Home › Tutorials › Machine Learning

What is Topic Modeling? An Introduction With Examples

Unlock insights from unstructured data with topic modeling. Explore core concepts, techniques like LSA & LDA, practical examples, and more.

Oct 2023 · 13 min read

Kurtis Pykes
Data Science & AI Blogger | Top 1000 Medium Writers on AI and Data Science

TOPICS

Machine Learning

The objective of analytics is to derive insights from data. Traditionally, such data was structured, meaning it's in a standardized format for efficient access. As the world shifts and becomes more digitalized, much of the data being generated is unstructured, meaning there's no predefined data model.

According to Gartner, unstructured data represents 80-90% of all new enterprise data. Furthermore, it's growing three times faster than structured data. As a result, analytics experts must employ new techniques to obtain relevant information from their datasets.

One such technique being leveraged is topic modeling from the field of text mining. In the remainder of this article, we will cover:

- What is topic modeling?
- Core concepts
- Two popular topic modeling techniques
- A hands-on example
- Topic modeling use cases
- How topic modeling differs from other techniques

What is Topic Modeling?

Topic modeling is a frequently used approach to discover hidden semantic patterns portrayed by a text corpus and automatically identify topics that exist inside it.

Namely, it's a type of statistical modeling that leverages unsupervised machine learning to analyze and identify clusters or groups of similar words within a body of text.

For example, a topic modeling algorithm may be deployed to determine whether the contents of a document imply it's an invoice, complaint, or contract.

The role of topic modeling in business

According to some sources, the average person generates in excess of 1.7MB of digital data per second. This number amounts to more than 2.5 quintillion bytes of data per day, of which 80-90% is unstructured.

Consider a scenario where a business employs a single individual to review each piece of unstructured data and segment them based on the underlying topic. It would be an impossible task.

It would take a significant amount of time to complete and be extremely tedious, plus there's much more risk involved since humans are naturally biased and more error-prone than machines.

The solution is **topic modeling**.

With topic modeling, insights from the data can be derived faster and possibly better. This technique combines the topics into a comprehensible structure, enabling businesses to rapidly understand what's happening.

For example, a business that wants to understand customers' biggest challenges may employ topic modeling to learn this information through unstructured data.

In short, topic modeling aids businesses in:

- Performing real-time analysis on unstructured textual data
- Learn from unstructured data at scale
- Build a consistent understanding of data, regardless of its format.

Core Concepts of Topic Modeling

We've established topic modeling enables data professionals to rapidly analyze and identify clusters or groups of similar words within a body of text at scale.

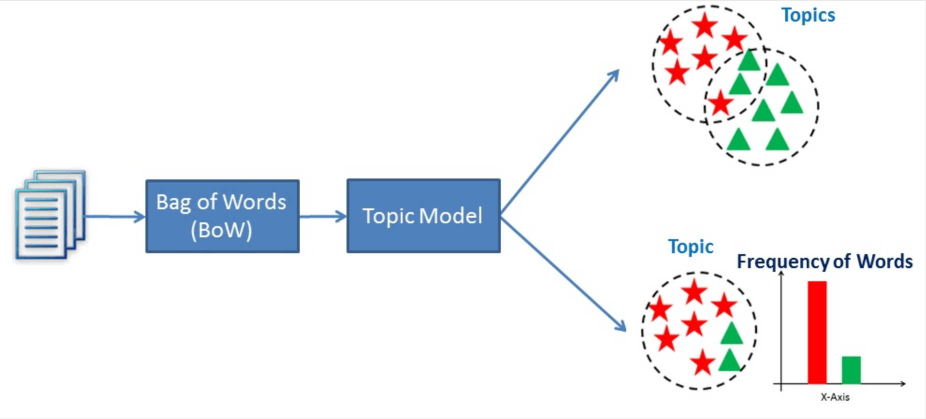
But what are topics, and how does topic modeling work?

What are topics, and how do topic models work?

Topics are the latent descriptions of a corpus (large group) of text. Intuitively, documents regarding a specific topic are more likely to produce certain words more frequently.

For example, the words "dog" and "bone" are more likely to appear in documents concerning dogs, whereas "cat" and "meow" are more likely to be found in documents regarding cats. Consequently, the topic model would scan the documents and produce clusters of similar words.

Essentially, topic models work by deducing words and grouping similar ones into topics to create topic clusters.



A visualization of how topic modeling works

Exploring Topic Modeling Techniques

Two popular topic modeling techniques are Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). Their objective to discover hidden semantic patterns portrayed by text data is the same, but how they achieve it is different.

Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is a natural language processing technique used to analyze relationships between documents and the terms they contain. The method was first introduced in a paper from 1988 titled *"Using Latent Semantic Analysis to Improve Access to Textual Information"* and is still used today to create structured data from a collection of unstructured text.

Namely, LSA assumes words with similar meanings will appear in similar documents. It does so by constructing a matrix containing the word counts per document, where each row represents a unique word, and columns represent each document, and then using a Singular Value Decomposition (SVD) to reduce the number of rows while preserving the similarity structure among columns. SVD is a mathematical method that simplifies data while keeping its important features. It's used here to maintain the relationships between words and documents.

To determine the similarity between documents, cosine similarity is used. This is a measure that calculates the cosine of the angle between two vectors, in this case, representing documents. A value close to 1 means the documents are very similar based on the words in them, whereas a value close to 0 means they're quite different.

Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) was initially proposed in 2000 in a paper titled ***"Inference of population structure using multilocus genotype data."*** The paper predominantly focused on population genetics, which is a subfield of genetics concerned with genetic differences within and among populations. Three years later, **Latent Dirichlet Allocation** was applied in machine learning.

The authors of the paper describe the technique as *"a generative model for text and other collections of discrete data."* Thus, LDA may be described as a natural language technique used to identify topics a document belongs to based on the words contained within it.

More specifically, LDA is a Bayesian network, meaning it's a generative statistical model that assumes documents are made up of words that aid in determining the topics. Thus, documents are mapped to a list of topics by assigning each word in the document to different topics. This model ignores the order of words occurring in a document and treats them as a bag of words.

LSA vs LDA

Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) are both natural language processing techniques used to create structured data from a collection of unstructured text.

However, LSA leverages Singular Value Decomposition (SVD) to reduce the dimensionality of the term-document matrix and is based on the assumption that words with similar meanings will appear in similar documents. By creating a lower-dimensional representation of the text, the model can capture the underlying relationships between words to determine how similar two documents are.

In contrast, LDA is a generative probabilistic model that leverages Bayesian inference to find the underlying topics in a corpus of texts. It assumes each document is a combination of a small number of latent topics, and each word is generated by a particular topic.

Ultimately, LSA attempts to discover the underlying relationships between words, whereas LDA seeks to discover the underlying topics in a corpus of text. Although they both are techniques used to create a vector representation of text, they make different underlying assumptions.

Practical Implementation of Topic Modeling

Let's see how these techniques work. Use this **DataCamp Workspace** to follow along with the code.

Data preparation

The first thing we need is data.

For topic modeling, the data we use is called a corpus, which is simply a collection of text.

Here's a small corpus I created using facts from the internet:

```
# Creating example documents
doc_1 = "A whopping 96.5 percent of water on Earth is in our oceans, covering 71

doc_2 = "One-third of your life is spent sleeping. Sleeping 7-9 hours each night

doc_3 = "A newborn baby is 78 percent water. Adults are 55-60 percent water. Wate
```

```
doc_4 = "While still in high school, a student went 264.4 hours without sleep, fo

doc_5 = "We experience water in all three states: solid ice, liquid water, and ga

# Create corpus
corpus = [doc_1, doc_2, doc_3, doc_4, doc_5]
```

Explain code

OpenAI

Preprocessing

The next step is cleaning the text:

```
# Code source: https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-  to

import string
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer

# remove stopwords, punctuation, and normalize the corpus
stop = set(stopwords.words('english'))
exclude = set(string.punctuation)
lemma = WordNetLemmatizer()

def clean(doc):
    stop_free = " ".join([i for i in doc.lower().split() if i not in stop])
    punc_free = "".join(ch for ch in stop_free if ch not in exclude)
    normalized = " ".join(lemma.lemmatize(word) for word in punc_free.split())
    return normalized

clean_corpus = [clean(doc).split() for doc in corpus]
```

Explain code

OpenAI

In the code above we:

1. Imported the necessary libraries and downloaded stopwords and wordnet
2. Defined the English stopwords
3. Instantiated the set of punctuation we want to exclude
4. Created the instance of the wordnet lemmatizer
5. Created a function to remove stopwords and punctuation and to lemmatize the documents.
6. Applied the clean function to each document in the corpus.

But this still doesn't mean we're ready.

Before we can use this data as input to a LDA or LSA model, it must be converted to a term-document matrix. A term-document matrix is merely a mathematical representation of a set of documents and the terms contained within them.

It's created by counting the occurrence of every term in each document and then normalizing the counts to create a matrix of values that can be used for analysis.

To do this in Python, we're going to leverage the Gensim library.

```
from gensim import corpora

# Creating document-term matrix
dictionary = corpora.Dictionary(clean_corpus)
doc_term_matrix = [dictionary.doc2bow(doc) for doc in clean_corpus]
```

Explain code OpenAI

Now, we can fit our models.

Modeling

The first model we'll use in LSA:

```
from gensim.models import LsiModel

# LSA model
lsa = LsiModel(doc_term_matrix, num_topics=3, id2word = dictionary)

# LSA model
print(lsa.print_topics(num_topics=3, num_words=3))

"""
[
(0, '0.555*"water" + 0.489*"percent" + 0.239*"planet"'),
(1, '0.361*"sleeping" + 0.215*"hour" + 0.215*"still"'),
(2, '-0.562*"water" + 0.231*"rain" + 0.231*"planet"')
]
"""
```

Explain code OpenAI

This outputs the topics (each line) with individual topic terms (terms) and their weights.

Let's try it with LDA:

```
from gensim.models import LdaModel

# LDA model
lda = LdaModel(doc_term_matrix, num_topics=3, id2word = dictionary)

# Results
print(lda.print_topics(num_topics=3, num_words=3))

"""
[
(0, '0.071*"water" + 0.025*"state" + 0.025*"three"'),
(1, '0.030*"still" + 0.028*"hour" + 0.026*"sleeping"'),
(2, '0.073*"percent" + 0.069*"water" + 0.031*"rain"')
]
"""
```

Explain code OpenAI

What is Topic Modeling Used For?

By removing manual and repetitive tasks, topic modeling can easily and inexpensively speed up processes in a simple fashion. Here are a few examples:

Tagging support tickets

Topic modeling can be used to help customer service staff analyze support queries to identify the primary issues and determine any that repeatedly occur. Based on that data, they may be able to create more informative self-serve content or help customers directly.

Enhancing customer experience

Topic modeling may be used to tag conversations such that they may be routed to the most appropriate team. For example, a conversation that includes words such as "pricing," "subscription," "renewal," etc. could be sent straight to the accounting department for support.

Topic Modeling vs Other Techniques

Topic modeling vs clustering

Topic modeling is used to discover latent topics that exist within a collection of documents. This involves identifying patterns in the words and phrases that appear in documents and grouping them into topics based on how similar they are.

Contrastingly, clustering is a technique used to group similar objects based on a measure of similarity. Such methods are employed to discover patterns and structure in data by grouping together similar data points.

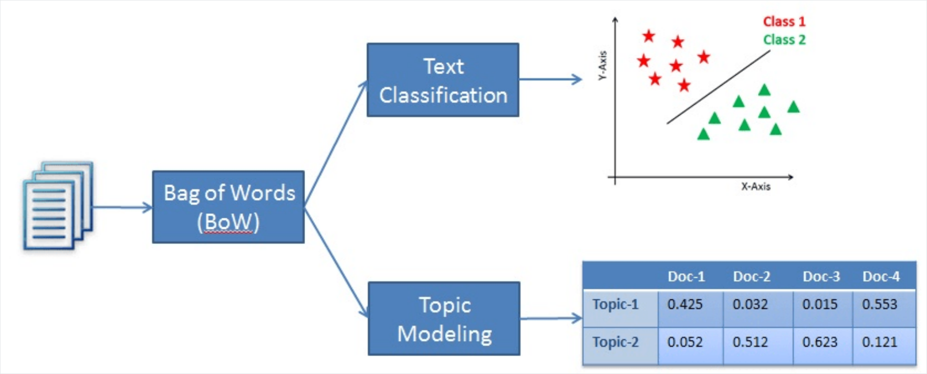
Though both approaches can uncover patterns in text data, they have different goals. Topic modeling is concerned with identifying latent topics in a collection of documents, while clustering is concerned with grouping similar data points together.

