DSA4213

Lecture 4 - 20240205

Dr Vishal Sharma,
Data Science Lead, H2O.AI

Are We Recording

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

NLP Applications

- Classification
- Regression
- Summarization
- Question Answering
- Similarity

NLP Applications

- Classification
- Regression
- Summarization
- Question Answering
- Similarity

Supervised

NLP Applications

- Classification
- Regression
- Summarization
- Question Answering
- Similarity

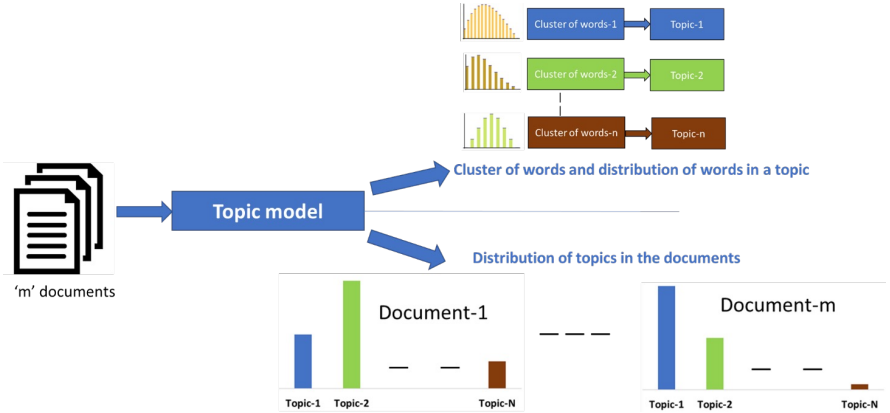
- Topic Modeling

Topic Modelling

What is Topic Modeling

- Each document consists of more than one topics, and
- Each topic consists of a collection of tokens.
- Topic = Similar records (sentences/paragraphs/documents) clustered together based on the tokens (words)

- Topic Modeling = Unsupervised method of finding topics in collections of documents



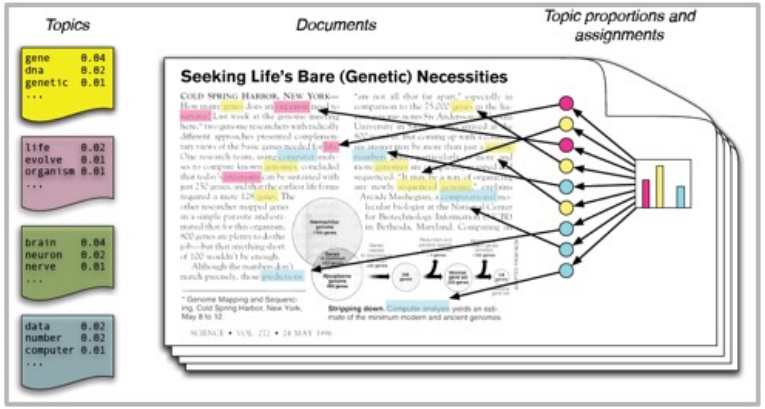
- Topic Modelling is the process of dividing a corpus of documents into the following two things:
  - A list containing all the topics that are covered by the documents in the corpus.
  - Grouped several sets of documents from the corpus-based on the topics they cover.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Source: Google Images

Why Topic Modeling

Large amounts of data are collected every day. As more information becomes available, it becomes a tedious task to find what we are looking for. So, we require some sort of tools and techniques to organize, search and understand vast quantities of information.



Topic modelling helps us to organize, understand and summarize large collections of textual information.

- To extract hidden topical patterns that are present across the collection of documents.
- Annotation of all the documents according to these topics.
- With the help of annotations, we can organize, search and summarize texts.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Source: Google Images

Key Objectives

1. What are the most important topics? → **Topic term distribution**.

2. What are the topics which are assigned to every document? → **Document to topic distribution**.

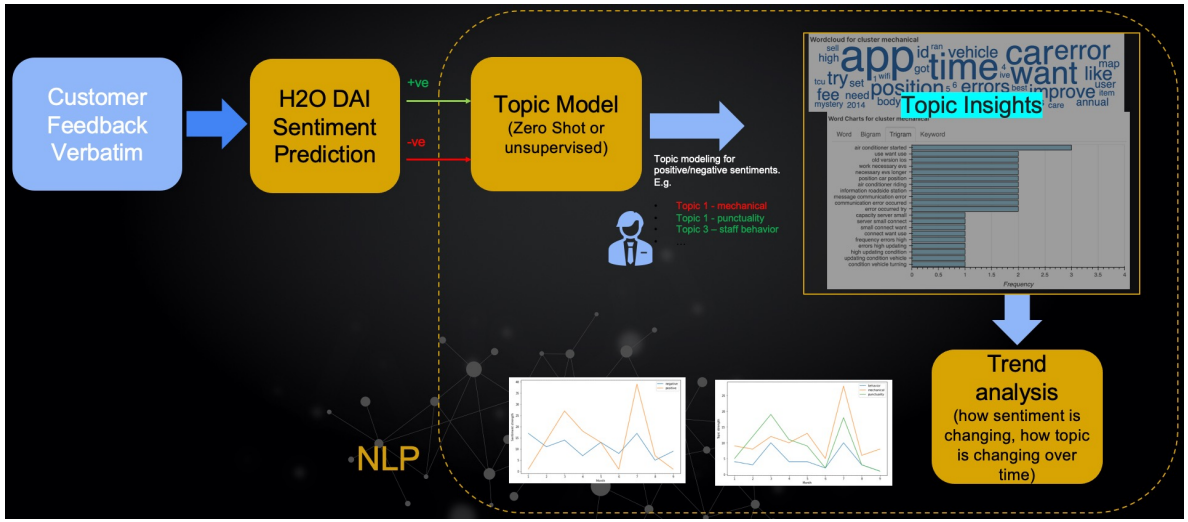
Topic Modelling tries to find the latent structure in a text corpus that:

- Resembles “topics” (also “concepts”)
- Best summarizes the collection
- Is based on Statistical Patterns
- Are obscured by synonyms, homonyms, stopwords,…
- May overlap

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Source: Google Images

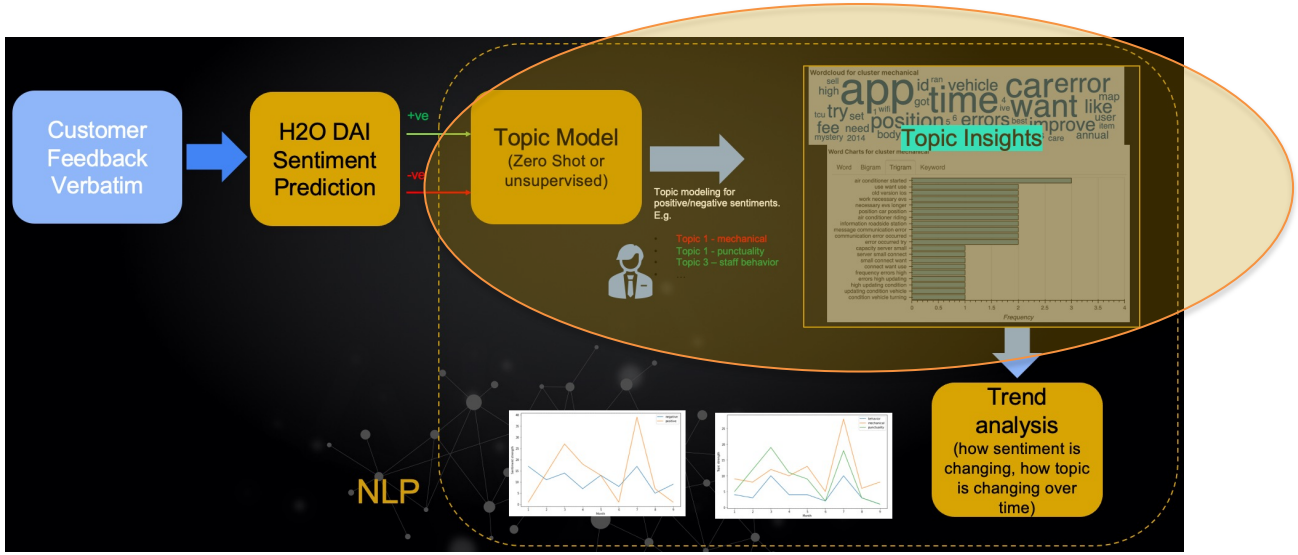
Topic Modeling in VOC



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Source: Google Images

Topic Modeling in VOC



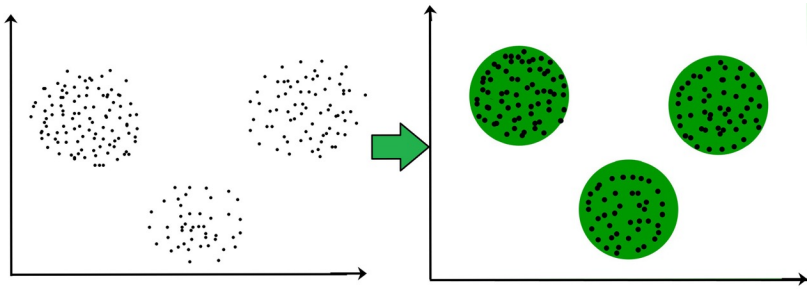
For Example, Imagine you are a manager of a software company and you want to know what customers are saying about particular features of your product.
Instead of spending our valuable time going through heaps of feedback, in an attempt to find which texts are talking about your topics of interest, you could analyze them with the help of topic modelling algorithms.

Therefore, by detecting patterns such as word frequency and distance between words, a topic model clusters feedback that is similar.
Now you can quickly deduce what each set of texts are talking about. Remember, this technique is 'unsupervised' in nature which means that no training is required.

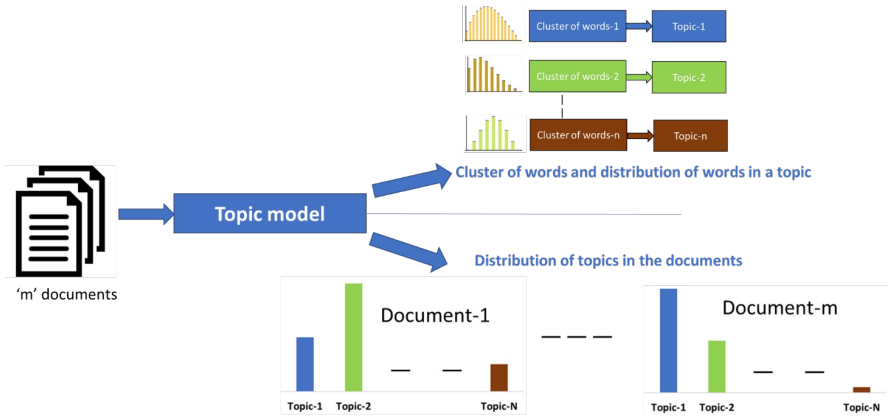
Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Source: Google Images

Similarities with Clustering



Clustering



Topic Modeling

Topic modelling is similar to clustering but with a slightly different "mindset":

- In clustering, the focus is on the data points/documents.
- In topic modelling, the focus is on the topics/cluster themselves.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Source: Google Images

Different Methods of Topic Modeling

Then with a suitable embedding (DNN or linear projection) of the skip-gram features, we find that word meaning has an algebraic structure:

- LDA – latent Dirichlet Allocation
  - Bayesian
  - Matrix factorization
- NMF – Non-negative Matrix Factorization
- LSA – Latent Semantic Allocation
- Doc2Vec
- BERTopic
- …

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Different Methods of Topic Modeling

Then with a suitable embedding (DNN or linear projection) of the skip-gram features, we find that word meaning has an algebraic structure:

- LDA – latent Dirichlet Allocation
  - Bayesian
  - Matrix factorization
- NMF – Non-negative Matrix Factorization
- LSA – Latent Semantic Allocation
- Doc2Vec
- BERTopic
- …

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Different Methods of Topic Modeling

Then with a suitable embedding (DNN or linear projection) of the skip-gram features, we find that word meaning has an algebraic structure:

