**What is Topic Modelling**

* Topic Models, in a nutshell, are a type of statistical language models used for uncovering hidden structure in a collection of texts.
* Dimensionality Reduction, where rather than representing a text T in its feature space as {Word\_i: count(Word\_i, T) for Word\_i in Vocabulary}, you can represent it in a topic space as {Topic\_i: Weight(Topic\_i, T) for Topic\_i in Topics}
* Unsupervised Learning, where it can be compared to clustering, as in the case of clustering, the number of topics, like the number of clusters, is an output parameter. By doing topic modeling, we build clusters of words rather than clusters of texts. A text is thus a mixture of all the topics, each having a specific weight
* Tagging, abstract “topics” that occur in a collection of documents that best represents the information in them.
* There are several existing algorithms you can use to perform the topic modeling. The most common of it are, Latent Semantic Analysis (LSA/LSI), Probabilistic Latent Semantic Analysis (pLSA), and Latent Dirichlet Allocation (LDA)
* Topic modeling is an unsupervised technique that intends to analyze large volumes of text data by clustering the documents into groups. In the case of topic modeling, the text data do not have any labels attached to it. Rather, topic modeling tries to group the documents into clusters based on similar characteristics.
* A typical example of topic modeling is clustering a large number of newspaper articles that belong to the same category. In other words, cluster documents that have the same topic. It is important to mention here that it is extremely difficult to evaluate the performance of topic modeling since there are no right answers. It depends upon the user to find similar characteristics between the documents of one cluster and assign it an appropriate label or topic.
* Two approaches are mainly used for topic modeling: [Latent Dirichlet Allocation](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation) and [Non-Negative Matrix factorization](https://en.wikipedia.org/wiki/Non-negative_matrix_factorization).

**Latent Dirichlet Allocation (LDA)**

The LDA is based upon two general assumptions:

* Documents that have similar words usually have the same topic
* Documents that have groups of words frequently occurring together usually have the same topic.

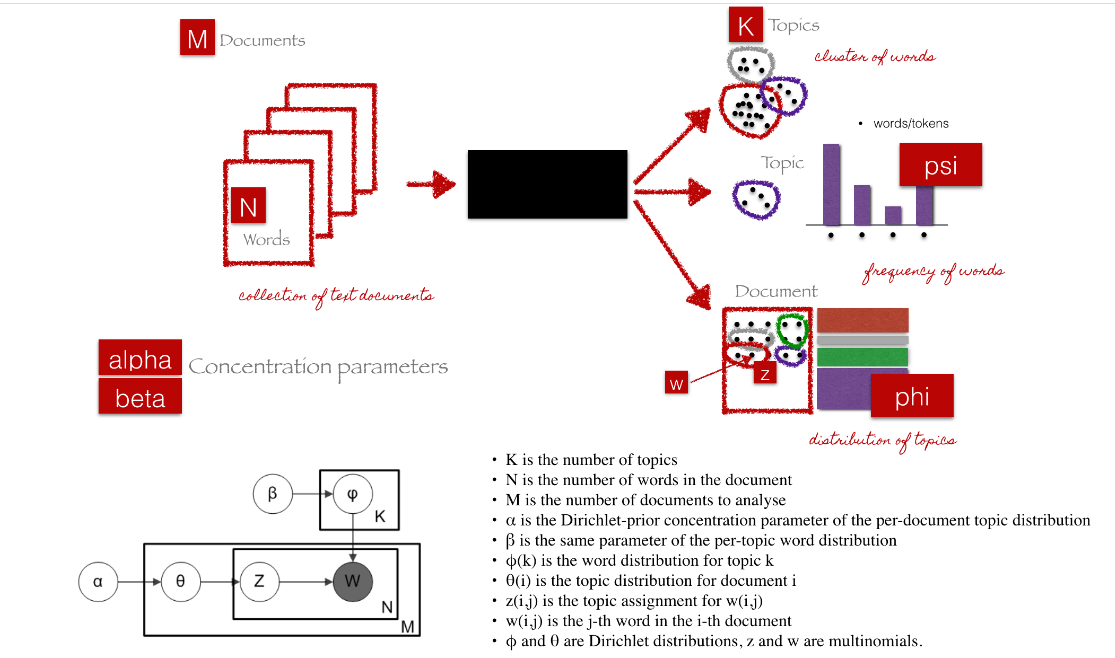
These assumptions make sense because the documents that have the same topic, for instance, Business topics will have words like the "economy", "profit", "the stock market", "loss", etc. The second assumption states that if these words frequently occur together in multiple documents, those documents may belong to the same category.

Mathematically, the above two assumptions can be represented as:

* Documents are probability distributions over latent topics
* Topics are probability distributions over words

# Theoretical Overview

LDA is a generative probabilistic model that assumes each topic is a mixture over an underlying set of words, and each document is a mixture of over a set of topic probabilities.



We can describe the generative process of LDA as, given the *M*number of documents, *N*number of words, and prior *K* number of topics, the model trains to output:

*psi*, the distribution of words for each topic *K*

*phi*, the distribution of topics for each document *i*

# Parameters of LDA

**Alpha parameter** is Dirichlet prior concentration parameter that represents document-topic density — with a higher alpha, documents are assumed to be made up of more topics and result in more specific topic distribution per document.

**Beta parameter** is the same prior concentration parameter that represents topic-word density — with high beta, topics are assumed to made of up most of the words and result in a more specific word distribution per topic.

# LDA Implementation

1. Loading data
2. Data cleaning
3. Exploratory analysis
4. Preparing data for LDA analysis
5. LDA model training
6. Analyzing LDA model results

#### Step 6: Analyzing our LDA model

Now that we have a trained model let’s visualize the topics for interpretability. To do so, we’ll use a popular visualization package, pyLDAvis which is designed to help interactively with:

1. Better understanding and interpreting individual topics, and
2. Better understanding the relationships between the topics.

For (1), you can manually select each topic to view its top most frequent and/or “relevant” terms, using different values of the λ parameter. This can help when you’re trying to assign a human interpretable name or “meaning” to each topic.

For (2), exploring the Intertopic Distance Plot can help you learn about how topics relate to each other, including potential higher-level structure between groups of topics.

**Non-Negative Matrix Factorization (NMF)**

Non-negative matrix factorization is also a supervised learning technique which performs clustering as well as dimensionality reduction. It can be used in combination with TF-IDF scheme to perform topic modeling. In this section, we will see how Python can be used to perform non-negative matrix factorization for topic modeling.