polysemy: A world record. A record of the conversation. Record it
- Co-Reference: John put the cake on the plate and ate it.
- Comments/Articles Clean up is required

Source of Data



SFU Linguistic Department

The SFU Opinion and Comments Corpus (SOCC) is part of a project a that investigates the linguistic characteristics of online comments.

- 10,339 opinion Globe and Mail articles (editorials, columns, and op-eds)
- 663,173 comments in response to the articles
- 303,665 comment threads in response to the articles
- a subset of SOCC for constructiveness and toxicity.

Source: https://researchdata.sfu.ca/islandora/object/islandora%3A9109

Questions we want to answer…

1. What are the top topics mentioned in the articles and comments? What is the overlap?

SKY WATER TREE
MOUNTAIN PEOPLE

SCOTLAND WATER
FLOWER HILLS TREE

SKY WATER BUILDING
PEOPLE WATER

FISH WATER OCEAN
TREE CORAL

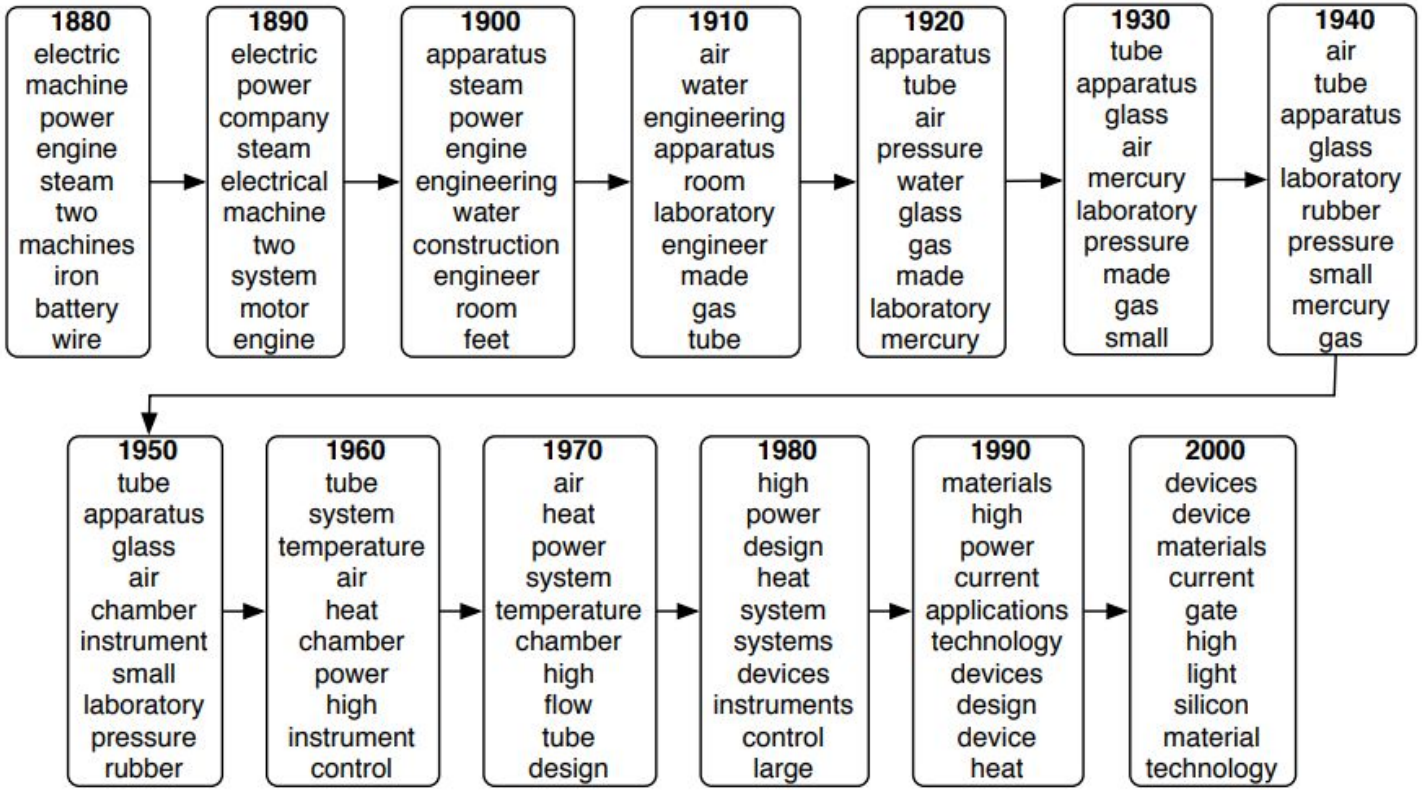
PEOPLE MARKET PATTERN
TEXTILE DISPLAY

BIRDS NEST TREE
BRANCH LEAVES

Pipeline (Topic Modelling and Overlap):