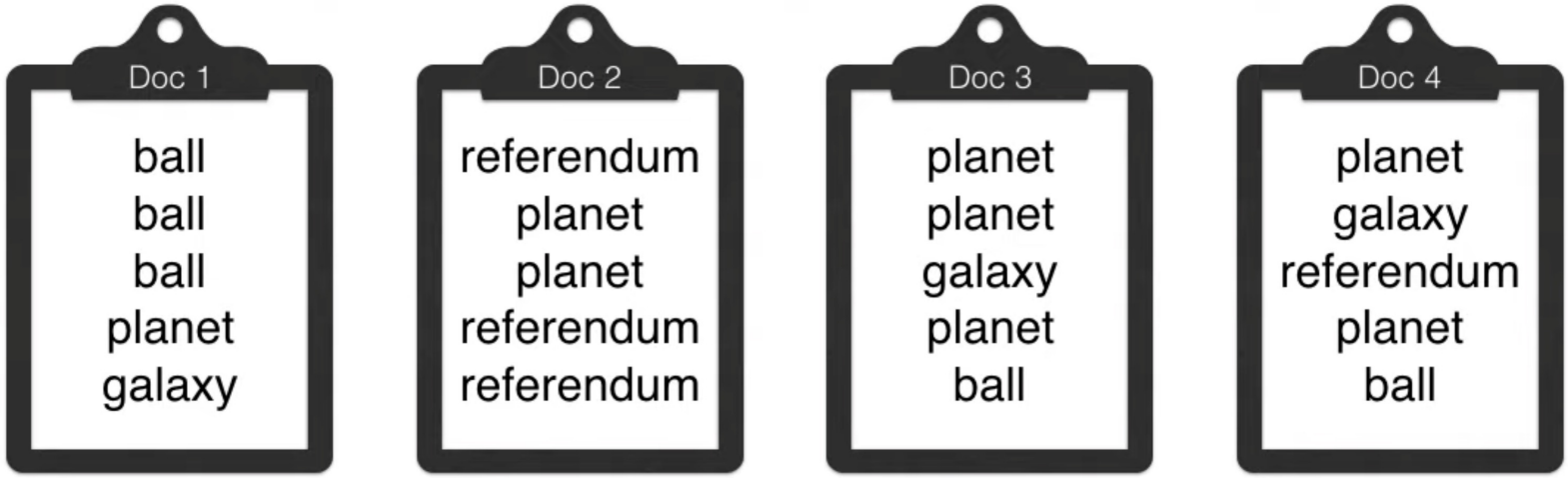
- LDA – latent Dirichlet Allocation
  - Bayesian
  - Matrix factorization
- NMF – Non-negative Matrix Factorization
- LSA – Latent Semantic Allocation
- Doc2Vec
- BERTopic
- …

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – Latent Dirichlet Allocation

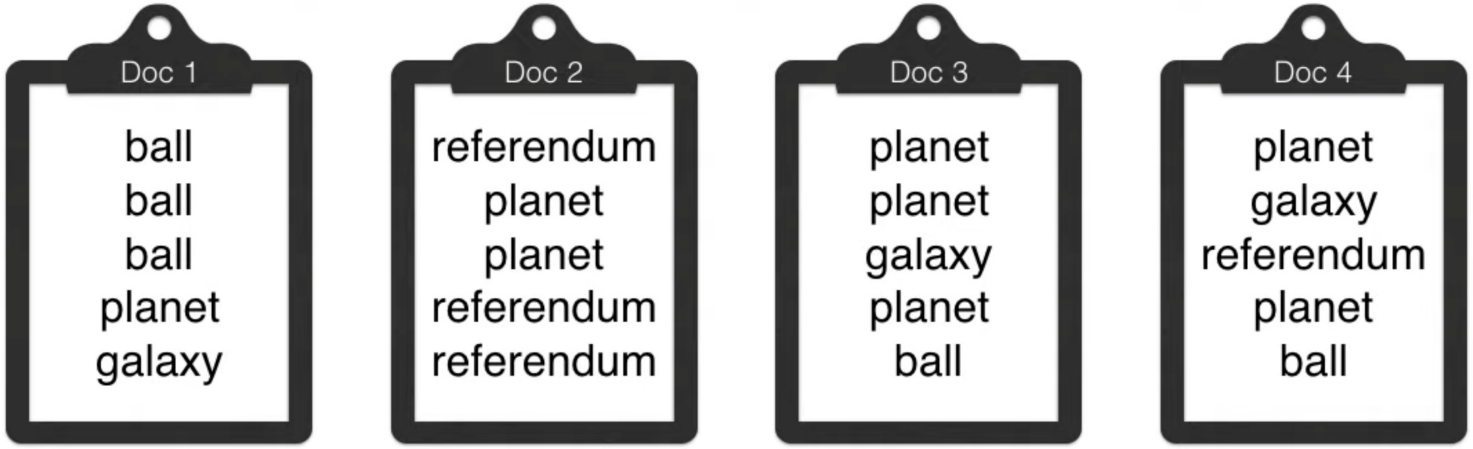
Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – Latent Dirichlet Allocation



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – Latent Dirichlet Allocation



Topic 1 Topic 2 Topic 3

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – Latent Dirichlet Allocation

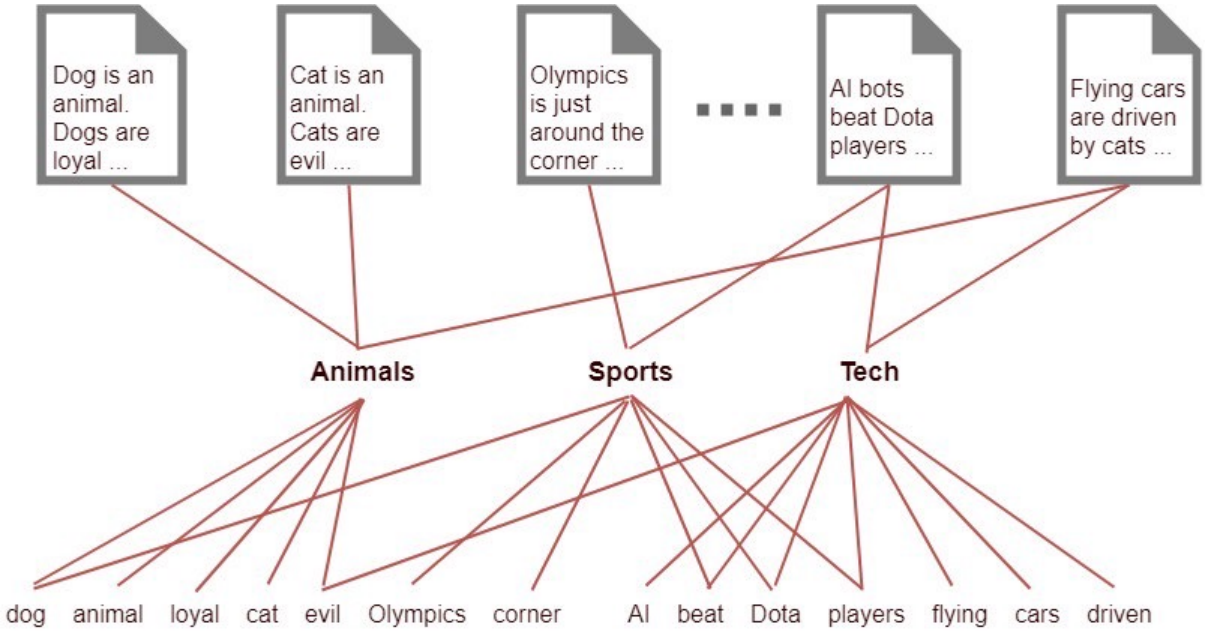
Topic 1 Topic 2 Topic 3

- **Word belongs to Topics**
- **Documents consist of words**

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

- **Latent**: This refers to everything that we don't know a priori and are hidden in the data. Here, the themes or topics that document consists of are unknown, but they are believed to be present as the text is generated based on those topics.



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

- **Latent**: This refers to everything that we don't know a priori and are hidden in the data. Here, the themes or topics that document consists of are unknown, but they are believed to be present as the text is generated based on those topics.

- **Dirichlet**: 'distribution of distributions'.
  - Let's suppose there is a machine that produces dice and we can control whether the machine will always produce a dice with equal weight to all sides, or will there be any bias for some sides.
  - So, the machine producing dice is a distribution as it is producing dice of different types. Also, we know that the dice itself is a distribution as we get multiple values when we roll a dice. This is what it means to be a distribution of distributions and this is what Dirichlet is
  - In the context of topic modeling, the Dirichlet is the distribution of topics in documents and distribution of words in the topic

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

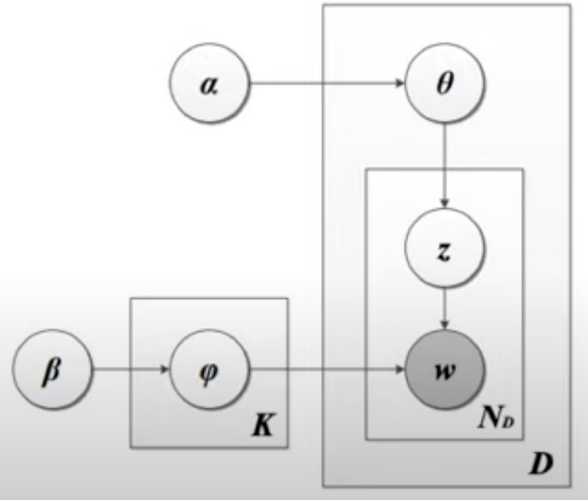
- **Latent**: This refers to everything that we don't know a priori and are hidden in the data. Here, the themes or topics that document consists of are unknown, but they are believed to be present as the text is generated based on those topics.

- **Dirichlet**: 'distribution of distributions'.
  - Let's suppose there is a machine that produces dice and we can control whether the machine will always produce a dice with equal weight to all sides, or will there be any bias for some sides.
  - So, the machine producing dice is a distribution as it is producing dice of different types. Also, we know that the dice itself is a distribution as we get multiple values when we roll a dice. This is what it means to be a distribution of distributions and this is what Dirichlet is
  - In the context of topic modeling, the Dirichlet is the distribution of topics in documents and distribution of words in the topic

- **Allocation**: of words of the document to topics .. and .. topics to documents and

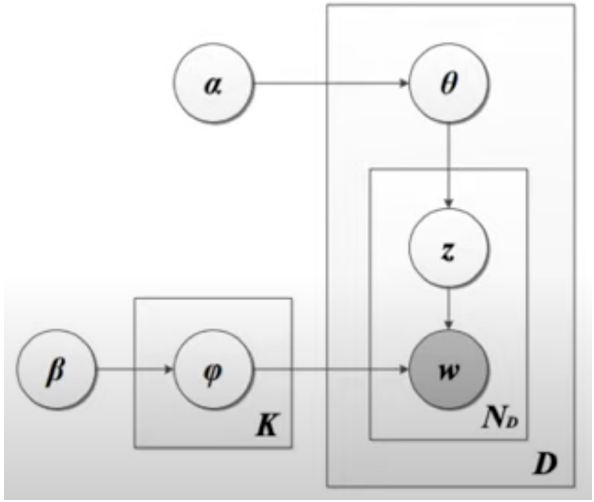
Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

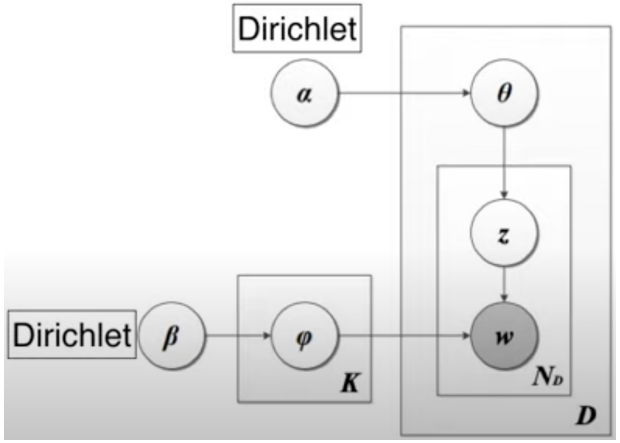
LDA



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

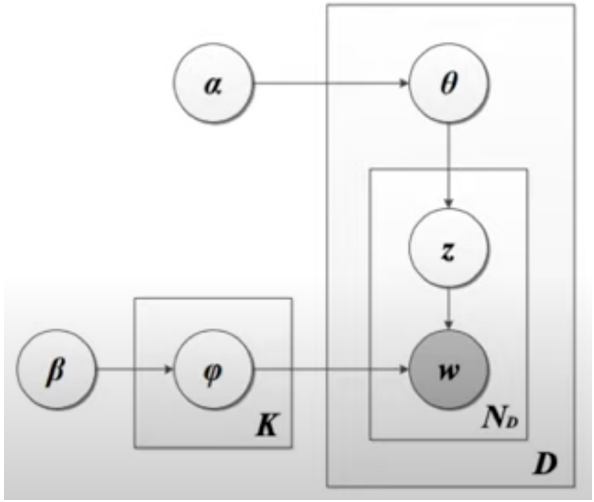
LDA

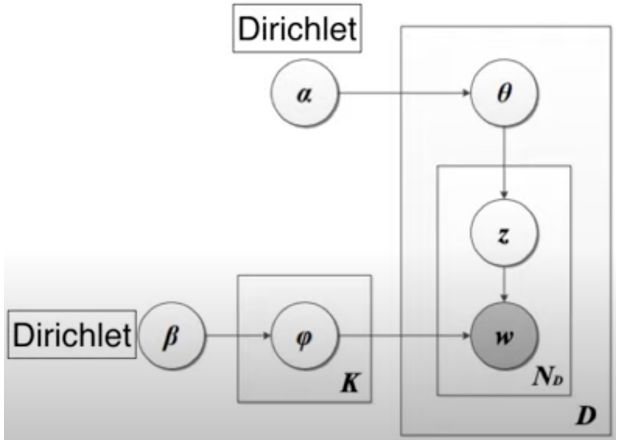


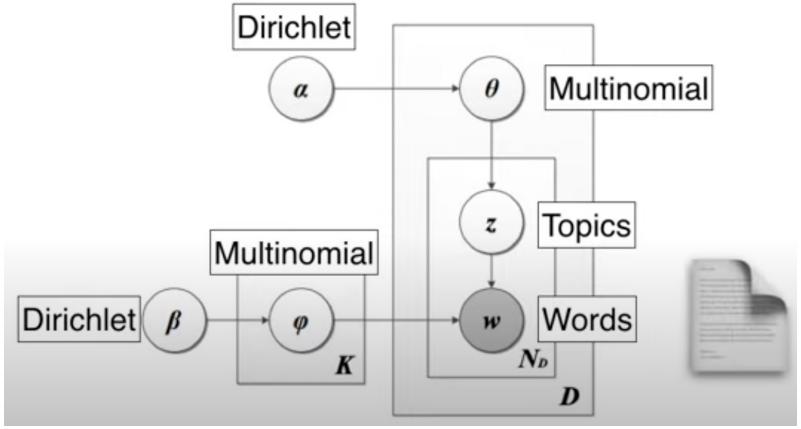


Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA







Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

- **LDA** → each word in each document comes from a topic and the topic is selected from a per-document distribution over topics. So we have two matrices:

  - *Θtd = P(t|d)* which is the probability distribution of topics in documents
  - *Φwt = P(w|t)* which is the probability distribution of words in topics

we can say that the probability of a word given document i.e. P(w|d) is equal to:

$$\sum_{t \epsilon T} p(w|t,d)\, p(t|d)$$

T = total number of topics.
W = number of words in our corpus' vocabulary

assuming conditional independence

$$P(w|t,d) = P(w|t)$$

Hence,

$$P(w|d) = \sum_{t=1}^{T} p(w|t)\, p(t|d)$$

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – Bayesian Approach

step by step procedure of the probabilistic approach for LDA is shown below:

- **Step-1** Go through each of the documents in a corpus and randomly assign each word in the document to one of K topics (K is chosen beforehand or given by the user).

- **Step-2** With the help of random assignment, we got the topic representations for all the documents and word distributions of all the topics, but these are not very good ones.

  So, to improve upon them

  For each document d, we go through each word w and compute the following:
  - **p(topic t | document d):** represents the proportion of words present in document d that are assigned to topic t of the corpus.
  - **p(word w | topic t):** represents the proportion of assignments to topic t, over all documents d, that comes from word w.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

we are trying to find conditional probability distribution of a single word's topic assignment conditioned on the rest of the topic assignments.

probability equation for a single word *w* in document *d* that belongs to topic *k*:

$$p(z_{d,n} = k | \vec{z}_{-d,n}, \vec{w}, \alpha, \lambda) = \frac{n_{d,k} + \alpha_k}{\sum_i^K n_{d,i} + \alpha_i} \frac{v_{k,w_{d,n}} + \lambda_{w_{d,n}}}{\sum_i v_{k,i} + \lambda_i}$$

where:
n(d,k): Number of times document d use topic k
v(k,w): Number of times topic k uses the given word
αk: Dirichlet parameter for document to topic distribution
λw: Dirichlet parameter for topic to word distribution

how much each topic is present in a document

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

we are trying to find conditional probability distribution of a single word's topic assignment conditioned on the rest of the topic assignments.

probability equation for a single word *w* in document *d* that belongs to topic *k*:

$$p(z_{d,n} = k | \vec{z}_{-d,n}, \vec{w}, \alpha, \lambda) = \frac{n_{d,k} + \alpha_k}{\sum_i^K n_{d,i} + \alpha_i} \frac{v_{k,w_{d,n}} + \lambda_{w_{d,n}}}{\sum_i v_{k,i} + \lambda_i}$$

where:
n(d,k): Number of times document d use topic k
v(k,w): Number of times topic k uses the given word
αk: Dirichlet parameter for document to topic distribution
λw: Dirichlet parameter for topic to word distribution

how much each topic is present in a document

how much each topic likes a word.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

we are trying to find conditional probability distribution of a single word's topic assignment conditioned on the rest of the topic assignments.

probability equation for a single word *w* in document *d* that belongs to topic *k*:

$$p(z_{d,n} = k | \vec{z}_{-d,n}, \vec{w}, \alpha, \lambda) = \frac{n_{d,k} + \alpha_k}{\sum_i^K n_{d,i} + \alpha_i} \frac{v_{k,w_{d,n}} + \lambda_{w_{d,n}}}{\sum_i v_{k,i} + \lambda_i}$$

where:
n(d,k): Number of times document d use topic k
v(k,w): Number of times topic k uses the given word
αk: Dirichlet parameter for document to topic distribution
λw: Dirichlet parameter for topic to word distribution

how much each topic is present in a document

how much each topic likes a word.

OUTPUT = For each word, we will get a vector of probabilities that will explain how likely this word belongs to each of the topics.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – Bayesian Approach

step by step procedure of the probabilistic approach for LDA is shown below:

- **Step-3** Reassign word w a new topic t', where we choose topic t' with probability p(topic t' | document d)* p(word w | topic t')
  This generative model predicts the probability that topic t' generate word w.

- **Step-4** Repeating step-3 till we reach a steady-state and at that state the topic assignments are good. And finally, we use these assignments to determine the topic mixtures of each document.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

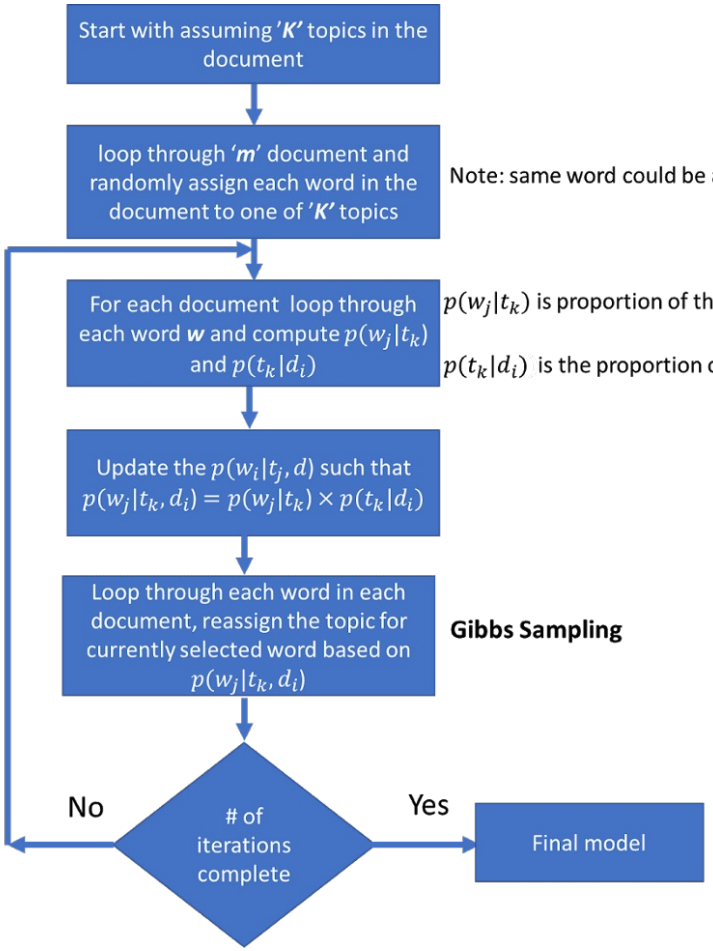
To identify the correct weights, we will use an algorithm called Gibbs sampling.

Gibbs sampling = an algorithm for successively sampling conditional
distributions of variables
distribution over states converges to the true distribution in the long run

- we start with Θ and Φ matrices.
- we will slowly change these matrices and get to an answer that maximizes the likelihood of the data that we have.
- will do this on word by word basis by changing the topic assignment of one word.
- We will assume that we don't know the topic assignment of the given word but we know the assignment of all other words in the text and we will try to infer what topic will be assigned to this word.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – Bayesian Approach



Note: same word could be assigned different topics at different instances initially

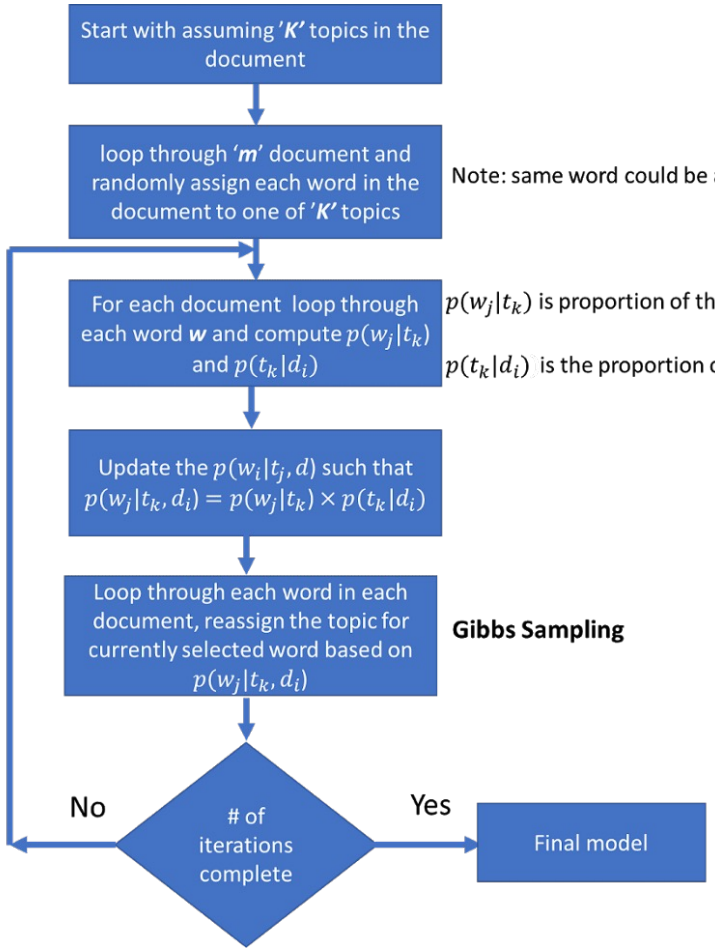
$p(w_j|t_k)$ is proportion of the all documents assigned to a topic t_K for a given word w_j

$p(t_k|d_i)$ is the proportion of words in document d_i that are assigned to topic t_k

Gibbs Sampling

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – Bayesian Approach



Note: same word could be assigned different topics at different instances initially

$p(w_j|t_k)$ is proportion of the all documents assigned to a topic t_K for a given word w_j

$p(t_k|d_i)$ is the proportion of words in document d_i that are assigned to topic t_k

Gibbs Sampling

Hyperparameters in LDA

There are three hyperparameters in LDA

1.α → document density factor

2.β → topic word density factor

3.K → number of topics selected

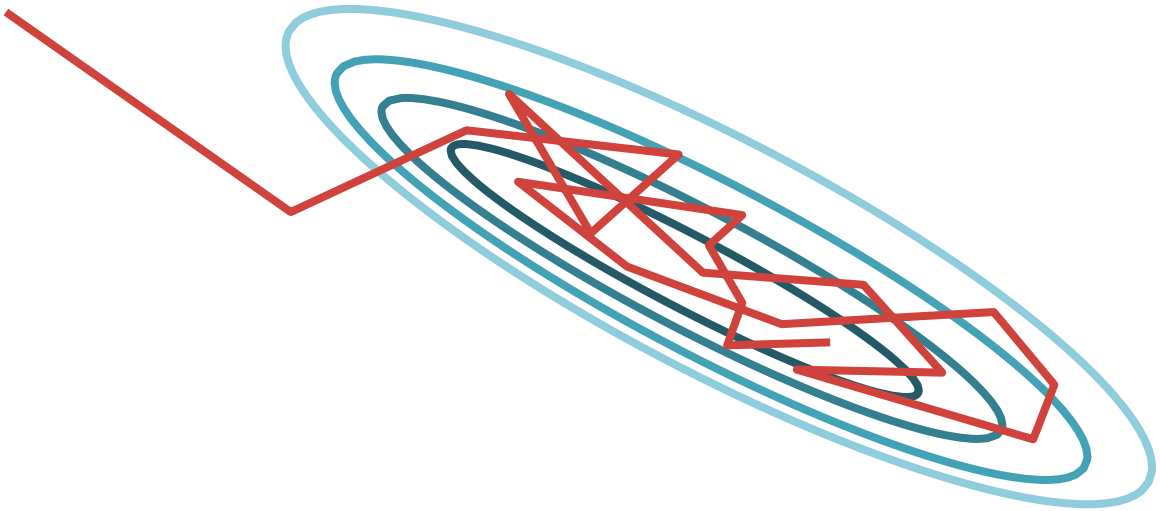
The **α** hyperparameter controls the number of topic *expected* in the document.

