Topic modeling using LDA

We will first try to understand how text is seen/managed in NLP. What is the topic , what does topic modeling mean?

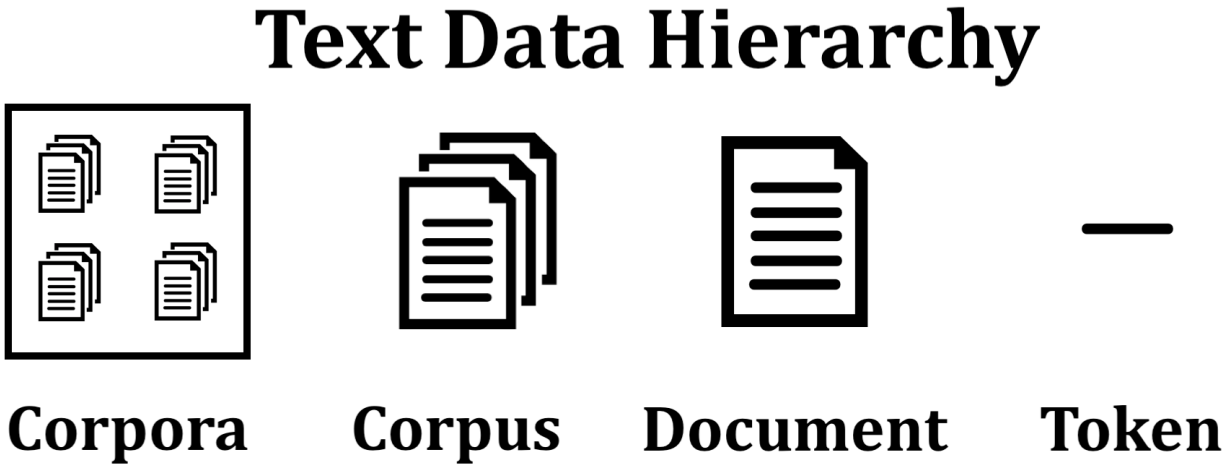
What are Topics?

Topics or themes are a group of statistically significant “tokens” or words in a “corpus”.

- A corpus is the group of all the text documents whereas a document is a collection of paragraphs.
- A paragraph is a collection of sentences and a sentence is a sequence of words (or tokens) in a grammatical construct.

So basically, a book or research paper, which collectively has pages full of sentences, can be broken down into words. In the world of Natural Language Processing (NLP), **these words are known as tokens that are a single and the smallest unit of text**.

The vocabulary is the set of unique tokenized words. And, **the first step to work through any text data is to split the text into tokens**. The process of splitting a text into smaller units or words is known as **tokenization**.



Now, the main question is **how would a machine tell us what is the topic of the document**? The only way to interpret this is to use the **language of Statistics** that machine can understand.

That's why we defined the topic as : **group of statistically significant "tokens".**

So, the next question that arises for us is to unravel **what do we mean by statistical significance in the context of the text data? The statistically significant words imply that this collection of words are similar to each other** and we see that in the following way within a text data:

- The group of words occurs together in the documents
- These words have similar **TF-IDF** term and inverse document frequencies
- This group of words occurs regularly at frequent intervals

Inverse Document Frequency: Mainly, it tests how relevant the word is. The key aim The search is to locate the appropriate records that fit the demand.

$$idf(t) = N/\ df(t) = N/N(t)$$

Here , N = number of documents in the corpus

N(t) = Number of documents containing the term t

$$idf(t) = log(N/\ df(t))$$

The words with higher scores of weight are deemed to be more significant.

