**Topic Modeling on News Articles**

**Manoj Kumar**

**1.Data Description:**

The dataset contains a set of news articles for each major segment consisting of business, entertainment, politics, sports and technology. You need to create an aggregate dataset of all the news articles and perform topic modeling on this dataset. Verify whether these topics correspond to the different tags available.

**2.Problem Description:**

In this project your task is to identify major themes/topics across a collection of BBC news articles. You can use clustering algorithms such as Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA) etc.

**3. Methods involved:**

we are using a two methods here

1.Topic Modeling on News Articles using spacy

2.Topic Modeling on News Articles Using NLTK\_Gensim

* **Topic Modeling on News Articles using spacy**

Unlike NLTK, which is widely used for teaching and research, spaCy focuses on providing software for production usage.[5][6] spaCy also supports deep learning workflows that allow connecting statistical models trained by popular machine learning libraries like TensorFlow, PyTorch or MXNet through its own machine learning library Thinc.[7][8] Using Thinc as its backend, spaCy features convolutional neural network models for part-of-speech tagging, dependency parsing, text categorization and named entity recognition (NER).

* **Topic Modeling on News Articles Using NLTK\_Gensim**

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.

**steps involved in nltk:**

* Loading the data
* Clean the Data
* Handling the Stop-words
* Lemmatization
* TF-IDF Vectorization
* Topic modelling using LSA
* Topic Modelling using Gensim's LDA

**4. Lemmatization:**

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meanings to one word.

Text preprocessing includes both Stemming as well as Lemmatization. Many times people find these two terms confusing. Some treat these two as the same. Actually, lemmatization is preferred over Stemming because lemmatization does morphological analysis of the words.

Applications of lemmatization are

* Used in comprehensive retrieval systems like search engines.
* Used in compact indexing

**5.TF-IDF Vectorization:**

Bag of words (BoW) converts the text into a feature vector by counting the occurrence of words in a document. It is not considering the importance of words. Term frequency — Inverse document frequency (TFIDF) is based on the Bag of Words (BoW) model, which contains insights about the less relevant and more relevant words in a document. The importance of a word in the text is of great significance in information retrieval.

Example — If you search something on the search engine, with the help of TFIDF values, search engines can give us the most relevant documents related to our search.

**6.Topic modelling using LSA:**

Basic assumptions on which all topic modeling algorithms are based:

Each document consists of more than one topics, and

Each topic consists of a collection of words.

In other words, topic modeling algorithms are built around the idea that the semantics of our document is actually being governed by some hidden, or “latent,” variables that we are not observing directly after seeing the textual material.



As a result, to uncover these latent variables which shape the meaning of our document and corpus, we require topic modeling algorithms. In the later part of this blog post, we will build up an understanding of how different topic models uncover these latent topics. But in this article, we will discuss the LSA technique first, and then as we go ahead we will also discuss different techniques of Topic modeling such as LDA, pLSA, etc.

Why do we need Latent Semantic Analysis?

As we discussed starting from the first article, we have discussed that all-natural languages have their own intricacies and nuances which are quite difficult for a machine to capture, and also sometimes they’re even misunderstood by us humans!. This involves different words that have the same thing, and also the words which have the same spelling but gives different meanings.

For Example, consider the following two sentences:

Sentences:

I liked the last novel of Premchand quite a lot.

They would like to go for a novel marketing campaign.

In the first sentence, the word ‘novel’ represents a book, while in the second sentence, it means new or fresh.

We as a human can easily differentiate between these two words since we can understand the context behind these two words. However, the machines would not be able to capture this concept as they cannot understand the context in which the words have been used. This is where the role of Latent Semantic Analysis (LSA) comes into the picture!

LSA tries to leverage the context around the words to capture the hidden or latent concepts, which are called topics.

So, if we simply mapped the words to documents, then it won’t really helpful for us. So, What we really require is to extract the hidden concepts or topics behind the words. LSA is one such technique that can be used to find these hidden topics

**7. Topic Modelling using Gensim's LDA:**

Topic Modeling is a technique to extract the hidden topics from large volumes of text. Latent Dirichlet Allocation(LDA) is a popular algorithm for topic modeling with excellent implementations in the Python’s Gensim package. The challenge, however, is how to extract good quality of topics that are clear, segregated and meaningful. This depends heavily on the quality of text preprocessing and the strategy of finding the optimal number of topics.

Can LDA be used for topic Modelling?

Latent Dirichlet Allocation (LDA) is an example of topic model and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

The Work Flow for executing LDA in Python

After importing the required libraries, we will compile all the documents into one list to have the corpus.

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)

**8. Conclusion:**

* Loading the data
* Clean the Data
* Transforming text into something an algorithm can digest it a complicated process. We cannot feed the data as it is, some preprocessing needs to be done. In this task we will be doing some preprocessing to convert our data in a form that we can feed our model with.
* Handling the Stop-words
* Text may contain stop words like ‘the’, ‘is’, ‘are’. Stop words can be filtered from the text to be processed. There is no universal list of stop words in nlp research, however the nltk module contains a list of stop words. We will remove these stopwords in this task.
* Lemmatization
* TF-IDF Vectorization
* Apart from Count vectorizer an alternative to calculate word frequencies , and by far the most popular method is called TF-IDF. This is an acronym than stands for “Term Frequency – Inverse Document” Frequency which are the components of the resulting scores assigned to each word.
* Topic modelling using LSA
* Latent Semantic Analysis, or LSA, is one of the foundational techniques in topic modeling. The core idea is to take a matrix of what we have — documents and terms — and decompose it into a separate document-topic matrix and a topic-term matrix.
* Topic Modelling using Gensim's LDA
* One of the drawbacks of LSA is that though it is really fast, its effectiveness in finding good topics is not great. One assumption that LSA makes is that the topics are orthogonal to each other, while Latent Dirichlet Allocation (LDA) relaxes this assumption. Moreover, LDA generalizes the way the documents are generated and this modelling assumption leads to better topics.