The **β** hyperparameter controls the distribution of words per topic in the document, and **K** defines how many topics we need to extract.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Gibbs Sampling

- **Goal:** Draw approximate, correlated samples from a target distribution p(x)
- **MCMC:** Performs a biased random walk to explore the distribution



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Gibbs Sampling Psuedocode

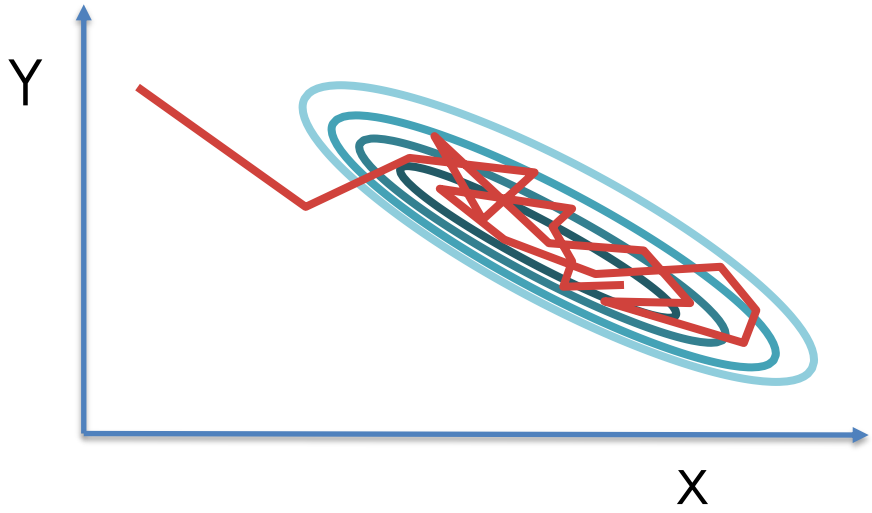
initialize Y^0, X^0

for j = 1, 2, 3,… do

sample $X^j \sim p(X|Y^{j-1})$

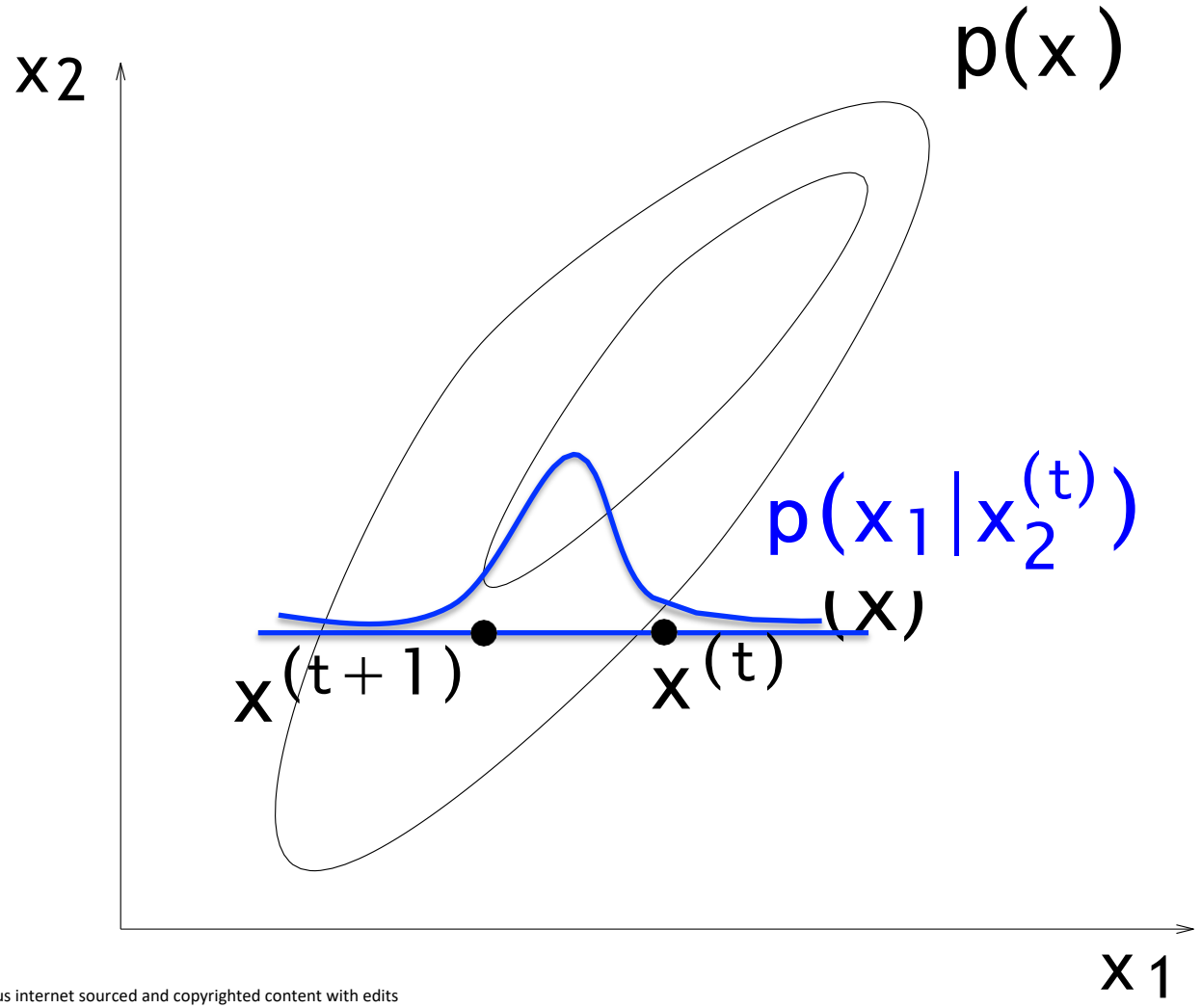
sample $Y^j \sim p(Y|X^j)$

end for



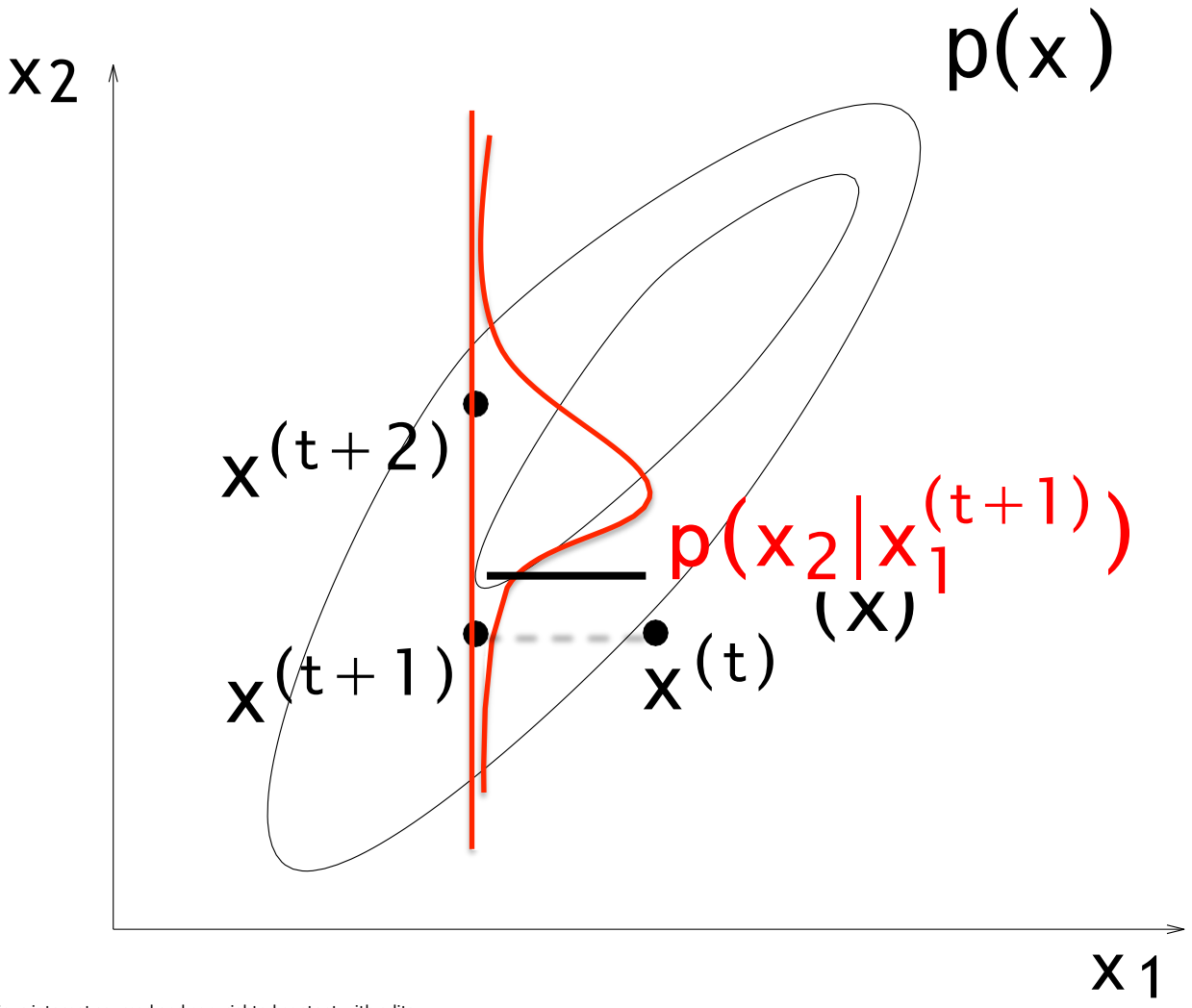
Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Gibbs Sampling



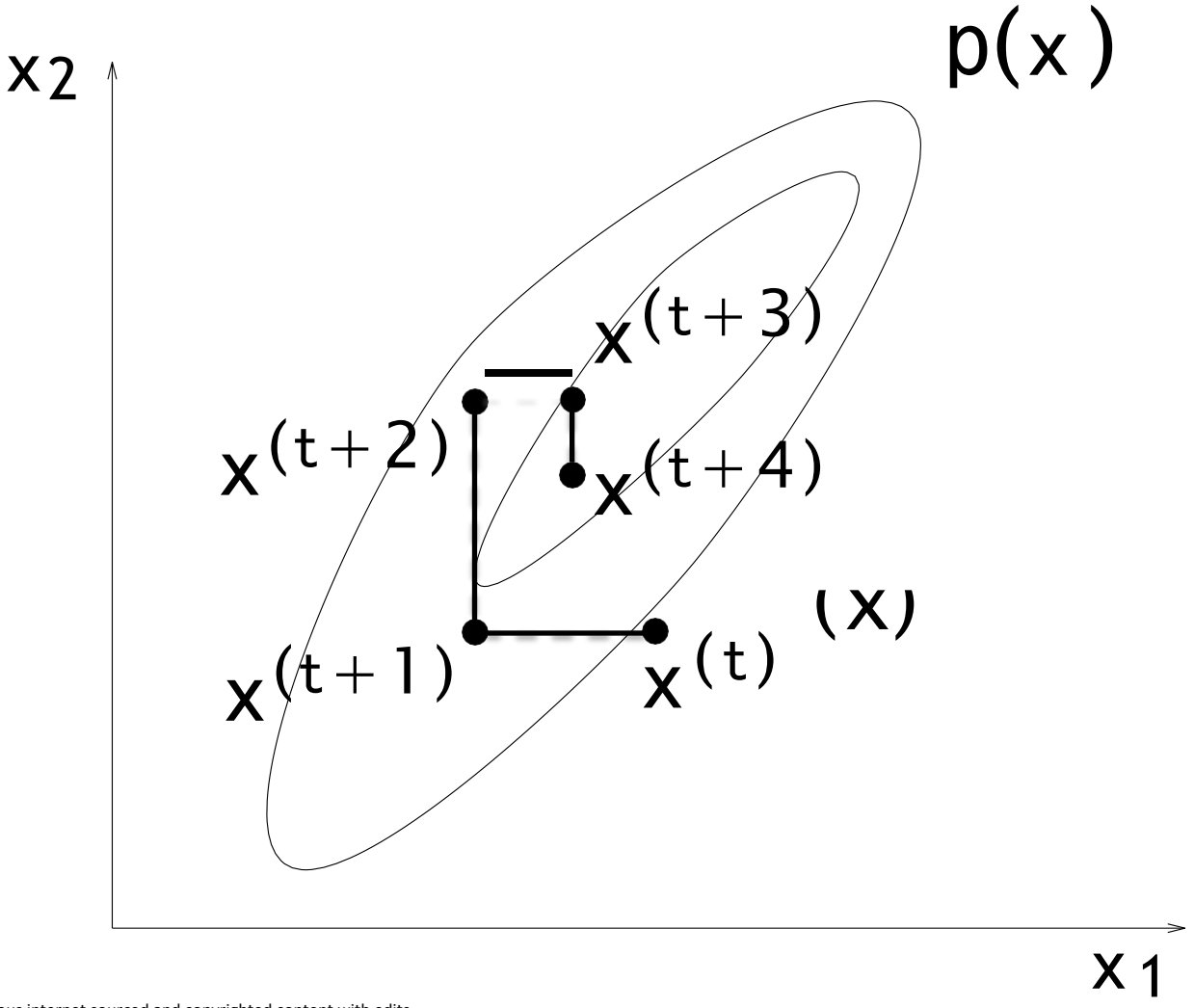
Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Gibbs Sampling



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Gibbs Sampling

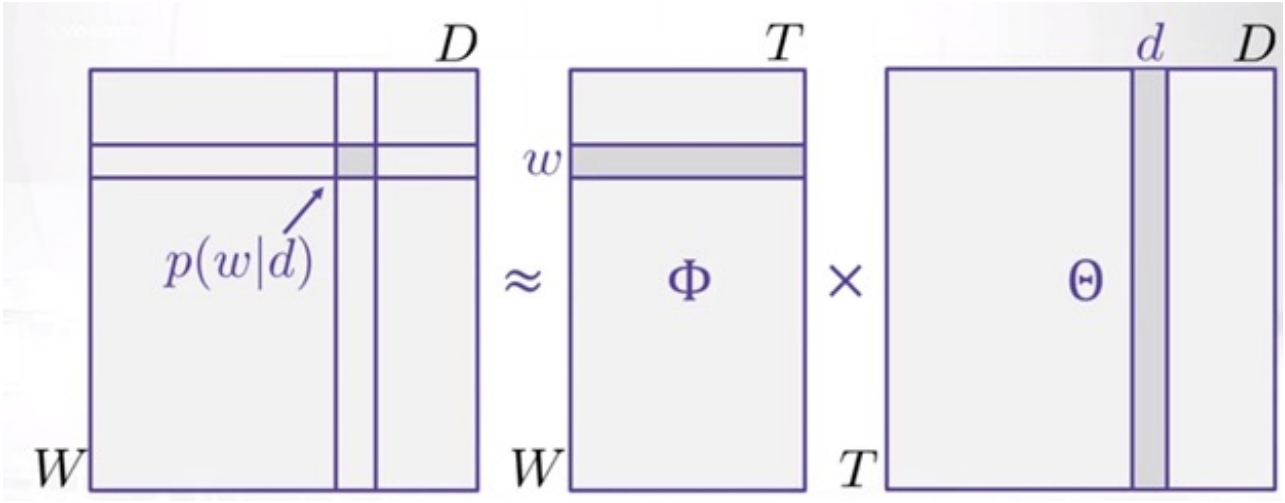


Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation – matrix view

P(w|d) = $\sum_{t=1}^{T} p(w|t)\, p(t|d)$

matrix form



- LDA similar to that of matrix factorization or SVD, where we decompose the probability distribution matrix of word in document in two matrices consisting of distribution of topic in a document and distribution of words in a topic.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

$$P(w|d) = \sum_{t=1}^{T} p(w|t)\, p(t|d)$$

Step-1

Create a document term matrix that shows a corpus of N documents D1, D2, D3 … Dn and vocabulary size of M words W1, W2 .. Wn. In that matrix, a particular cell (i, j) represents the frequency count of word Wj in the Document Di of the corpus.

	W1	W2	W3	Wn
D1	0	2	1	3
D2	1	4	0	0
D3	0	2	3	1
Dn	1	1	3	0

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

$$P(w|d) = \sum_{t=1}^{T} p(w|t)\, p(t|d)$$

Step-2

LDA converts this Document-Term Matrix into two lower dimensional matrices, M1 and M2 where M1 and M2 represent the document-topics and topic–terms matrix with dimensions (N, K) and (K, M) respectively, where

- N is the number of documents,
- K is the number of topics,
- M is the vocabulary size.

For Example, A sample matrix M1 is described below:

	K1	K2	K3	K
D1	1	0	0	1
D2	1	1	0	0
D3	1	0	0	1
Dn	1	0	1	0

(document-topics)

	W1	W2	W3	Wm
K1	0	1	1	1
K2	1	1	1	0
K3	1	0	0	1
K	1	1	0	0

(document-words)

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

LDA – latent Dirichlet Allocation

$$P(w|d) = \sum_{t=1}^{T} p(w|t)\, p(t|d)$$

Step-3

In this step, we iterate through each word “w” present in each of the document “d” and tries to adjust the current topic – word assignment with a new assignment.
A new topic “k” is assigned to the word “w” with a probability P which is the multiplication of two probabilities p1 and p2.

- p1: **p(topic t / document d)** represents the proportion of words in document d that are currently assigned to topic t.

- p2: **p(word w / topic t)** represents the proportion of assignments to topic t over all documents that come from this word w.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Credit: Higher School of Economics

LDA – latent Dirichlet Allocation

$$P(w|d) = \sum_{t=1}^{T} p(w|t)\, p(t|d)$$

Step-3

In this step, we iterate through each word “w” present in each of the document “d” and tries to adjust the current topic – word assignment with a new assignment.
A new topic “k” is assigned to the word “w” with a probability P which is the multiplication of two probabilities p1 and p2.

For every topic, the following two probabilities p1 and p2 are calculated.

- p1: **p(topic t / document d)** represents the proportion of words in document d that are currently assigned to topic t.

- p2: **p(word w / topic t)** represents the proportion of assignments to topic t over all documents that come from this word w.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Credit: Higher School of Economics

LDA – latent Dirichlet Allocation

$$P(w|d) = \sum_{t=1}^{T} p(w|t)\, p(t|d)$$

Step-4

The current topic – word assignment is updated with a new topic with the probability, which is the product of p1 and p2 probabilities.
In this step, the model assumes that all the existing word–topic assignments except the current word are correct. This is essentially the probability that topic t generated word w, so it makes sense to adjust the current word's topic with a new probability.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Credit: Higher School of Economics

LDA – latent Dirichlet Allocation

$$P(w|d) = \sum_{t=1}^{T} p(w|t)\, p(t|d)$$

Step-5

The current topic – word assignment is updated with a new topic with the probability, which is the product of p1 and p2 probabilities.
In this step, the model assumes that all the existing word–topic assignments except the current word are correct. This is essentially the probability that topic t generated word w, so it makes sense to adjust the current word's topic with a new probability

Step-6

After a number of iterations, we achieved a steady-state where the document topic and topic term distributions are fairly good and This is considered as the convergence point for LDA.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Credit: Higher School of Economics

LDA – Matrix Factorization Intuition

suppose we have a document with some random word topic assignment, for example, as shown below:

India	enters	world	cup	final
1	3	1	2	4

We also have our count matrix v(k,w) as shown below:

	1	2	3	4
India	70	5	0	8
enters	2	3	15	6
world	28	4	12	1
cup	6	43	6	0
final	7	0	9	31

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

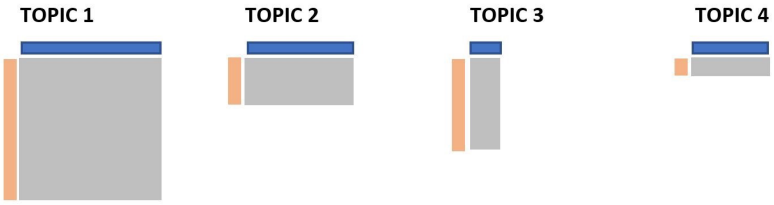
LDA – Matrix View

Now let's change the assignment of word **world** in the document.

- First, we will reduce the count of world in topic 1 from 28 to 27 as we don't know to what topic world belongs.
- Second let's represent the matrix *n(d,k)* in the following way to show how much a document use each topic

- Let's represent *v(k,w)* in the following way to show how many times each topic is assigned to this word

- multiply these two matrices to get our conditional probabilities



Sources: Attribution and credit to var applied

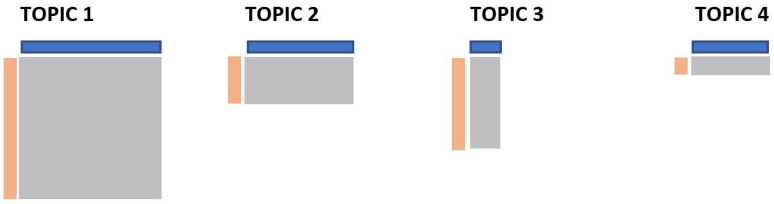
LDA – Matrix Factorization

Now let's change the assignment of word **world** in the document.

- First, we will reduce the count of world in topic 1 from 28 to 27 as we don't know to what topic world belongs.
- Second let's represent the matrix *n(d,k)* in the following way to show how much a document use each topic

- Let's represent *v(k,w)* in the following way to show how many times each topic is assigned to this word

- multiply these two matrices to get our conditional probabilities



*we will randomly pick any of the topic and will assign that topic to **world** and we will repeat these steps for all other words as well. Intuitively, topic with highest conditional probability should be selected but as we can see other topics also have some chance to get selected*

Sources: Attribution and credit to var applied

LDA – latent Dirichlet Allocation

Some of the advantages of LDA are as follows:

Fast

The model is usually fast to run. But of course, it depends on your data. You can verify it by using the %time command in Jupyter Notebook.

Several factors which can slow down the model are as follows:

- Very long documents
- A large number of documents in the corpus
- Large vocabulary size, especially when you use n-grams with a very high value of n

Intuitive

This Modelling approach to extract the topics gives weighted lists of words which is a very simple approximation yet a very intuitive approach for interpretation, as there is no embedding nor hidden dimensions, just bags of words with corresponding weight values.

Can predict topics for new unseen documents

Once your model is trained, it is ready to allocate topics to any document.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Credit: Higher School of Economics

LDA – latent Dirichlet Allocation

Disadvantages of LDA
Some of the disadvantages of LDA are as follows:

Requires Lots of fine-tuning
If LDA is fast to run, it will give you some trouble to get good results with it. That's why knowing in advance how to fine-tune it will really help you.

Needs Human Interpretation
After finding the Topics from the set of documents with the help of a machine, we also require manual human efforts to label them in order to present the results to non-experts people.

You cannot influence topics
Sometimes what happens is that based on our prior knowledge we know some of the topics that your documents talk about, but when we do the same thing with the help of LDA, you will not find those topics, which will definitely be frustrating for you. And there is no method to say to the model that some words should belong together. So, you have to sit and wait for the LDA to give you what you want.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Image Credit: Higher School of Economics

NMF – Non negative Matrix Factorization

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

NMF – Non negative Matrix Factorization

NMF is a statistical method that helps us to reduce the dimension of the input corpora or corpora.

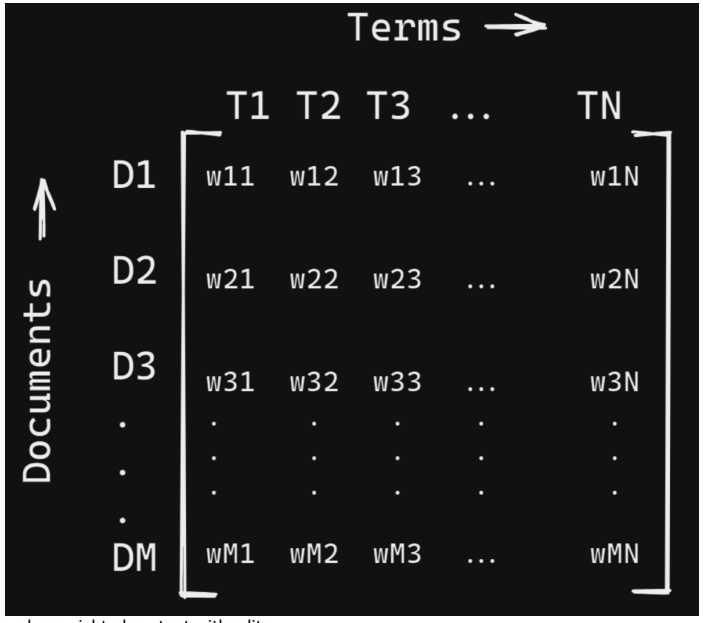
Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

NMF – Non negative Matrix Factorization

NMF is a statistical method that helps us to reduce the dimension of the input corpora or corpora.