- Text cleaning: tokenizing, removing stop words, stemming, lemmatizing
- Creating the term-document matrix
- LDA model (NLTK and Gensim) - a probabilistic approach
- NMF model (Scikit-Learn) - a matrix factorization approach (SVD)
- Train the above models for articles and comments.
- Find the overlap using methods such as Jaccard similarity
- Visualize the results - Ex. A histogram of the topics in the articles and comments, word clouds, etc

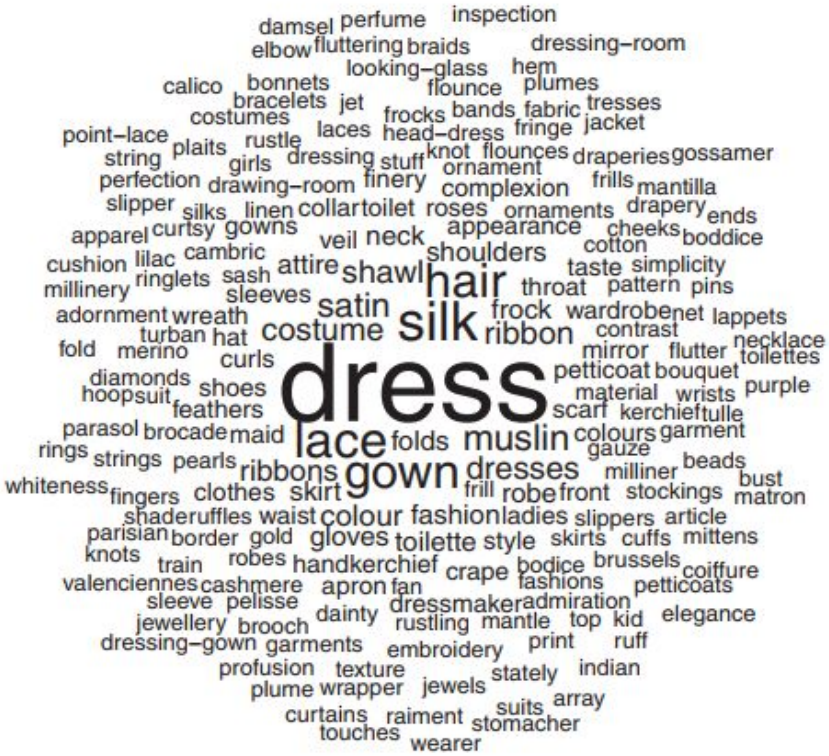
2. What are the most mentioned people, places, organizations in the articles and comments? What is the overlap?



Pipeline (Named Entity Recognition):

- Text cleaning: tokenizing, removing stop words, stemming, lemmatizing
- **Named entity recognition: Use pre-trained models to identify people, places, organizations mentioned in articles and comments**
- Visualize the results - Ex. A histogram of the entities mentioned, the overlapping entities, etc

3. What are the sentiments expressed in the comments for the articles? Who are the most optimistic and pessimistic commentators?



Pipeline (Sentiment Analysis):

- Use the annotated data with labels (constructiveness, toxicity)
- Text cleaning: tokenizing, removing stop words, stemming, lemmatizing
- Use NLTK library to predict the sentiment (positive, negative, neutral) in the form of sentences from the articles and comments.
- Find people from the NER in the positive/negative/neutral sentences above to link them to optimism/pessimism.
- Vectorize authors using sentiment analysis linking the entities found and the topics written on.
- Use the results to cluster (density-based) authors.

