



Text Analysis



Scott Pletcher
INSTRUCTOR

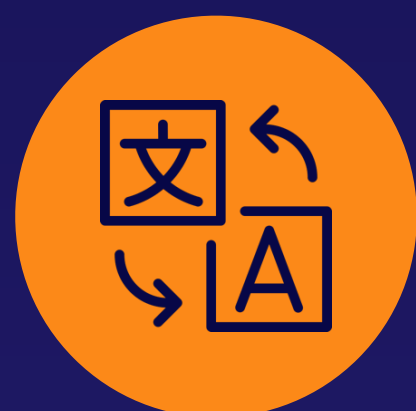
Latent Dirichlet Allocation (LDA)



Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) algorithm is an unsupervised learning algorithm that attempts to describe a set of observations as a mixture of distinct categories. LDA is most commonly used to discover a user-specified number of topics shared by documents within a text corpus. Here each observation is a document, the features are the presence (or occurrence count) of each word, and the categories are the topics.

{UNSUPERVISED}



Used to figure out how similar documents are based on the frequency of similar words.

Latent Dirichlet Allocation (LDA) - Use Cases

- **Article Recommendation**

Example: Recommend articles on similar topics which you might have read or rated in the past.

- **Musical Influence Modeling**

Example: Explore which musical artists over time were truly innovative and those who were influenced by those innovators.

Latent Dirichlet Allocation (LDA) - Use Cases

Modeling Musical Influence with Topic Models

Uri Shalit
ICNC-ELSC & Computer Science Department, The Hebrew University of Jerusalem, 91904 Jerusalem Israel

URI.SHALIT@MAIL.HUJI.AC.IL

Daphna Weinshall
Computer Science Department, The Hebrew University of Jerusalem, 91904 Jerusalem Israel

DAPHNA@CS.HUJI.AC.IL

Gal Chechik
The Gonda Brain Research Center, Bar Ilan University, 52900 Ramat-Gan, Israel

GAL.CHECHIK@BIU.AC.IL

Abstract

The role of musical influence has long been debated by scholars and critics in the humanities, but never in a data-driven way. In this work we approach the question of influence by applying topic-modeling tools (Blei & Lafferty, 2006; Gerrish & Blei, 2010) to a dataset of 24941 songs by 9222 artists, from the years 1922 to 2010. We find the models to be significantly correlated with a human-curated influence measure, and to clearly outperform a baseline method. Further using the learned model to study properties of influence, we find that musical influence and musical innovation are not monotonically correlated. However, we do find that the most influential songs were more innovative during two time periods: the early 1970’s and the mid 1990’s.

bilities to study the structure of the music collective itself. Specifically, this paper provides a quantitative modeling approach to study musical influence. Musical influence is often discussed, but has never been studied quantitatively and at a large scale before.

Although central to understanding musical creation, the concept of musical influence is loosely defined, and its role debated among scholars of art history and cultural critics. For instance Bloom claimed that important artistic work results when an artist creates original work against existing influence (“The Anxiety of Influence”, Bloom, 1997), while Lethem claimed that “originality and appropriations are as one” in all artistic endeavor, (“The Ecstasy of Influence”, Lethem, 2007). In a cultural-artistic landscape that is very much shaped by sampling, remixing, and copy-pasting, the question of the role of influence in art is always present (Reynolds, 2011). Unfortunately these questions were never studied in a data-driven way.



Neural Topic Model (NTM)



Neural Topic Model

Unsupervised learning algorithm that is used to organize a corpus of documents into topics that contain word groupings based on their statistical distribution. Topic modeling can be used to classify or summarize documents based on the topics detected or to retrieve information or recommend content based on topic similarities.



Similar uses and function to LDA in that both NTM and LDA can perform topic modeling. However, NTM uses a different algorithm which might yield different results than LDA.

{UNSUPERVISED}

Sequence to Sequence (seq2seq)



Sequence to Sequence

Supervised learning algorithm where the input is a sequence of tokens (for example, text, audio) and the output generated is another sequence of tokens.

{SUPERVISED}



Think a language translation engine that can take in some text and predict what that text might be in another language. We must supply training data and vocabulary.

Sequence to Sequence (seq2seq)

1

**Steps consist of embedding,
encoding and decoding**

Using a neural network model (RNN and CNN), the algorithm uses layers for embedding, encoding and decoding into the target.

Sequence to Sequence (seq2seq)

1

Steps consist of embedding, encoding and decoding

Using a neural network model (RNN and CNN), the algorithm uses layers for embedding, encoding and decoding into the target.

