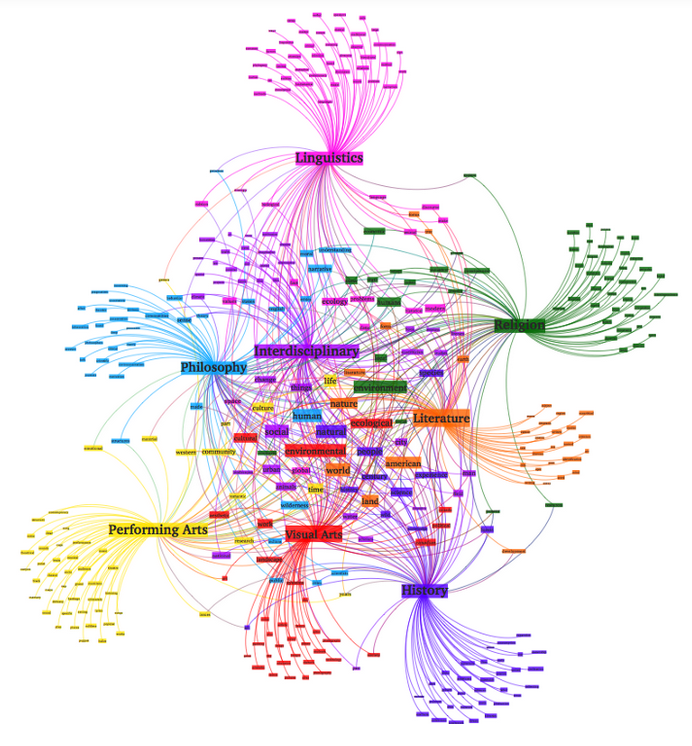
**Topic Modeling or Analysis**



In Natural Language Understanding(NLU) tasks there is a hierarchy of lenses or different lenses through which we can extract meaning from words to sentences to paragraphs to documents.

At the **document leve**l, one of the most useful ways to understand text is by analyzing its topics.

The process of learning, recognizing, and extracting these **topics** across a collection of documents is called **topic modelling.**

All topic models are based on the same basic assumption that

Each Document consists of different or mixture of topics, and

Each Topic consists of a collection of words

Topic models are built around the idea that the semantics (meaning) of our documents are actually being governed by some hidden or “latent” variables called **topics** that we are not observing.

As a result the goal of topic modeling is to uncover or reveal these latent(hidden) variables called topics that shape the meaning of our document and corpus.

**Topic Modeling Techniques**

currently there are 4 most popular topic modeling techniques.

1 LSA

2 pLSA

3 LDA

4 lda2vec

From those the first three are statistical based topic modellling but the 4th one is the newer, deep learning based topic modelling technique.

**1 LSA(Latent Semantic Analysis)**

**Latent** → present but need particular conditions to be revealed or visible. It means currently its hidden or concealed state.

Semantic → related to meaning in a language.

So Latent Semantic Analysis means the analysis or extraction of hidden meaning of a documents, which is a topic.

LSA is quick and efficient to use, but it does have a few primary drawbacks:

* lack of interpretable embeddings (we don’t know what the topics are, and the components may be arbitrarily positive/negative)
* need for really large set of documents and vocabulary to get accurate results
* less efficient representation

**What is Gensim ?**

Gensim is a free python library designed to automatically extract semantic topics from documents, as efficiently(computer-wise) and painlessly(human-wise) as possible.

Gensim is designed to process raw, unstructured digital texts(‘plain texts)

Gensim is a NLP package or Library that contains efficient implementations of many well known functionalities for the task of topic modeling such as tf-idf, LDA, LSA.

The algorithims in Gensims are like

* Word2Vec
* FastText
* LSA
* LDA
* tf-idf

these algorithim in Gensim library automatically discover the semantic structure of documents by examining statistical co-occurrence patterns within a corpus of training documents.

These algorithims are Unsupervised, which means No human input is necessary – you only need a corpus of plain text document.

**What does it mean by Document in Text analysis ?**

Most of the time in Text Analysis Document means not the file itself, but it means one line in a text.

And in a text one line can contains one sentence or 2 or 10 or many sentence can be just a single Sentence in a text file. So document can be single sentence or multiple sentence just in a single line.

But what if single line contains only just one word not sentence can we count this single word as document.

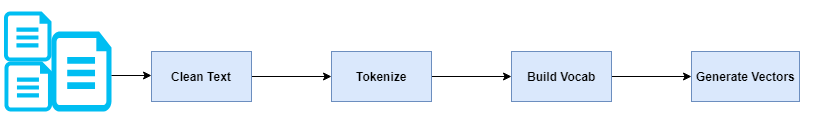
**e.g** the ff file.txt contains 4 documents because it contains 4 lines

1“*It was the best of times”  
2“It was the worst of times”  
3“It was the age of wisdom”  
4“It was the age of foolishness”*



**What is Bag of Words(BOW)**

* we need a way to represent text data for machine learning algorithim and the bag of words model helps us to achieve that task.
* Bow model is simple to understand and implement
* it is a way of extracting features from the text for use in machine learning algorithim
* in simple terms bag of word is just a collection of words to represent a sentence witht word count and mostly disregarding the order in which they appear.
* to create bag-of-words
  + 1st reads all the lines or documents in the file or texts as a lists of lines
  + then make tokenization means change all the list into list of tokens
  + then create a dictionary(Vocabulary) of tokens from all the documents with keys as the word and value as the frequency of that word in the whole texts. So this dictionary will contains N size of unique words.
  + Then Order this dictionary based on their frequency value. Large value first seat.
  + then make a Vectorization, means create a vectors with size 1xN for each lines or documents in the file. Note this vector contains value of 1 and O.
  + take the first word in dictionary and the first word from sentence if they are the same puts 1 if not puts 0, …
  + like this you will create a vector for each line and finally you can append all the list into one list as matrix with size of MXN.
  + M is the number of lines or documents.
  + N is the total number of dictionary words.
  + So this big list or matrix is called bag-of-words, which contains all the text data in the form of vectors, or 0 and 1 values.



The Process of converting NLP texts into Number is called Vectorization in ML.

And there are different ways to convert text into vectors

* CountVectorizer
  + based on term frequency
  + calculating frequency of each word in whole document
* TFIDF
  + based on term frequency and inverse frequency
  + calculating frequency of each word in whole document except current document.

But I dont understand this one

**simple example of bag-of-word.**

1“*It was the best of times”  
2“It was the worst of times”  
3“It was the age of wisdom”  
4“It was the age of foolishness”*

*size of documents(M=4)*

***dictionary:*** *[*‘*It’, ‘was’, ‘the’, ‘best’, ‘of’, ‘times’, ‘worst’, ‘age’, ‘wisdom’, ‘foolishness’]*

*size of dictionary(N=10)*

**Vectorization** size each 1 x 10

“*It was the best of times” = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]  
“It was the worst of times” = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]  
“It was the age of wisdom” = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]  
“It was the age of foolishness” = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]*

so **Bag-of words** is size (4 x 10)

[  
*[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]*

*[1, 1, 1, 0, 1, 1, 1, 0, 0, 0]*

*[1, 1, 1, 0, 1, 0, 0, 1, 1, 0]*

*[1, 1, 1, 0, 1, 0, 0, 1, 0, 1]*

]

**What is the difference between Bag-of-words and Word2Vec ?**

**LDA Topic Modelling in Python with NLTK and Gensim**

Research paper topic modelling is an unsupervised machine learning method that helps us to discover hidden semantic structures in a paper.

Research paper text data is just a bunch of unlabeled texts and can be found here

This topic model can be applied to any kinds of labels on documents.

The Process of Topic Modelling or Analysis

* we can pick the number of topics a head of time, even if we’re not sure what topics are
* each document is represented as a distribution over topics and
* each topic is represented as a distribution over words.

**Text Cleaning**

we use python library called spacy to clean our texts and return a list of tokens

**what is spaCy ?**

* Industrial Strength Natural Language Processing
* its a free open-source library for NLP in Python.
* with spaCy, you can easily construct linguistically sophisticated statistical models for a variety of NLP problems
* its the best way to prepare text for deep learning
* it interoperates seamlessly with Tensorflow, PyTorch, Scikit-Learn, Gensim and the rest of Python’s awesome AI ecosystem.

What is pickle Library ?

**What is pyLDAvis ?**

**→ Python LDA Visualization**

Python library for interactive topic model visualization.

\*\*pyLDAvis\*\* is designed to help users interpret the topics in a topic model that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualization.

pip install pyldavis or

conda install -c conda-forge pyldavis

Saliency: a measure of how much the term tells you about the topic.

Relevance: a weighted average of the probability of the word given the topic and the word given the topic normalized by the probability of the topic.

The size of the bubble measures the importance of the topics, relative to the data.

First, we got the most salient terms, means terms mostly tell us about what’s going on relative to the topics. We can also look at individual topic.

**Summary of LDA in Topic Analysis Written by Eyob**

**Libraries Used**

**psutil(process and system utilities)**

* is a cross-platform library for retrieving information on **running processes**and **system utilization** (CPU, memory, disks, network, sensors) in Python.
* It is useful mainly for **system monitoring**, **profiling and limiting process resources** and **management of running processes**.
* It implements many functionalities offered by UNIX command line tools such as: ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap.

**Questions.**

**1** my big questions in Topic Modeling is does it generates a new topic for that Documents, or does it just selects the word that is available in the documents as a topic.

If its going to generate a topic by itself, it have to understand the whole documents.

But if its just going to selects the word from the document as a topic, I think it does not need to understand the documents, since it can just selects the word by some probability and statistical methods.

2. when we say term is its just a single word or some collection of word in a sentence or just one sentence.

Also when we find tf-idf is we find for the single word or for some collection of word or for sentence in a document. To find the topic.

3. how about the word order of sentence in bag of words ?

References

Bag of Words

<https://www.freecodecamp.org/news/an-introduction-to-bag-of-words-and-how-to-code-it-in-python-for-nlp-282e87a9da04/>