Tools and Instruments

GraphLab

Spark

- For parallelizing the data processing.

scikit learn

- Topic modeling using LDA and NMF

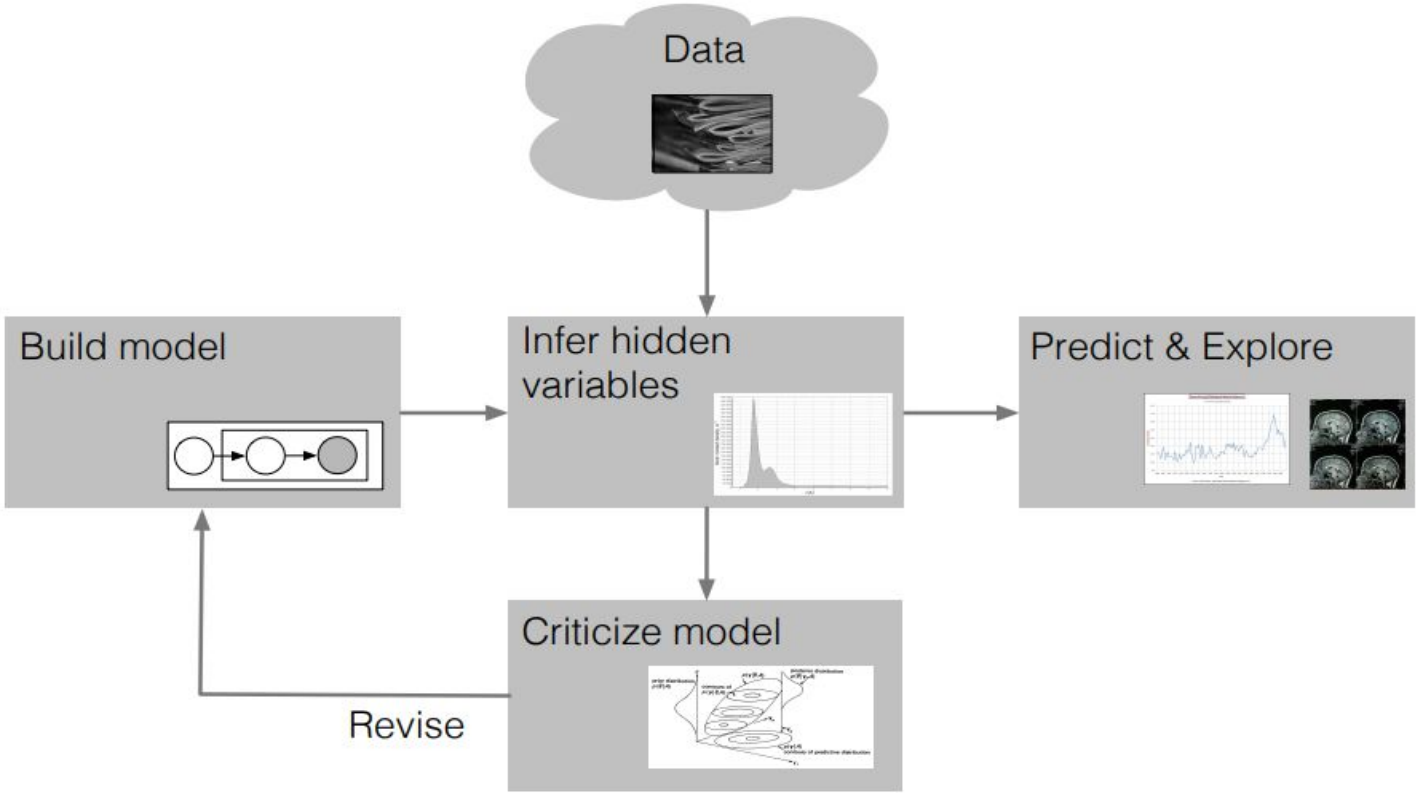
Natural Language Toolkit (NLTK)