Topic modeling vs text classification

Text classification, though a natural language processing technique, falls under the category of supervised learning. Namely, text classification is employed to label predefined categories or a given piece of text. In order for the model to achieve this feat, it must first learn from a labeled data set before it can be used to make predictions on new, unseen text samples.

On the other hand, topic modeling is an unsupervised learning technique used to find the underlying topics in a collection of text documents. This means it doesn't have to learn from a labeled dataset.

Thus, the difference between the two methods is text classification is used to assign predefined labels to text, whereas topic modeling discovers the underlying topics in a collection of documents.



	Doc-1	Doc-2	Doc-3	Doc-4
Topic-1	0.425	0.032	0.015	0.553
Topic-2	0.052	0.512	0.623	0.121

An example of classification vs topic modeling

Conclusion

Topic modeling is a popular natural language processing technique used to create structured data from a collection of unstructured data. In other words, the technique enables businesses to learn the hidden semantic patterns portrayed by a text corpus and automatically identify the topics that exist inside it.

Two popular topic modeling approaches are LSA and LDA. They both seek to discover the hidden patterns in text data, but they make different assumptions to achieve their objective. Where LSA assumes words with similar meanings will appear in similar documents, LDA assumes documents are made up of words that aid in determining the topics.

In this tutorial, we've covered the core concepts of topic modeling, a practical implementation, and how topic modeling differs from other techniques, such as text classification and clustering. To continue your learning, check out some of our other resources:

- Introduction to Natural Language Processing in R

EN

AUTHOR
Kurtis Pykes
in

TOPICS

Machine Learning

Start Your Topic Modeling Journey Today!

COURSE

Introduction to Natural Language Processing in Python

4 hr 111.7K

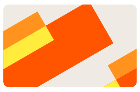
Learn fundamental natural language processing techniques using Python and how to apply them to extract insights from real-world text data.

See Details →

Start Course

See More →

Related



Navigating the World of MLOps Certifications

Adel Nehme

How to Learn Machine Learning in 2024

Adel Nehme

Becoming Remarkable with Guy Kawasaki, Author and Chief...

Richie Cotton

See More →

Grow your data skills with DataCamp for Mobile

Make progress on the go with our mobile courses and daily 5-minute coding challenges.

Download on the App Store GET IT ON Google Play

LEARN

Learn Python

Learn R

Learn AI

Learn SQL

Learn Power BI

Learn Tableau

Learn Data Engineering

Assessments

Career Tracks

Skill Tracks

Courses

Data Science Roadmap

DATA COURSES

Python Courses

R Courses

SQL Courses

Power BI Courses

Tableau Courses

Azure Courses

Spreadsheets Courses

AI Courses

Data Analysis Courses

Data Visualization Courses

Machine Learning Courses

Data Engineering Courses

Probability & Statistics Courses

WORKSPACE

Get Started

Templates

Integrations

Documentation

CERTIFICATION

Certifications

Data Scientist

Data Analyst

Data Engineer

Hire Data Professionals

RESOURCES

Resource Center

Upcoming Events

Blog

Code-Alongs

Tutorials

Open Source

RDocumentation

Course Editor

Book a Demo with DataCamp for Business

Data Portfolio

Portfolio Leaderboard

PLANS

Pricing

For Business

For Universities

Discounts, Promos & Sales

DataCamp Donates

SUPPORT

Help Center

Become an Affiliate

ABOUT

About Us

Learner Stories

Careers

Become an Instructor

Press

Leadership

Contact Us

DataCamp Español

Privacy Policy Cookie Notice Do Not Sell My Personal Information Accessibility Security Terms of Use