Ayanlowo Babatunde

Summary

The article provides a comprehensive guide to topic modeling using LDA

Understanding Topic Modelling Models: LDA, NMF, LSI, and their implementation

Introduction

Natural language processing is the processing of languages used in the system that exists in an nltk library to process by transforming text dataset to new analyzable dataset for insights. If an NLP processing is done on another language, you have to add that language to the existing NLP library. NLP is mainly used in text processing, and there are many kinds of tasks that can be made easier using NLP examples are chatbots, Autocorrection, Speech Recognition, Language translator, Social media monitoring, Hiring and

Search, Machine Learning.

This article focused on Topic Modelling and the comparison of three common topic models, namely; Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), and Latent Semantic Indexing (LSI).

TOPIC MODELLING

According to Wikipedia, Topic modeling is “a statistical model for determining abstract topics that appear in a collection of documents.” Given that we are talking about a document containing politics, it is evident that the main topics for each of these documents will be related to political information; however, topic modeling looks for specific words found within each of the cluster’s top issues as well as potential relationships between the different clusters.

Topic modeling is an unsupervised method similar to a clustering algorithm that detects patterns and divides the data into various pieces. Topic modeling also learns about the various themes by analyzing the manuscript’s word patterns clusters and terms frequency. Therefore, based on the above documents are divided into different topics. As the procedures for dividing the topics do not have any outputs through which the task can be done, it is an unsupervised learning method. This type of modeling is beneficial when we have many documents and are willing to know what information is

can leverage NLP topic modeling for very little time.

The remaining sections describe the step-by-step process for topic modeling using LDA, NMF, LSI models. The implementation is done on Gutenberg's online book "Title of Book: A secret Service: Being tale of Nihilist" from the following links.

Topic Modelling using LDA, NMF, LSI

First, some of the essential topics which makes text processing easier in NLP topic labeling are the following:

a) Gathering dataset to be used for topic modeling

- Removing stopwords and punctuation marks
- Stemming
- Lemmatizing
- Encoding them to ML language using Countvectorizer or Tfidf vectorizer
- Displaying the essential topics from the document.

The dataset used for this topic modeling task was from Gutenberg websites (https://www.gutenberg.org/files/67278/67278-0.txt).

The data was extracted using the python URLLIB to request the text data from the link above. The extracted data were stored in a python list as shown in the following code snippet; get to know how urllib can be used in web scraping from websites using these links.

```
import urllib
url = "https://www.gutenberg.org/files/67278/67278-0.txt"
file = urllib.request.urlopen(url)
data=[]
for line in file:
  decoded_line = line.decode("utf-8")
  data.append(decoded_line)

  data
```

Text Mining hosted with ❤ by GitHub view raw

Data Preprocessing

To extract good quality of clear, segregated, and meaningful *topics. Quality of text preprocessing is highly necessary. This article attempts to tackle these problems by*

- *Removing emails, unwanted characters such as asci characters removing stopwords and punctuation*
- *Stemming*
- *Lemmatizing*

Removing Emails and Unwanted characters

Emails links and unwanted characters within the dataset extracted were removed by defining a using a regular expression in a user define function remove_emails_newlinexters() function as shown in the following code snippet.

Removing Stopwords and punctuations

Stopwords are the most common words in a natural language such as ‘the’, ‘is’, “in”, “for”, “where”, “when”, “to”, “at” etc. However, these stopwords might not add much value to the document’s meaning to analyze text data and build NLP models. On the other hand, punctuations include periods, question marks, exclamation points, commas, semicolons, colon, dash, hyphen, parentheses, brackets, braces, apostrophes, quotation marks, and

analysis, so we need to remove them from our datasets.

The following code snippet shows how stopwords and punctuation were removed from the text data extracted from Gutenberg.

```
#tokenization
def sent_to_words(sentences):
  for sentence in sentences:
    yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=True removes


#stop-words removal
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
def remove_stopwords(texts):
  return [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc
```

Stemming and Lemmatization

What is Stemming, Lemmatization? When Stemming is applied to the words in the corpus, the word gives the base for that particular word. E.g., fix, fixing, set provides fix when stemming is applied. There are different types of Stemming modules used in practice. Some popular ones are:

- Porter Stemmer
- Lancaster Stemmer
- Snowball Stemmer

cuts the word into its lemma word meaning to make it more meaningful than Stemming does. So the output we get after Lemmatization is called 'lemma.'

Through lemma, words are gotten; some methods are WordNet, TextBlob, Spacy, Tree Tagger, Pattern, Genism, and Stanford CoreNLP lemmatization.

The following shows the application of stemming and Lemmatization for our Gutenberg text data.

Topic Models

This article tutorial uses the following three topic models, namely:

- NMF
- LSI

Brief description LDA and NMF

In LDA, latent indicates the hidden topics present in the data, then Dirichlet is a form of distribution. Dirichlet distribution is different from the normal distribution. When ML algorithms are applied, the data must be normally distributed or follow Gaussian distribution. The normal distribution represents the data in real numbers format. In contrast, the Dirichlet distribution represents the data such that the plotted data sums up to 1. It can also be said as Dirichlet distribution is a probability distribution that is sampling over a probability simplex instead of sampling from the space of real numbers as in Normal distribution.

NMF Non-Negative Matrix Factorization is a statistical method that helps us to reduce the dimension of the input corpora or corpora. Internally, it uses the factor analysis method to give comparatively less weightage to words with less coherence.

Some Important points about NMF:

- It belongs to the family of linear algebra algorithms used to identify the latent or hidden structure present in the data.
- It is represented as a non-negative matrix.

document matrix, typically TF-IDF normalized.

Input: Term-Document matrix, number of topics.

Output: Gives two non-negative matrices of the original n-words by k topics and those same k topics by original documents.

In simple words, we are using linear algebra for topic modeling.NMF has become so popular because of its ability to automatically extract sparse and easily interpretable factors.

Interested in the mathematical background of NMF, I read this article.

(https://www.analyticsvidhya.com/blog/2021/06/part-15-step-by-step-guide-to-master-nlp-topic-modelling-using-nmf/)

IMPLEMENTATION OF LDA, NMF, AND LSI TO GUTERBURG DATASET
The following code snippet shows how the three topic models were applied to the test dataset from Gutenberg.

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import GridSearchCV

count_vectorizer = CountVectorizer( analyzer='word',
                                stop_words='english', lowercase=True,
                                token_pattern='[a-zA-Z\-][a-zA-Z\-]{2,}')
count_data_vectorized = count_vectorizer.fit_transform(data_lemmatized[0])

```

Visualization

```
# inspect the inferred topics
def print_topics(model, vectorizer, top_n=10):
    for idx, topic in enumerate(model.components_):
        print("Topic %d:" % (idx))
        print([(vectorizer.get_feature_names()[i], topic[i])
                        for i in topic.argsort()[:-top_n - 1:-1]])


#Displaying resultant model result using pylDAvis visualization tools.
pyLDAvis.enable_notebook()
panel = pyLDAvis.sklearn.prepare(lda_model_count, count_data_vectorized, count_vectorizer
```

Full code can be found here on my Github repository

*Quodos!!! 💅 💪 You have come to the End of this Article. Guess you Enjoy the article? Kindly click on the **FOLLOW** button at the left corner of this page for more related and impactful articles by me*

https://www.analyticsvidhya.com/blog/2021/05/topic-modelling-in-natural-language-processing/

https://www.analyticsvidhya.com/blog/2021/06/part-15-step-by-step-guide-to-master-nlp-topic-modelling-using-nmf/

https://www.machinelearningplus.com/nlp/topic-modeling-python-sklearn-examples/

Topic Modeling in Python with Gensim

Topic Modeling is a technique to understand and extract the hidden topics from large volumes of text. Latent Dirichlet...

www.machinelearningplus.com

NLP | Topic Modeling | Latent Semantic Indexing | Web Scraping Tools | Text Preprocessing

5 min read

DhanushKumar

Topic Modelling with BERTopic

BERTopic is a topic modeling technique that leverages BERT (Bidirectional Encoder Representations from Transformers), a powerful language...

29 min read

Rahul Kumar

Recommender Systems

Table of Contents 1. Content-based filtering 2. Collaborative filtering 3. Hybrid Models 4. Recommendation Evaluation metrics 5. Cold Start...

5 min read

Turibio Hilaire

200X Faster: Speed Up Your Music Production with AI

Boost Your Workflow and Create Music Faster Than Ever with The Help of AI

7 min read

Sandeep Kasinampalli Rajkumar

5 min read

Shenggang Li

Dynamic Weight Models: Bridging GLM and Neural Networks

Introduction

15 min read