- Sentiment analysis, Named Entity Recognition

TensorFlow™

- Speed up learning using Pre-trained models

Models comparison



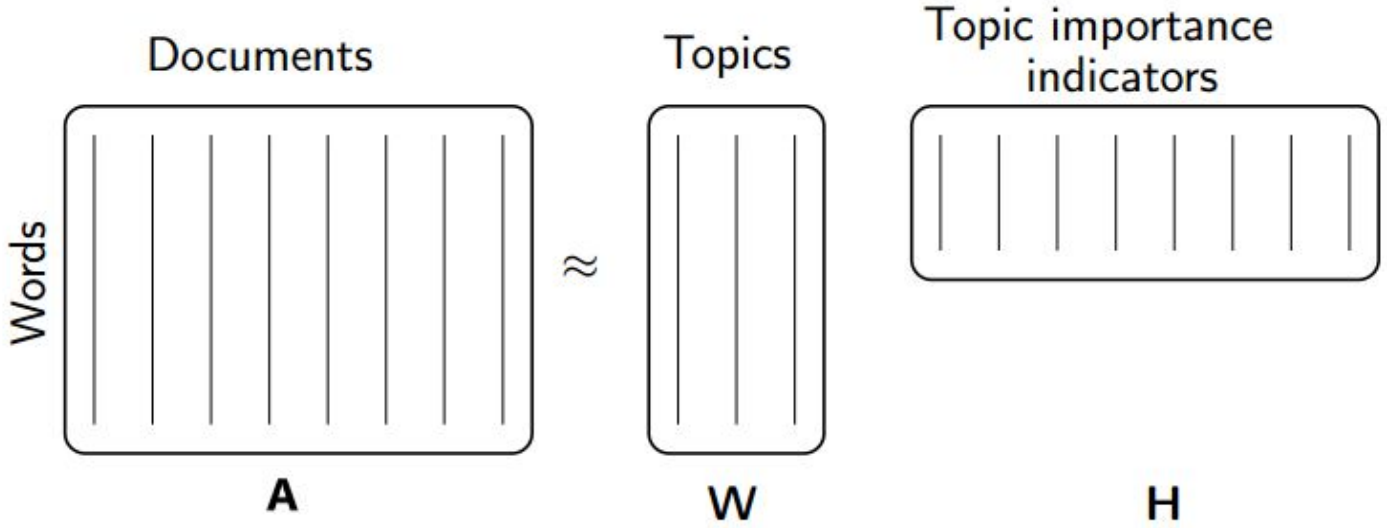
NMF (Non-Negative Matrix Factorization) vs. **LDA** (Latent Dirichlet Allocation)

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	157	73	0	0	0	1
BRUTUS	4	157	0	2	0	0
CAESAR	232	227	0	2	1	0
CALPURNIA	0	10	0	0	0	0
CLEOPATRA	57	0	0	0	0	0
MERCY	2	0	3	8	5	8
WORSER	2	0	1	1	1	5

NMF

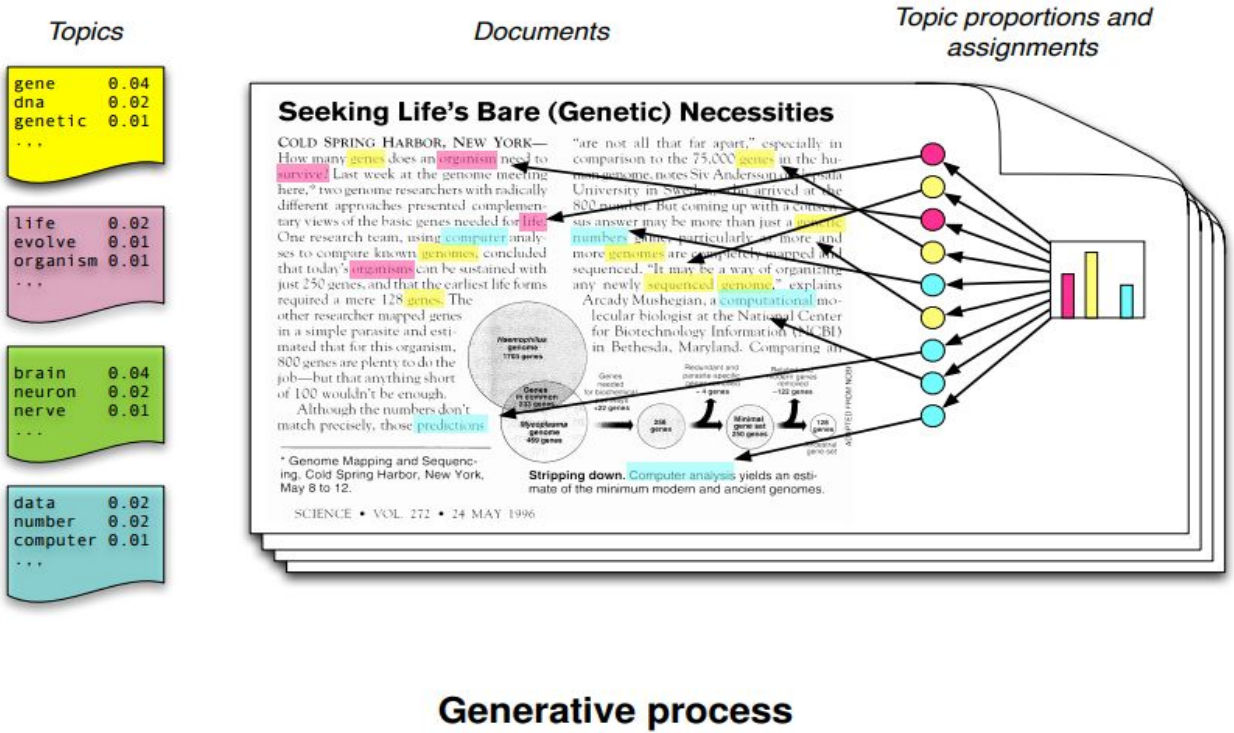
Non-negative Matrix Factorization is a Linear-algebraic model, that factors high-dimensional vectors into a low-dimensionality representation. Similar to Principal component analysis (PCA).