2

Commonly initialized with pre-trained word libraries.

A standard practice is initializing the embedding layer with a pre-trained word vector like FastText or Glove or to initialize it randomly and learn the parameters during training.

Sequence to Sequence (seq2seq)

1

Steps consist of embedding, encoding and decoding

Using a neural network model (RNN and CNN), the algorithm uses layers for embedding, encoding and decoding into the target.

2

Commonly initialized with pre-trained word libraries.

A standard practice is initializing the embedding layer with a pre-trained word vector like FastText or Glove or to initialize it randomly and learn the parameters during training.

3

Only GPU instances are supported.

Currently Amazon SageMaker seq2seq is only supported on GPU instance types and is only set up to train on a single machine. But it does also offer support for multiple GPUs.

Sequence to Sequence (seq2seq)

- **Language Translations**

Example: Using a vocabulary, predict the translation of a sentence into another language.

- **Speech to Text**

Example: Given an audio “vocabulary”, predict the textual representation of spoken words.



BlazingText

Highly optimized implementations of the Word2vec and text classification algorithms. The Word2vec algorithm is useful for many downstream natural language processing (NLP) tasks, such as sentiment analysis, named entity recognition, machine translation, etc.

{SUPERVISED}
{UNSUPERVISED}



Really, really optimized way to determine contextual semantic relationships between words in a body of text.

| Modes | Word2Vec (Unsupervised) | Text Classification (Supervised) |
|---|---|-------------------------------------|
| Single CPU Instance | Continuous Bag of Words Skip-gram Batch Skip-gram | Supervised |
| Single GPU Instance (1 or more GPUs) | Continuous Bag of Words Skip-gram | Supervised with 1 GPU |
| Multiple CPU Instances | Batch Skip-gram | None |

1

Expects single pre-processed text file

Each line in the file should contain a single sentence. If you need to train on multiple text files, concatenate them into one file and upload the file in the respective channel.

1

Expects single pre-processed text file

Each line in the file should contain a single sentence. If you need to train on multiple text files, concatenate them into one file and upload the file in the respective channel.

2

Highly Scalable

Improves on traditional Word2Vec algorithm by supporting scale-out for multiple CPU instances. FastText text classifier can leverage GPU acceleration.

1

Expects single pre-processed text file

Each line in the file should contain a single sentence. If you need to train on multiple text files, concatenate them into one file and upload the file in the respective channel.

2

Highly Scalable

Improves on traditional Word2Vec algorithm by supporting scale-out for multiple CPU instances. FastText text classifier can leverage GPU acceleration.

3

Around 20x faster than FastText

Supports pre-trained FastText models but also can perform training about 20x faster than FastText.

- **Sentiment Analysis**

Example: Evaluate customer comments in social media posts to evaluate whether they have a positive or negative sentiment.

- **Document Classification**

Example: Review a large collection of documents and detect whether the document should be classified as containing sensitive data like personal information or trade secrets.

- **Sentiment Analysis**

Example: Evaluate customer comments in social media posts to evaluate whether they have a positive or negative sentiment.



**Amazon
Comprehend**

- **Document Classification**

Example: Review a large collection of documents and detect whether the document should be classified as containing sensitive data like personal information or trade secrets.

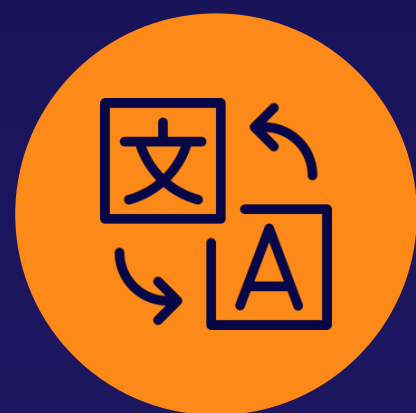


**Amazon
Macie**



Object2Vec

General-purpose neural embedding algorithm that can learn low-dimensional dense embeddings of high-dimensional objects while preserving the semantics of the relationship between the pairs in the original embedding space.



A way to map out things in a d-dimensional space to figure out how similar they might be to one another.