Leverage the matrix structure of Document-Word/terms/tokens

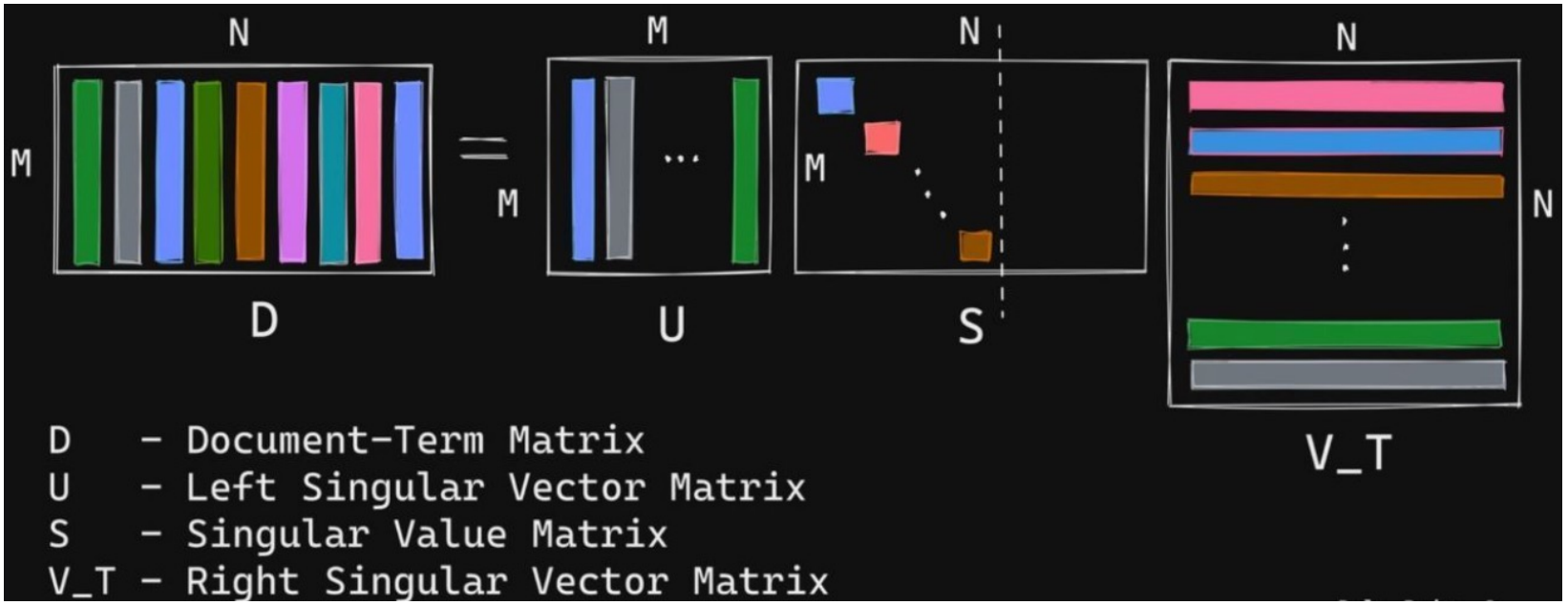
matrix factorization via SVD will give use orthogonal topics



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

SVD

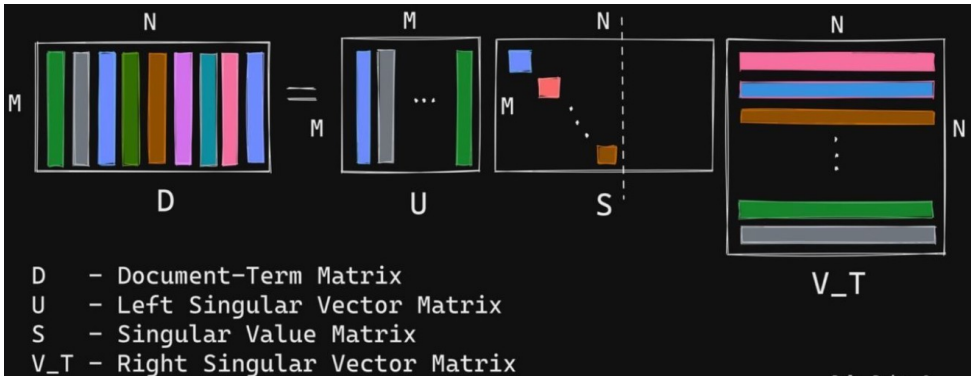
SVD

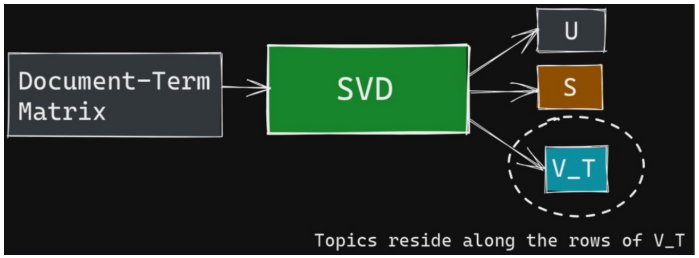


Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

SVD to find Topics

SVD of Document Term Matrix





•The left singular vector matrix **U -** The i,j-th entry of the document similarity matrix signifies how similar document i is to document j.

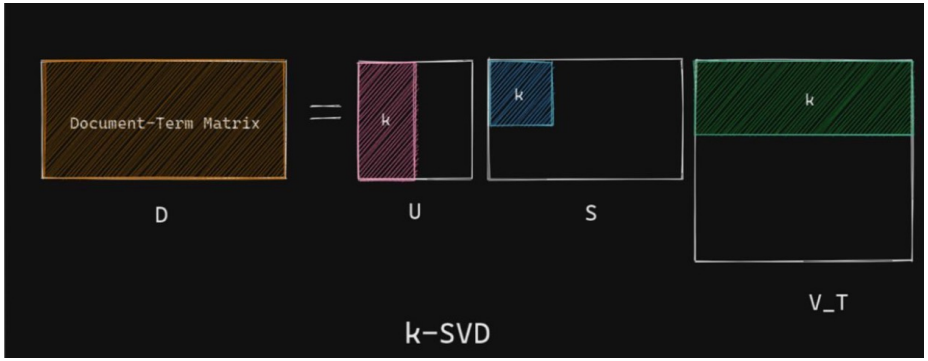
•The matrix of singular values **S**, which (values) signify the relative importance of topics.

•The right singular vector matrix **V_T**, which is also called the term topic matrix. The topics in the text reside along the rows of this matrix.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

NMF – Non negative Matrix Factorization

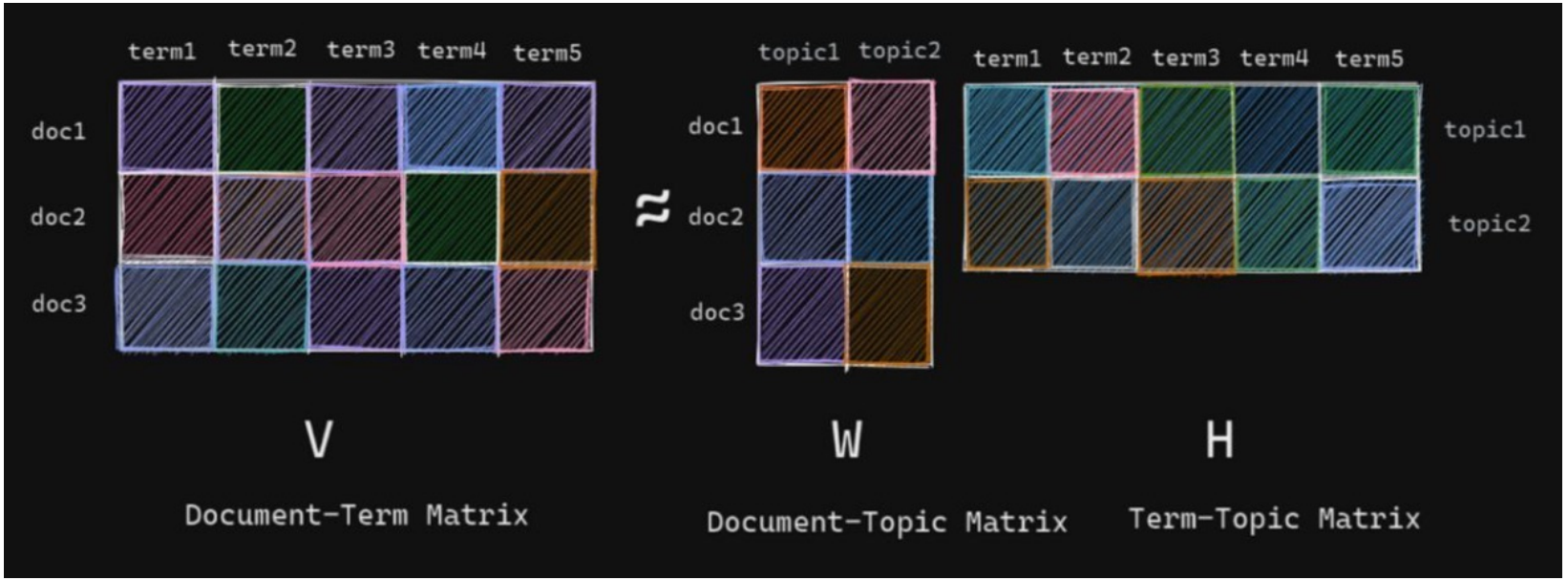
K-SVD of Document Term Matrix



•fix a small number of topics that best convey the content of the text.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Topic Modeling using NMF



- The matrix **W** which is called the **document-topic matrix**. This matrix shows the distribution of the topics across the documents in the corpus.
- The matrix **H** which is also called the **term-topic matrix**. This matrix captures the significance of terms across the topics.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Topic Modeling using NMF

- NMF is a *non-exact* matrix factorization technique.
- The matrices W and H are initialized randomly.
- Optimized iteratively to minimize the cost function between original and reconstructed document term matrix

$$minimize \ ||V - WH||_F$$
$$where, \ V : Document - Term \ Matrix$$
$$W : Document - Topic \ Matrix$$
$$and \ H : Term - Topic \ Matrix$$

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Topic Modeling using NMF

- NMF is a *non-exact* matrix factorization technique.
- The matrices W and H are initialized randomly.
- Optimized iteratively to minimize the cost function between original and reconstructed document term matrix

$$minimize \ ||V - WH||_F$$
$$where, \ V : Document - Term \ Matrix$$
$$W : Document - Topic \ Matrix$$
$$and \ H : Term - Topic \ Matrix$$

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$$

Frobenius norm

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

NMF – Non negative Matrix Factorization

We can calculate matrices W and H by optimizing over an objective function (like the EM algorithm), and updates both the matrices W and H iteratively until convergence.

$$\frac{1}{2}||\mathbf{A}-\mathbf{WH}||_{\mathrm{F}}^{2}=\sum_{i=1}^{n}\sum_{j=1}^{m}(A_{ij}-(WH)_{ij})^{2}$$

measure the error of reconstruction between the matrix A and the product of its factors W and H, on the basis of Euclidean distance.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

NMF – Non negative Matrix Factorization

Updating Matrix W:

$$W_{ic} \leftarrow W_{ic} \frac{(\mathbf{A}H)_{ic}}{(\mathbf{WH}H)_{ic}}$$

Updating Matrix H:

$$H_{cj} \leftarrow H_{cj} \frac{(W\mathbf{A})_{cj}}{(W\mathbf{WH})_{cj}}$$

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

NMF – Non negative Matrix Factorization

Updating Matrix W:

$$W_{ic} \leftarrow W_{ic} \frac{(\mathbf{A}H)_{ic}}{(\mathbf{WHH})_{ic}}$$

Updating Matrix H:

$$H_{cj} \leftarrow H_{cj} \frac{(W\mathbf{A})_{cj}}{(W\mathbf{WH})_{cj}}$$

parallelly update the values and using the new matrices that we get after updation W and H

Iterate till

- product of these matrices approaches to A; or
- the approximation error converges; or
- the maximum iterations are reached.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

NMF – Non negative Matrix Factorization

Some heuristics to initialize the matrix W and H

- Randomly
- TF-IDF weights
- **BOW**
- word vectors

better initial estimates with the aim of converging more rapidly to a good solution.