Given the original matrix **A**, we can obtain two matrices **W** and **H**, such that A= WH.

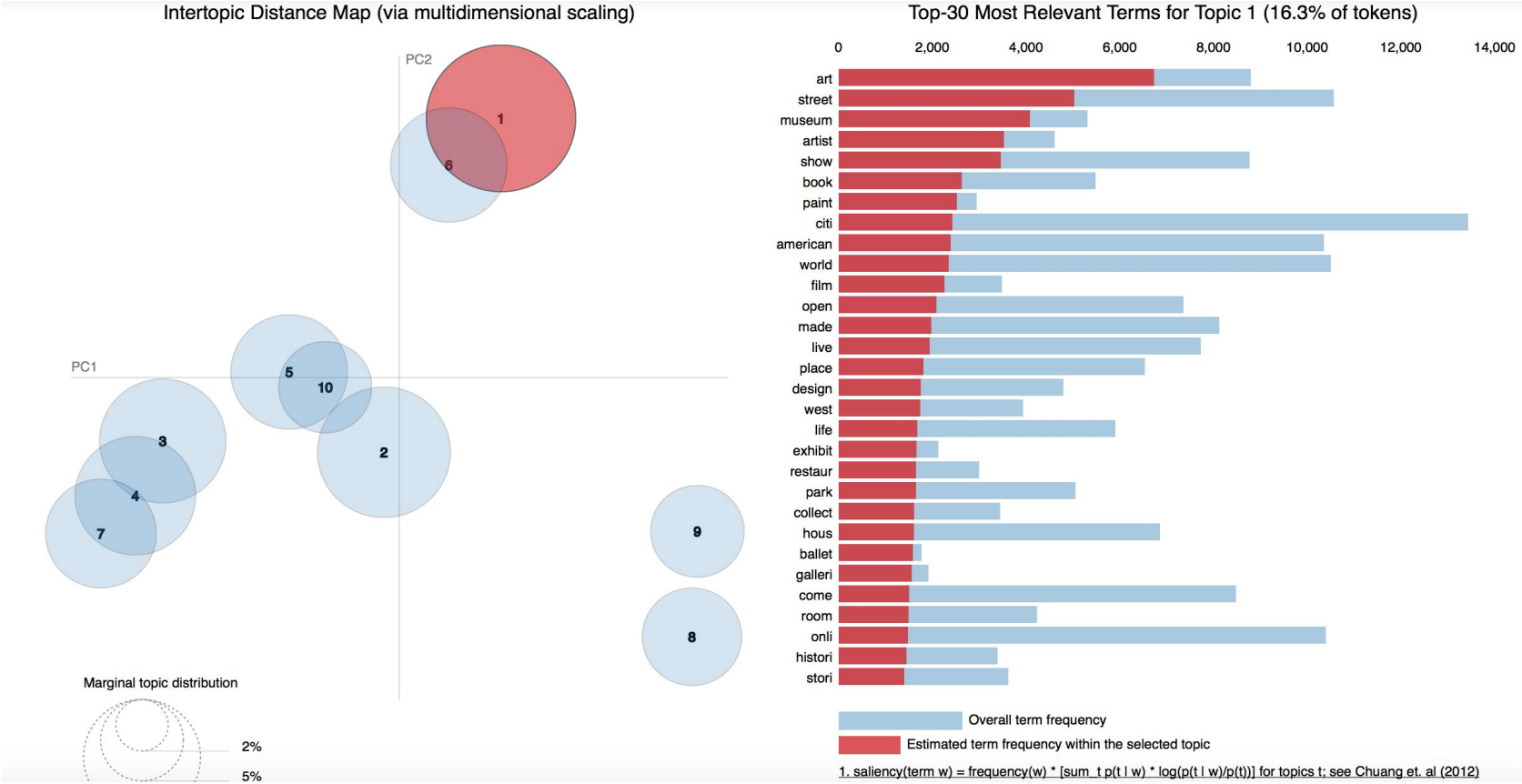


LDA

LDA, or Latent Derelicht Analysis is a probabilistic model, and to obtain cluster assignments, it uses two probability values: **P(word | topics)** and **P(topics | documents)**.