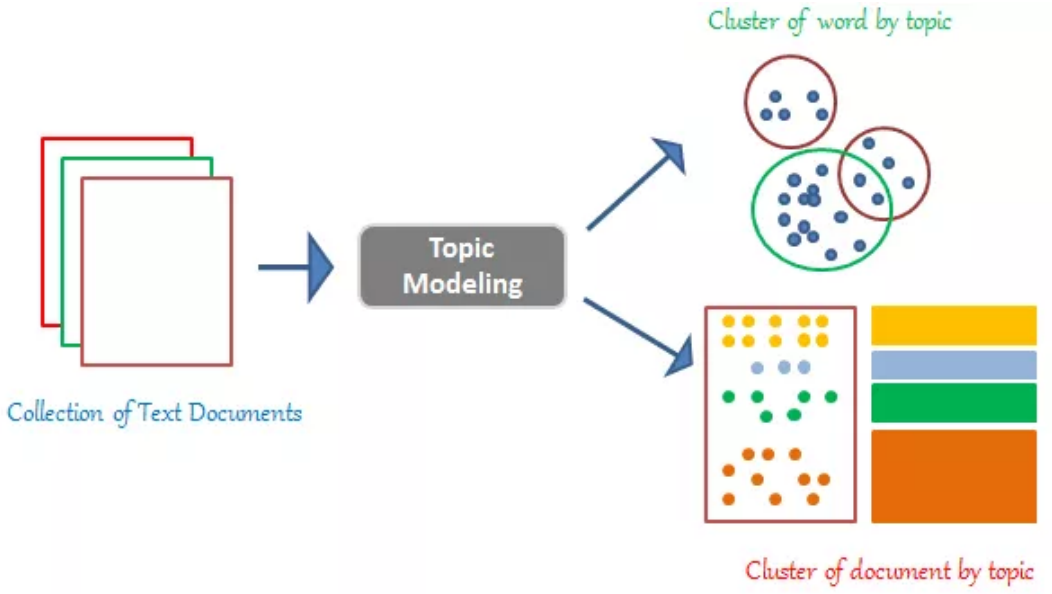
Topic 1		Topic 2		Topic 3	
Token	Weight	Token	Weight	Token	Weight
pasta	0.010	chess	0.005	neurons	0.011
food	0.017	game	0.015	DNA	0.005
sauce	0.008	football	0.009	brain waves	0.014
cheese	0.012	cricket	0.006	genes	0.012
pizza	0.007	play	0.004	neuroscience	0.013

In the above table, we have three different topics. Topic 1 on food, Topic 2 talks about games, and Topic 3 have words related to neuroscience. In each case, the words that are similar to each other come together as a topic.

What is Topic Modeling?

Topic modeling is the process of automatically finding the hidden topics in textual data. It is also referred to as the text or information mining technique that has the aim to find the recurring patterns in the words present in the corpus.

- Topic Modeling is an unsupervised learning method as we do not need to supply the labels to the topic modeling algorithm for the identification of the themes or the topics.
- Topic modeling can be seen as a clustering methodology, wherein the small groups (or clusters) that are formed based on the similarity of words are known as topics. Additionally, topic modeling returns another set of clusters which are the group of documents collated together on the similarity of the topics.



- In topic modeling we will generate certain topics and then find the weight of each topic in every document of corpus.
- if the topic is present in the document then the values (which are random as of now) assigned to it convey how much weightage does that topic has in the particular document.
- a document may be a combination of many topics. Our intention here with topic modeling is to find the main dominant topic or the theme.

What are the Uses of Topic Modeling?

Document Categorization: The goal is to categorize or classify a large set of documents into different categories based on the common underlying theme.

Document Summarization: It is a very handy tool for generating summaries of large documents.

Intent Analysis: Intent analysis means what each sentence (or tweet or post or complaint) refers to. It tells what is the text trying to explain in a particular document.

Information Retrieval , Dimensionality Reduction, Recommendation Engines , etc.

Topic Modeling Tools

Now, moving on to the techniques for executing topic modeling on a corpus. There are many methods for topic modeling such as:

- **Latent Dirichlet Allocation (LDA)**
- **Latent Semantic Allocation (LSA)**
- **Non-negative Matrix-Factorization (NNMF)**

Of the above techniques, we will dive into LDA as it is a very popular method for extracting topics from textual data.

The topic modeling technique, Latent Dirichlet Allocation (LDA) is a breed of generative probabilistic model. It generates probabilities to help extract topics from the words and collate documents using similar topics.

LDA and Its Working

The LDA model describes the pattern of the words that are repeating together, occurring frequently, and these words are similar to each other.