- rank-r approximation of A using SVD
- Picking *r* columns of A and just using those as the initial values for W.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Topic Modeling using Embeddings

Doc2Vec

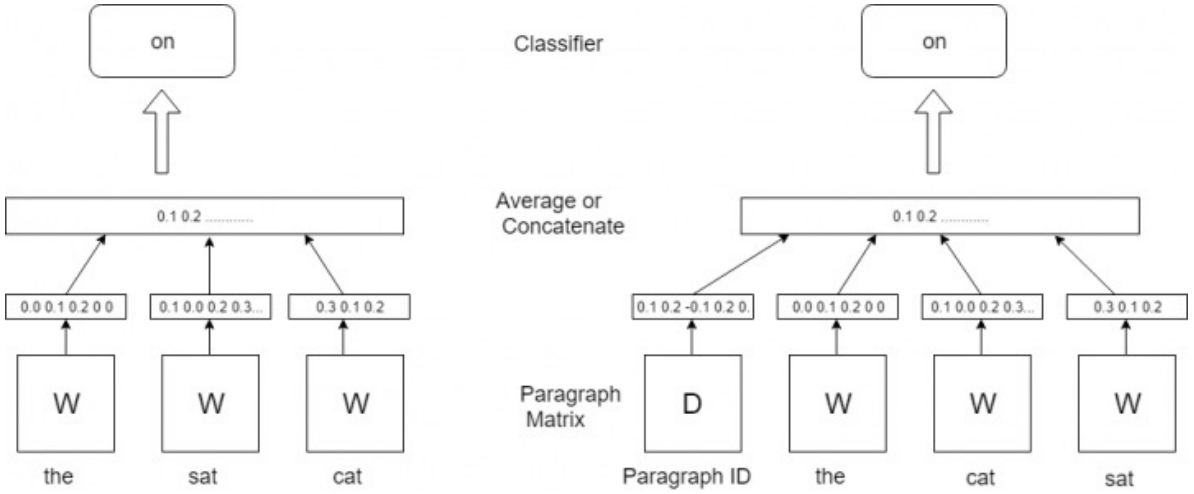
Sentence Embedding

Embedding Clustering

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Topic Modeling using Embeddings

Doc2Vec



Distributed Memory version of Paragraph Vector (PV-DM).
It acts as a memory that remembers what is missing from the current context — or as the topic of the paragraph.
While the word vectors represent the concept of a word, the document vector intends to represent the concept of a document.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

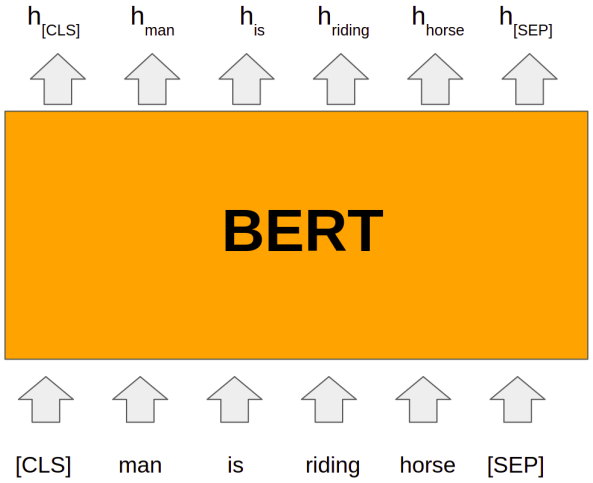
Topic Modeling using BERT (like) models

Sentence Embedding

Embedding Clustering

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

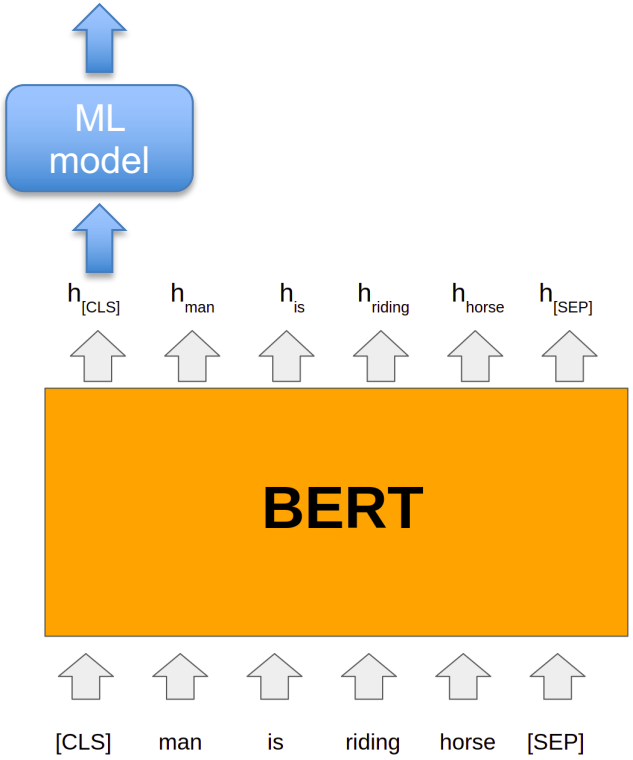
BERT – High Level Architecture



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

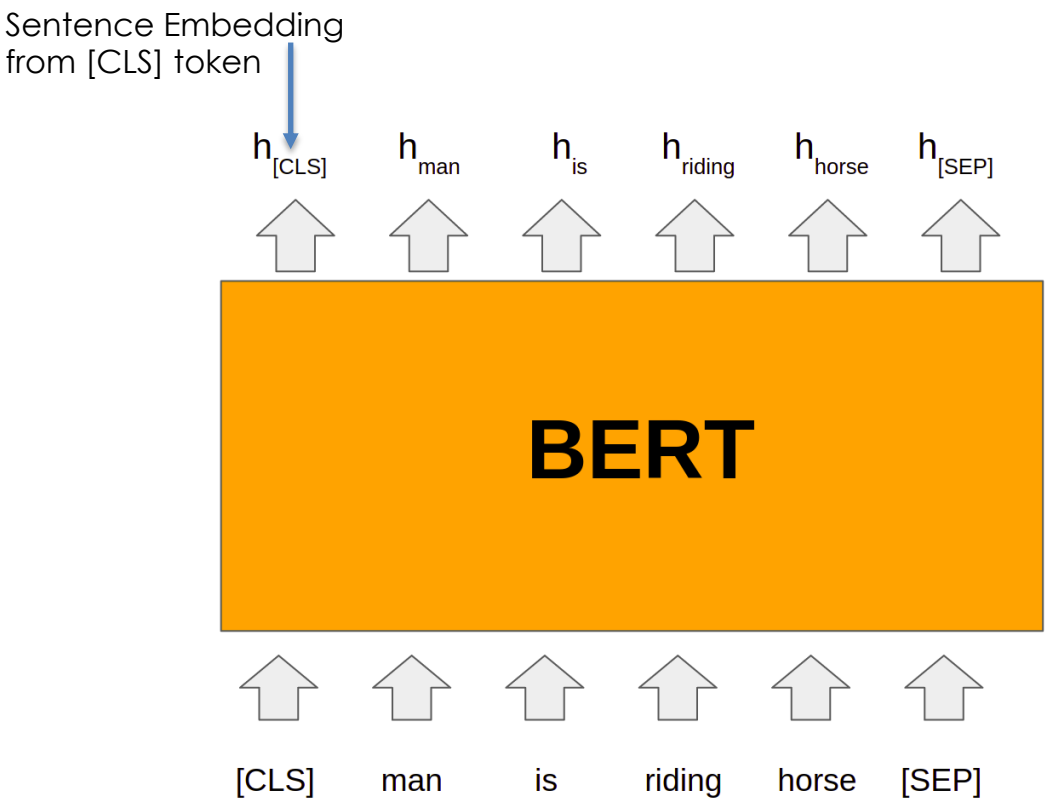
BERT – How BERT is used via transfer learning

Downstream task (classification, regression etc)



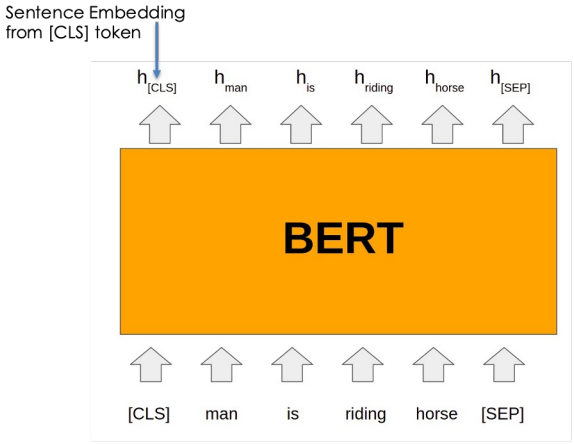
Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

BERT – Sentence Embedding



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

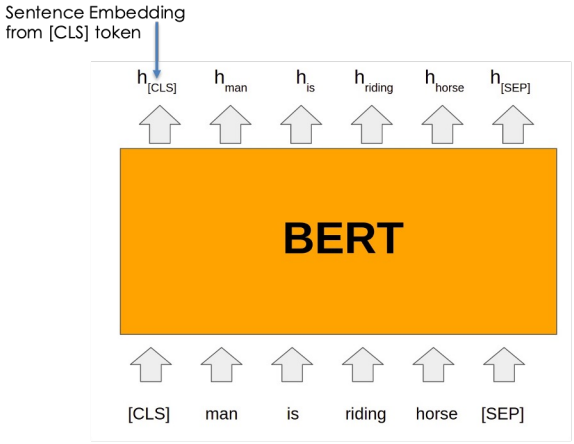
BERT – Sentence Embedding - different approaches



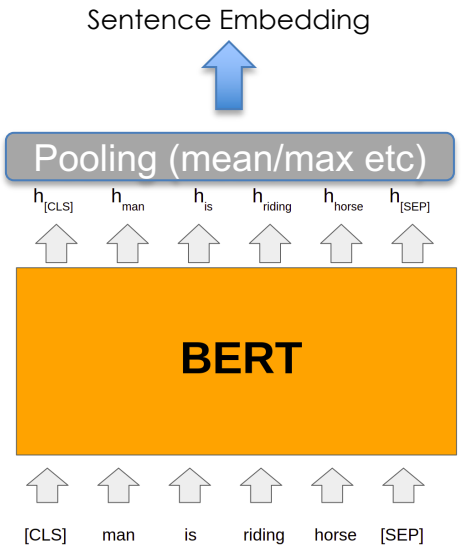
(1) From [cls] token

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

BERT – Sentence Embedding - different approaches



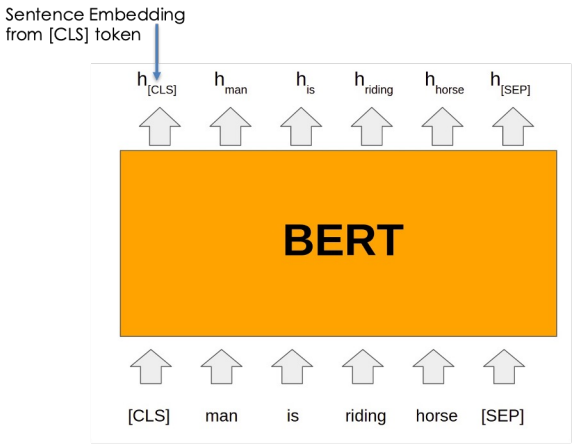
(1) From [cls] token



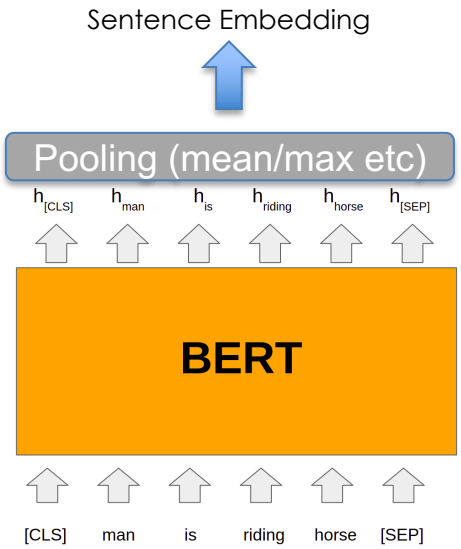
(2) From pooling

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

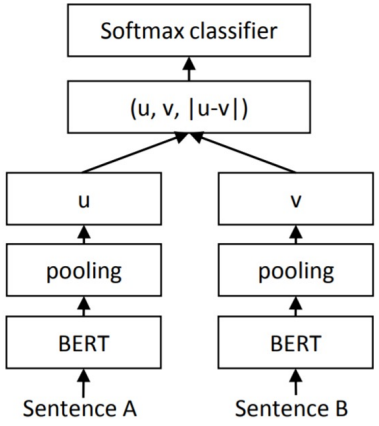
BERT – Sentence Embedding - different approaches



(1) From [cls] token



(2) From pooling



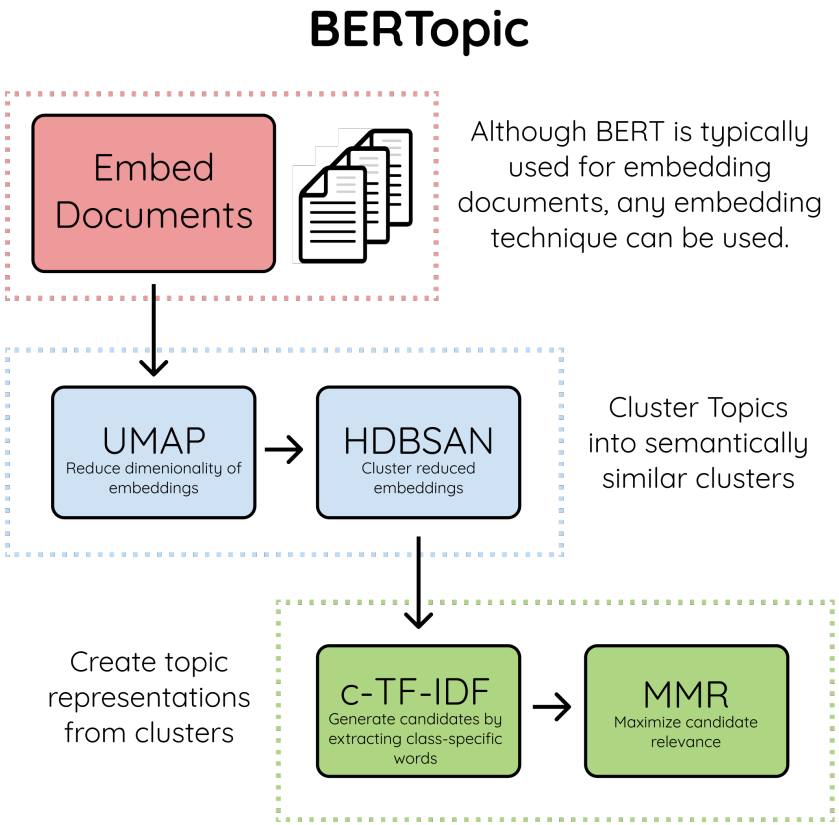
(3) From SBERT

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Topic Modeling using BERT (like) models

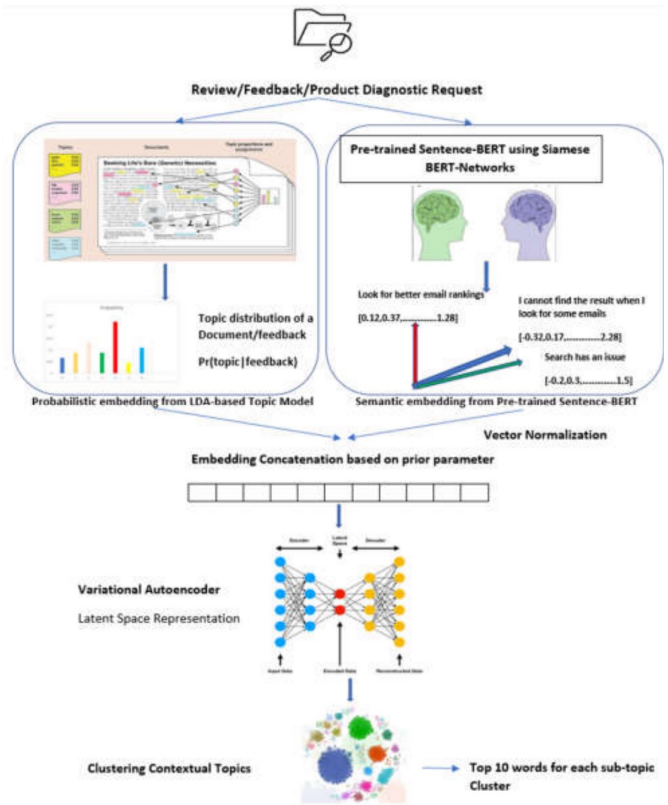
Sentence Embedding

Embedding Clustering



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Can hybrid/ensemble models perform better?



https://arxiv.org/pdf/2007.09303.pdf

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Evaluation of Topic Models

what approaches are commonly used for the evaluation:

Eye Balling Models
- Top N words
- Topics / Documents

Intrinsic Evaluation Metrics
- Capturing model semantics
- Topics interpretability

Human Judgements

Evaluation on downstream tasks
- Is model good at performing predefined tasks, such as classification

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

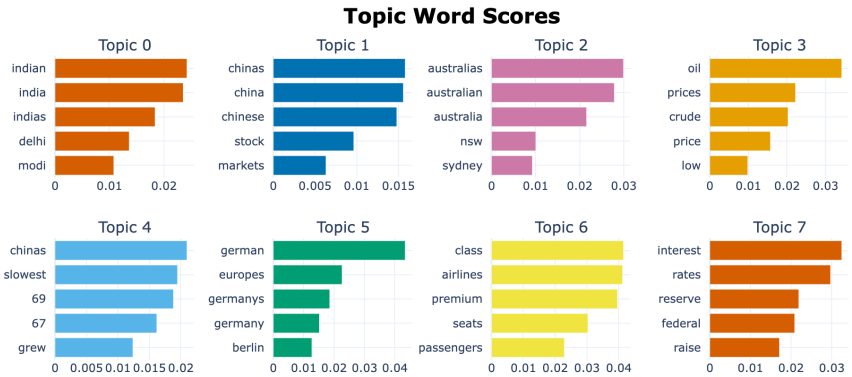
Evaluation of Topic Models

what approaches are commonly used for the evaluation:

Eye Balling Models

•Top N words

wordcloud



keywords

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Evaluation of Topic Models

what approaches are commonly used for the evaluation:

Topic Coherence

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Evaluation of Topic Models

what approaches are commonly used for the evaluation:

Intrinsic Evaluation Metrics
- Capturing model semantics
- Topics interpretability

Topic Coherence

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Evaluation of Topic Models

Topic Coherence

Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic. T

Coherence Measures

Let's take quick look at different coherence measures, and how they are calculated:

1.C_v measure is based on a sliding window, one-set segmentation of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity

2.C_p is based on a sliding window, one-preceding segmentation of the top words and the confirmation measure of Fitelson's coherence

3.C_uci measure is based on a sliding window and the pointwise mutual information (PMI) of all word pairs of the given top words

4.C_umass is based on document cooccurrence counts, a one-preceding segmentation and a logarithmic conditional probability as confirmation measure

5.C_npmi is an enhanced version of the C_uci coherence using the normalized pointwise mutual information (NPMI)

6.C_a is baseed on a context window, a pairwise comparison of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity

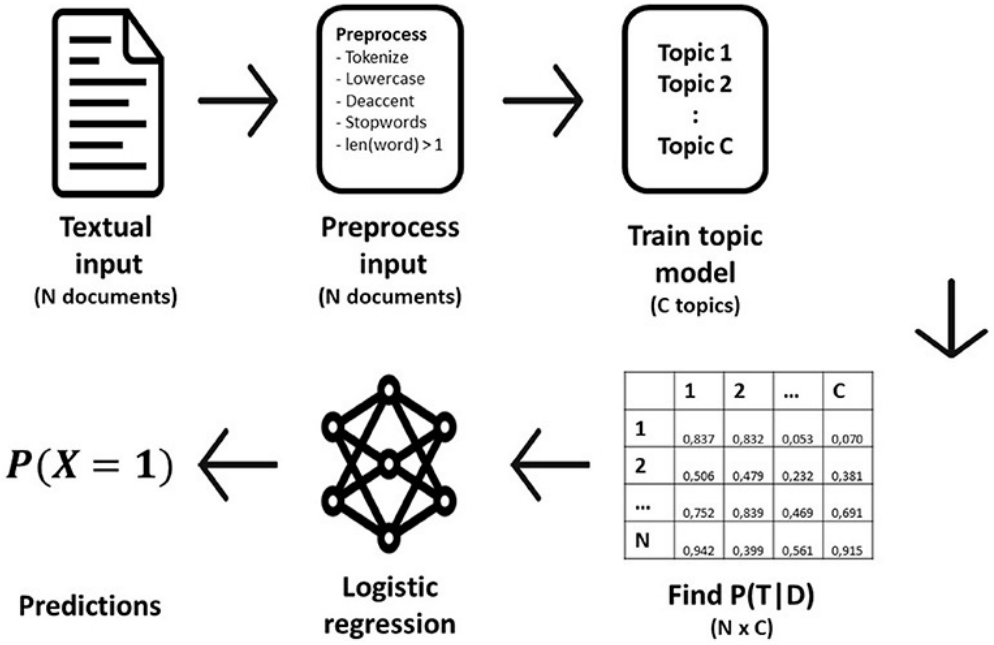
Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Downstream Tasks – Classification?

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Evaluation of Topic Models

Downstream Tasks – Classification?



	1	2	...	C
1	0,837	0,832	0,053	0,070
2	0,506	0,479	0,232	0,381
...	0,752	0,839	0,469	0,691
N	0,942	0,399	0,561	0,915

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Zero-shot Topic Prediction

- Firstly, the reviews are put into a list for the pipeline.
- Then, the candidate labels are defined.
- Finally, the text, the candidate labels, and the hypothesis template are passed into the zero-shot classification pipeline called classifier.

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Zero-shot Topic Prediction

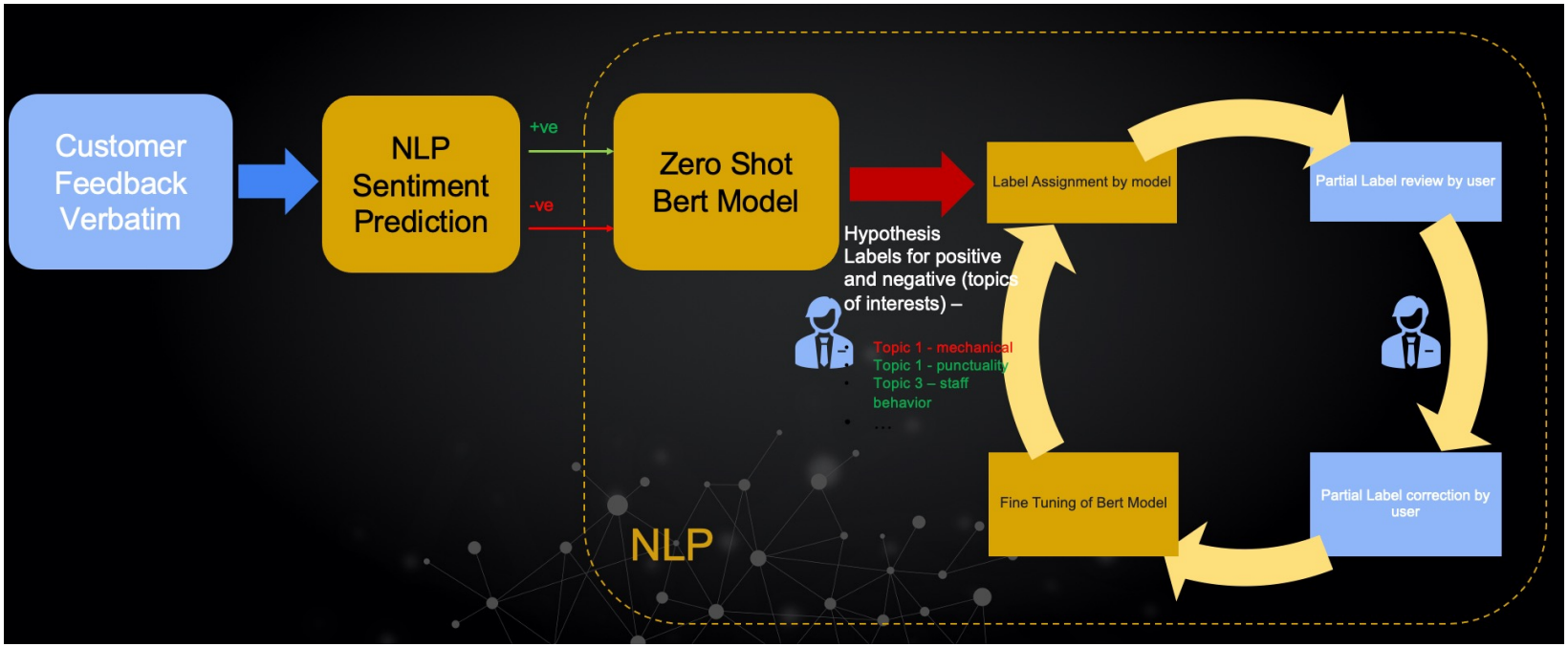
- Firstly, the reviews are put into a list for the pipeline.
- Then, the candidate labels are defined.
- Finally, the text, the candidate labels, and the hypothesis template are passed into the zero-shot classification pipeline called classifier.

Can this be reviewed and improved

MLOps deployment pattern

Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied

Topic Modeling with Human in the Loop



Sources: Attribution and credit to various internet sourced and copyrighted content with edits applied