Example of the final visualization (PyLDAvis)



Thank you !

Charles Darwin's library

Backup Slides

NMF (Non-Negative Matrix Factorization) vs LDA (Latent Dirichlet Allocation)

A. Non-Negative Matrix Factorization

Non-Negative Matrix factorization or NMF, is a linear-algebraic optimization algorithm. One of its properties is that it can extract meaningful information about topics without prior knowledge of the underlying meaning in the data. The mathematical objective of NMF is to decompose a single ‘ n×m ‘ input matrix into two matrices such that their product is a close estimate to the input matrix. For topic modeling, the input matrix of choice is the document-term matrix. This matrix is factorized into the document-topic matrix, of dimensions ‘ n×t, and a topic-term matrix, of dimensions *‘t* ×m', where *‘t’* is the number of topics that are to be produced. The NMF clustering algorithm has been employed successfully, mainly because it can be adapted to specific application such as natural language processing [13].

B. Latent Dirichlet Allocation

Latent Dirichlet Allocation or LDA, is a generative probabilistic model largely used for topic modeling. LDA is a three-level hierarchical Bayesian model, within which every item of a corpus is modeled as a finite mixture over an underlying set of topics. Each topic is then modeled as an infinite mixture over an underlying set of topic probabilities. There topic possibilities offer a specific illustration of a document [14]. LDA represents documents as a mixture of topics that contain words with certain probabilities of occurrence. Given a collection of documents, some fixed number of topics to find, LDA learns the topic representation of each document and therefore the words associated to each topic via an iterative procedure. LDA then tries to backtrack from the documents to seek out a collection of topics that are likely to have generated the collection [12].

LDA (cont.)

https://medium.com/ml2vec/topic-modeling-is-an-unsupervised-learning-approach-to-clustering-documents-to-discover-topics-fdfbf30e27df

LDA and LSI both describe mathematical models that are designed to be used for information retrieval – i.e. returning search results. LDA is a significant extension of LSI.

LDA, or Latent Derelicht Analysis is a probabilistic model, and to obtain cluster assignments, it uses two probability values: **P(word | topics)** and **P(topics | documents).** Three small documents, 2 prominent topics, food and pets.

Document 1		Document 2		Document 3	
Eat	A	Cat	B	Cat	B
Fish	A	Dog	B	Eat	A
Vegetables	A	Pet	B	Fish	?
Fish	A	Pet	B	Cat	B
Eat	A	Fish	B	Fish	A

LDA (cont.)

These values are calculated based on an initial random assignment, after which they are repeated for each word in each document, to decide their topic assignment. In an iterative procedure, these probabilities are calculated multiple times, until the convergence of the algorithm.

Document 1 is primarily about food (Topic A), Document 2 is about pets (Topic B), and Document 3 is evenly split between A and B, with one un-classified word Fish. Using the existing information, we will obtain the topic to which Fish in Document 3 should be classified to.

First, we will calculate what the probability is of the word appearing in the different topics:

P(‘Fish’ | topic A) = 0.75, P(‘Fish’ | topic B) = 0.25

Now, we need the probability of the topics in the document with the word in it, which is going to be: P(topic A | Document 3) = P(topic B | Document 3) = 0.5, because they are evenly split.

Weighing the conclusions from both probabilities, we will assign the word ‘Fish’ in Document 3 to topic A. The next step of the algorithm would be to repeat this step for all words in each document, and then repeat the entire classification multiple times. With multiple iterations, we will obtain better and better topic classifications each time because we will have more updated information for each document.

NMF (cont.)

We calculate W and H by optimizing over an objective function (like the EM algorithm), updating both W and H iteratively until convergence.

$$\frac{1}{2}\,||\mathbf{A}-\mathbf{WH}||_{\mathsf{F}}^{2}=\sum_{i=1}^{n}\sum_{j=1}^{m}(A_{ij}-(WH)_{ij})^{2}$$

Sentiment Analysis

Sentiment analysis – otherwise known as opinion mining – is a much bandied about but often misunderstood term.

In essence, it is the process of determining the emotional tone behind a series of words, used to gain an understanding of the the attitudes, opinions and emotions expressed within an online mention.

Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics.