Understanding and Using Topic Modeling

Using inferred document clusters

Philip Robinson

Presented to OpsLab
NASA - Jet Propulsion Lab

August 24, 2018

Jet Propulsion Laboratory
California Institute of Technology

Introduction - Philip (1762)

Computer Science MSc at Oregon Health and Science University.

- Whale vocalization
- Light pollution
- Modern cryptography
- ⋆ Information retrieval

Presentation Overview

What is Topic Modeling

Topic modeling is a text processing technique for automatically grouping documents by topics. This is usually used as a strategy to describe documents in low dimensional space or an exploratory tool for document collections.

Examples

In practice, this requires many more documents

The Tourist huddles in the station While slowly night gives way to dawn ; He finds a certain fascination In knowing all the trains are gone.

The Governess up in the attic Attempts to make a cup of tea ; Her mind grows daily more erratic From cold and hunger and ennui.

The Journalist surveys the slaughter, The best in years without a doubt; He pours himself a gin and water and wonders how it came about.

- Food
- Travel
- Time

From this annotation we know that Document 2 and 3 are about Food and Time

What can I solve?

- Find similar document pairs
- Cluster documents into groups with similar content
- Find relevent documents to a query or user's interests
- Explore shape of document collection
- Can be extended to address many other problems

Applications of Topic Modeling

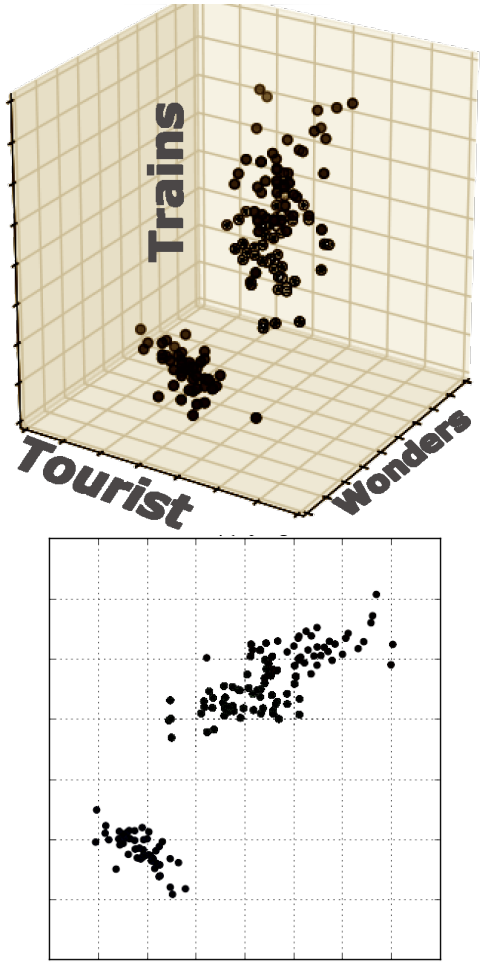


Figure: Dimmentionality Reduction

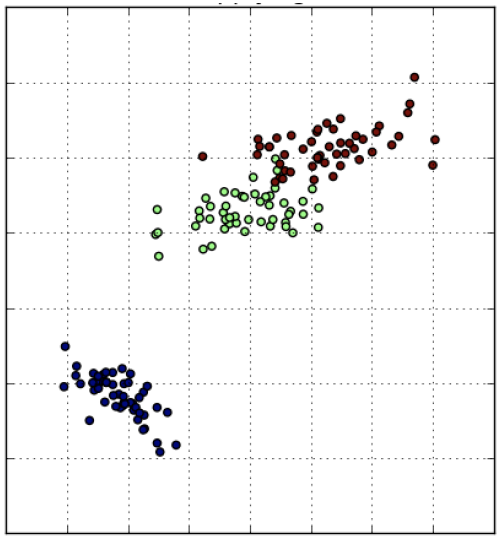


Figure: Cluster Points

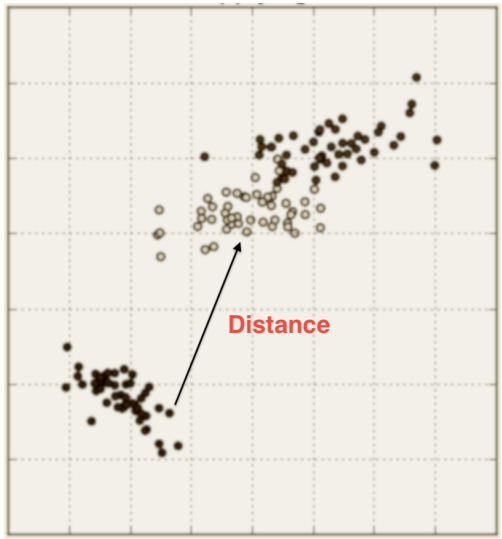


Figure: Exploratory Data Analysis

Latend Dirichlet Allocation (LDA)

A generative model

- Assume/Generalize how documents could have been generated
- Fit parameters that describe generalization
- Ask questions about the generalization in relation to documents
- Ask questions about documents in relation to the generalization

Topic modeling generalizes how a document is generated by claiming that words come from topics, and documents have multiple topics. Note that this ignores sentence structure, entities, authorship, or other things we may care about.

Topic Modeling

- Represent document as Bag-of-Words[1]
- Model/Fit topics as mixture of words
- Documents are projected into topic space
- Study relationships between document projections

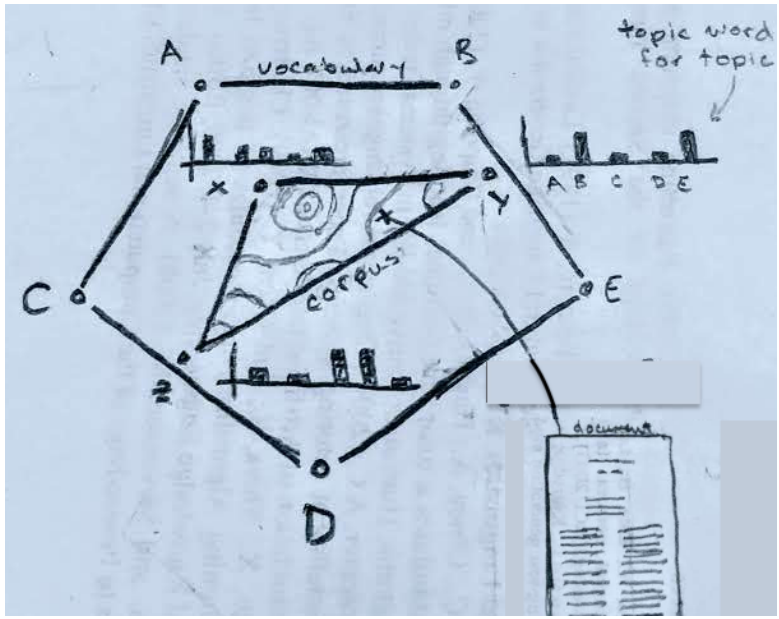


Figure: Latent Dirichlet Allocation

$$T(x) = \texttt{Project } x \texttt{ into topic-space}$$

$$\textit{Measure} = \textit{Distance}(T(\texttt{Doc 1}), T(\texttt{Doc 2}))$$

[1] equivelent to multinomial over the vocabulary

Dirichlet Distribution

In this case the Topic-Space is our Dirichlet Distribution

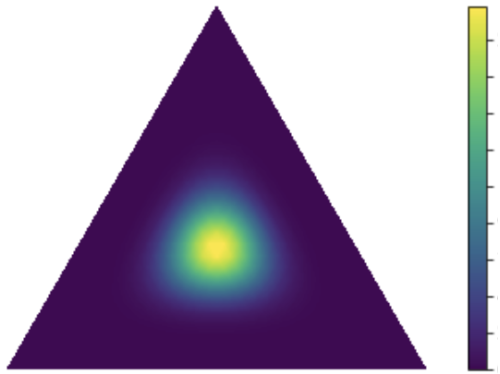


Figure: Non-Informative

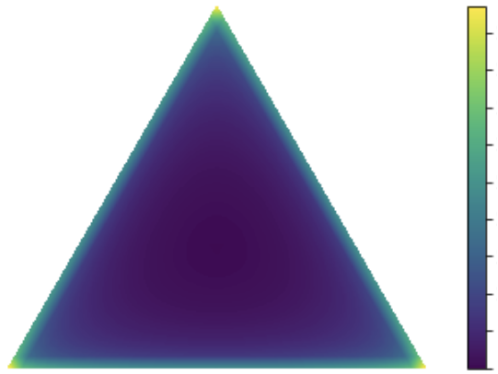


Figure: Little In Common

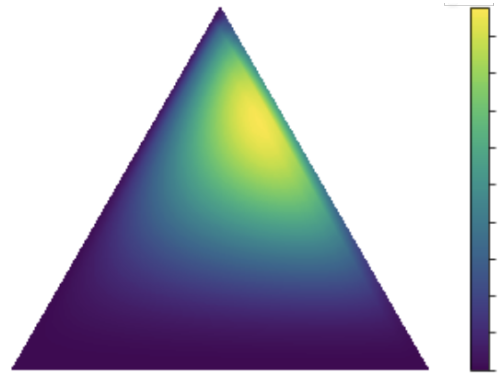


Figure: Shared Topics

Using this in Python[2]