{SUPERVISED}

WORD2VEC

sad lonely happy angry scared frightened appreciative upset nervous

WORD2VEC

sad

upset

angry

lonely

scared

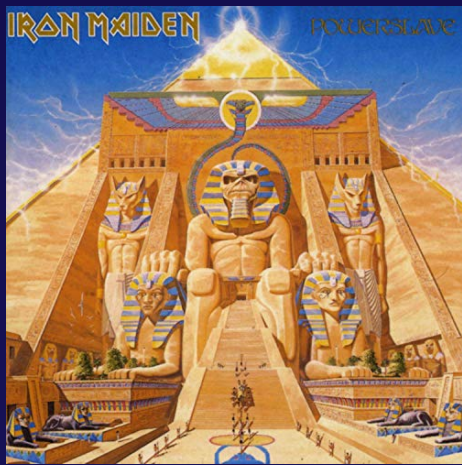
frightened

nervous

happy

appreciative

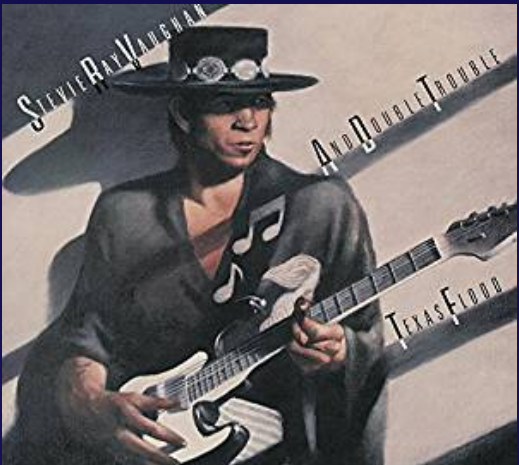
OBJECT2VEC



Rock



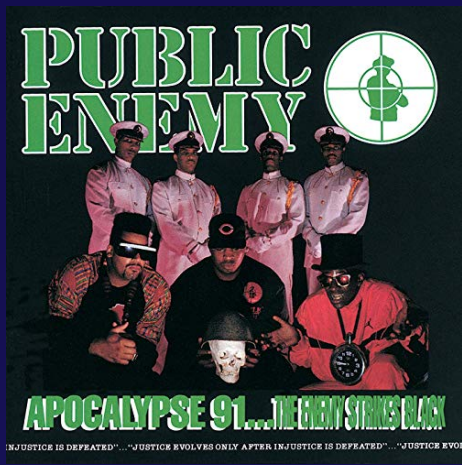
Blues



Blues



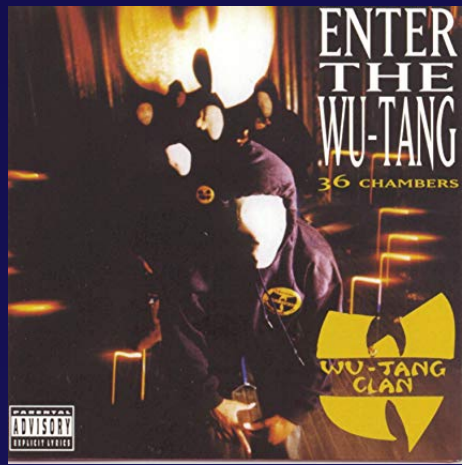
Rock













Rap



Rock

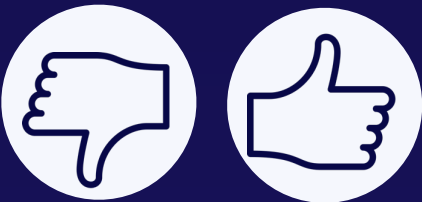


Rap

| | | | | | | |
|----------|---|--|---|---|---|---|
| Person 1 |  | |  | | |  |
| Person 2 | |  |  | | |  |
| Person 3 |  | | |  |  |  |

OBJECT2VEC

Person 4



1

Expects Pairs of Things

Looking for pairs of items and whether they are “positive” or “negative” from a relationship standpoint. Accepts categorical labels or rating/score-based labels.

1

Expects Pairs of Things

Looking for pairs of items and whether they are “positive” or “negative” from a relationship standpoint. Accepts categorical labels or rating/score-based labels.

2

Feature Engineering

Embedding can be used for downstream supervised tasks like classification or regression.

1

Expects Pairs of Things

Looking for pairs of items and whether they are “positive” or “negative” from a relationship standpoint. Accepts categorical labels or rating/score-based labels.

2

Feature Engineering

Embedding can be used for downstream supervised tasks like classification or regression.

3

Training Data Required

Officially, Object2Vec requires labeled data for training, but there are ways to generate the relationship labels from natural clustering.

- **Movie Rating Prediction**

Example: Predict the rating a person is likely to give a movie based on similarity to other's movie ratings.

- **Document Classification**

Example: Determine which genre a book is based on its similarity to known genres (history, thriller, biography, etc.)