The stochastic process uses Bayesian inferences for explaining "the prior knowledge about the distribution of random variables". In the case of topic modeling, the process helps in estimating what are the chances of the words, which are spread over the document, will occur again? This enables the model to build data points, estimate probabilities, that's why LDA is a breed of generative probabilistic model.

LDA generates probabilities for the words using which the topics are formed and eventually the topics are classified into documents.

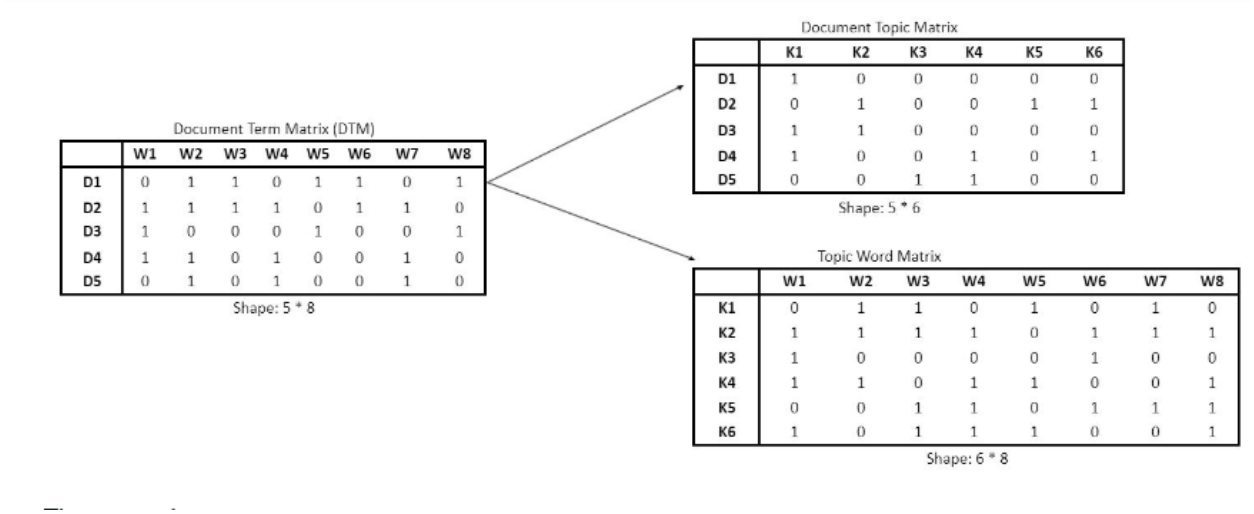
The LDA makes two key assumptions:

1. **Documents are a mixture of topics, and**
2. **Topics are a mixture of tokens (or words)**

The **Dirichlet process** is a probability distribution wherein the range of this distribution is itself a set of probability distributions. So, here **the documents are known as the probability density (or distribution) of topics and the topics are the probability density (or distribution) of words.**

Any given corpus, We know the first step with the text data is to clean, preprocess and tokenize the text to words. After preprocessing the documents,
Any corpus, which is the collection of documents, can be represented as a document-word (or document term matrix) also known as DTM. Which is whether a word is present in the document or not. If total documents are 10 in corpus and total unique words are 40. Then DTM is a (10,40) matrix.
In DTM , every row is a document and every column is the tokens or the words.

LDA converts this document-word matrix into two other matrices: Document Topic matrix and Topic Word matrix as shown below:



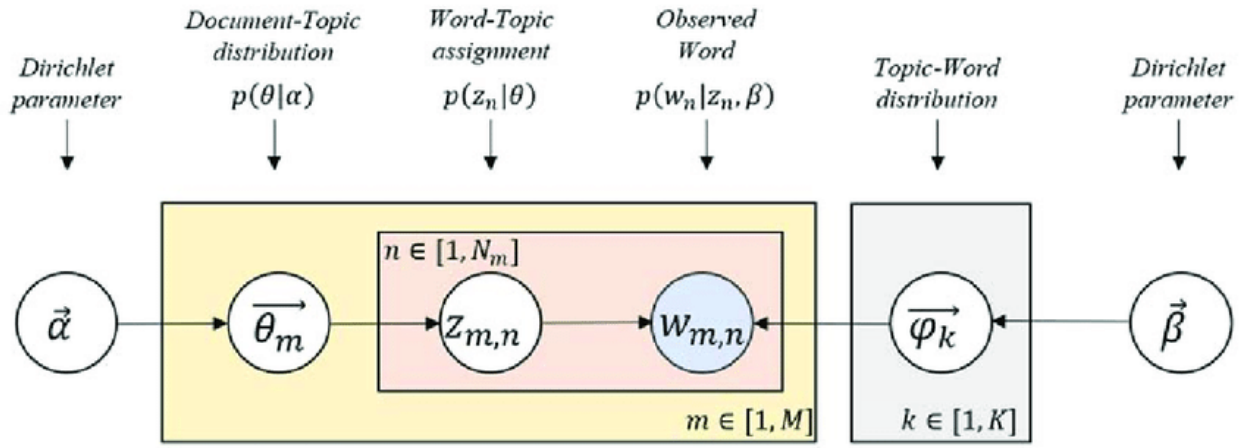
	W1	W2	W3	W4	W5	W6	W7	W8
D1	0	1	1	0	1	1	0	1
D2	1	1	1	1	0	1	1	0
D3	1	0	0	0	1	0	0	1
D4	1	1	0	1	0	0	1	0
D5	0	1	0	1	0	0	1	0

	K1	K2	K3	K4	K5	K6
D1	1	0	0	0	0	0
D2	0	1	0	0	1	1
D3	1	1	0	0	0	0
D4	1	0	0	1	0	1
D5	0	0	1	1	0	0

	W1	W2	W3	W4	W5	W6	W7	W8
K1	0	1	1	0	1	0	1	0
K2	1	1	1	1	0	1	1	1
K3	1	0	0	0	0	1	0	0
K4	1	1	0	1	1	0	0	1
K5	0	0	1	1	0	1	1	1
K6	1	0	1	1	1	0	0	1

These matrices:

- **The Document-Topic matrix already contains the possible topics (represented by K above) that the documents can contain.**
- **The Topic-Word matrix has the words (or terms) that those topics can contain.**



- The yellow box refers to all the documents in the corpus (represented by M)
- Next, the peach color box is the number of words in a document, given by NInside this peach box, there can be many words. One of those words is w, which is in the blue color circle.

According to LDA, every word is associated (or related) with a latent (or hidden) topic, which here is stated by Z. Now, this assignment of Z to a topic word in these documents gives a topic word distribution presenti n the corpus that is represented by theta.

The LDA model has two parameters that control the distributions:

1. Alpha (α) controls per-document topic distribution, Higher the Alpha doc will have more no. of topic in it.
2. Beta controls per topic word distribution. Higher the beta topic will have more no. of words in it.

- M: is the total documents in the corpus
- N: is the number of words in the document
- $w_{n,m}$: is the Specific Word
- $Z_{n,m}$: is the latent topic assigned to a nth word in document m
- theta (Θm): is the topic distribution for document m
- LDA model s parameters: Alpha (α) and Beta ( )

So, in other bag of word modeling techniques we will use frequency of words only but here we are using probabilistic approach to classify the words.

In LDA our **end goal** is to find the most optimal representation of the Document-Topic matrix and the Topic-Word matrix to **find the most optimized Document-Topic distribution and Topic-Word distribution.**

The Work Flow for executing LDA in Python:

01. we will compile all the documents into one list to have the corpus.

02. We will perform the following text preprocessing steps (can use either spacy or NLTK libraries for preprocessing):
    - Convert the text into lowercase
    - Split text into words
    - Remove the stop loss words
    - Remove the Punctuation, any symbols, and special characters
    - Normalize the word (I'll be using Lemmatization for normalization)

03. The next step is to convert the cleaned text into a numerical representation. Use either the Count vectorizer or TF-IDF vectorizer to transform the Document Term Matrix (DTM) into numerical arrays.

04. We pass the vectorized corpus to the LDA model for the package sklearn.