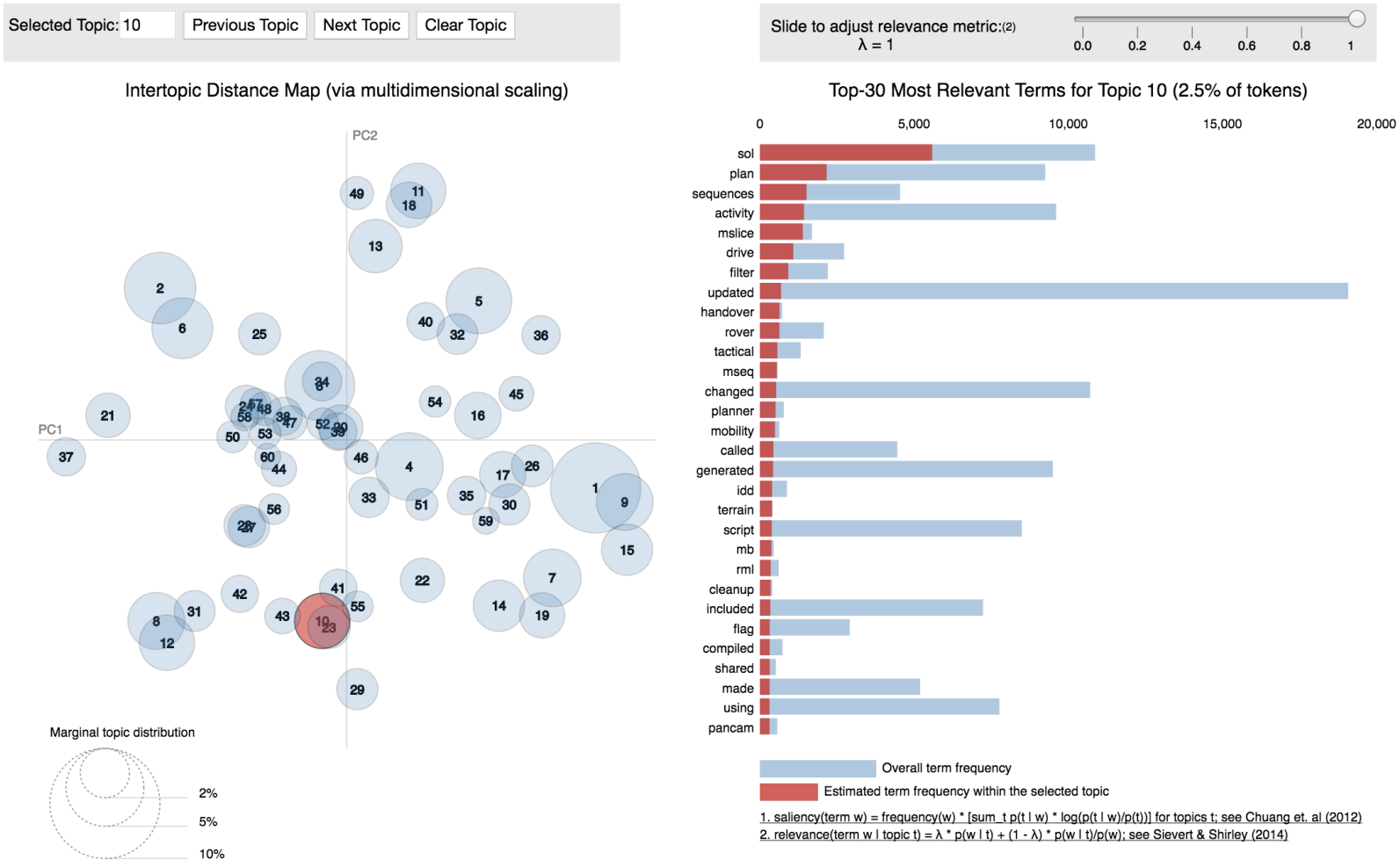
```
from nltk.corpus import brown

data = []

for fileid in brown.fileids():
    document = ' '.join(brown.words(fileid))
    data.append(document)

NO_DOCUMENTS = len(data)
print(NO_DOCUMENTS)
print(data[:5])

from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import CountVectorizer

NUM_TOPICS = 10

vectorizer = CountVectorizer(min_df=5, max_df=0.9,
                             stop_words='english', lowercase=True,
                             token_pattern='[a-zA-Z\-][a-zA-Z\-]{2,}')
data_vectorized = vectorizer.fit_transform(data)

# Build a Latent Dirichlet Allocation Model
lda_model = LatentDirichletAllocation(n_topics=NUM_TOPICS, max_iter=10, learning_method='online')
lda_Z = lda_model.fit_transform(data_vectorized)

text = "The economy is working better than ever"
x = lda_model.transform(vectorizer.transform([text]))[0]
```

[2] example in `scikit-learn`, I used `gensim`

Concerns about Topics

- Topics do not have labels
- Topics are not easily human interpretable
- In traditional LDA n-grams are not represented

Visualizing High Dimensional Space[3]



[3] used `pyLDAvis`

Looking at Topics

Breakout to Jupyter

Evaluation Notes

As this is a bayesian machine learning approach we will need to evaluate model fit.

- perplexity
- coherence
- visualization
- predictive power

Preprocessing Notes

As this model generates conditional probabilities on observed words, we need a clean/normalize vocabulary

- Lowercase corpus
- Tokenization & (Stemming — Lemmatization)
- Removing symbols
- Remove highly common words
- Remove extreamly rare words

Conclusion & Questions

I used this technique during my internship to automatically assign tickets to subject matter experts, for the Office of Safety and Mission Success (5x)[4]

[4] 1762 acts as data science support to JPL