**AI 620 Emerging Topics in Artificial Intelligence**

**HOS07A Neural Topic Model (NTM) in SageMaker**

03/20/2023 Developed by Yared Shewarade

09/21/2024 Updated by Anh Nguyen

9/15/2024 Reviewed by Jonathan Koerber

School of Technology and Computing (STC) @City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, the courses student worker for help.

**Learning Outcomes**

Students will be able to learn:

* Introduction to Neural Topic Model (NTM)
* Setup Notebook for SageMaker
* Training NTM in SageMaker
* Deploying NTM and running inference

**Resources**

* Tripuraneni, S., & Song, C. (2019). *Hands-on artificial intelligence on amazon web services: Decrease the time to market for AI and ML applications with the power of AWS* (1st ed.). Packt.

# Introduction to Neural Topic Model (NTM)

Topic modeling is the process of learning, recognizing, and extracting topics which is one of the most useful ways to understand text. Understanding topics in text can be used in the legal industry to surface themes from contracts, in the retails industry to identify broad trends in social media conversations, product innovation-introduce new merchandise into online and physical stores, to inform others of product assortment, and so on.

Structured and unstructured data are being generated at an unprecedented rate which is challenging to make the data useful. More than 80% of the data in enterprise is unstructured data that needs the right tools to organize, search, and understand this vast amount of information.

Text analysis is the process of converting unstructured text into meaningful data for analysis to support fact-based decision making. There are different techniques used for text analytics, such as topic modeling, entity and key phrases extraction, sentiment analysis, and coreference resolution.

Topic Modeling is used to organize a corpus of documents into “topics” which is a grouping based on a statistical distribution of words within the documents themselves. The technical definition of topic modeling is that each topic is a distribution of words, and each document is a mixture of topics across a set of documents. For example, a collection of documents that contains frequent occurrences of words such as ‘bike’, ‘car’, ‘mile’, ‘brake’ , and ‘speed’ are likely to share a topic on “transportation”. It can be used to summarize documents based on topic similarities.

The Neural Topic Model (NTM) is a generative document model that produces multiple representations of a document based on the vibrational autoencoder architecture. It generates two outputs:

* The topic mixture for a document
* A list of keywords that explain a topic, for all the topics across an entire corpus.

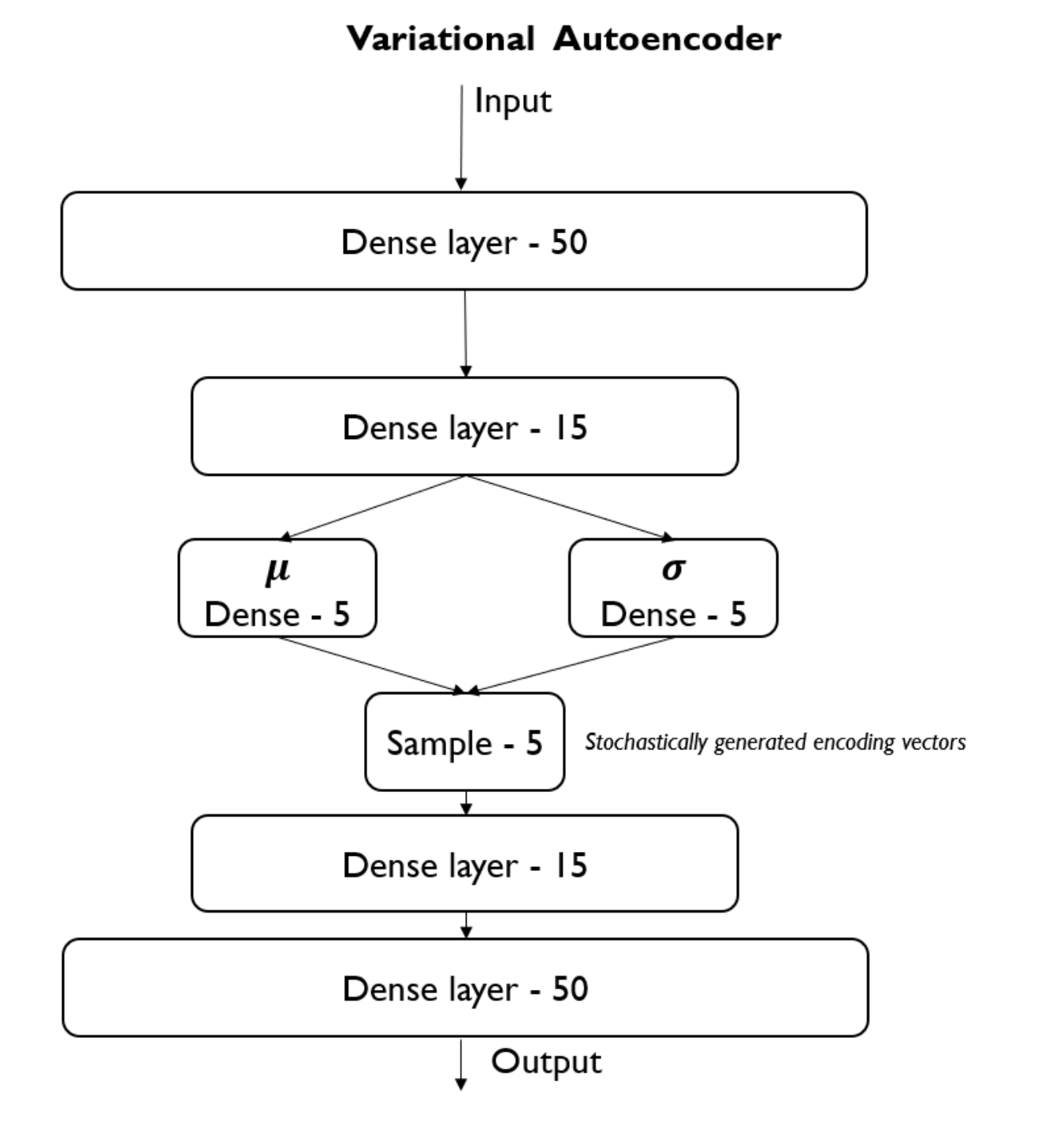
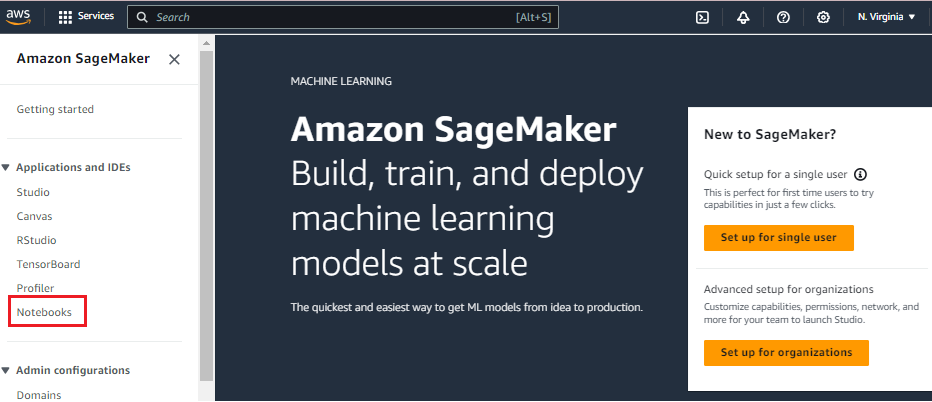


Fig. 1. The working structure of Neural Topic Model (NTM)

Note: For submission, take the screenshot for all steps and save it in your local repository along with your code.

# Setup Notebook for NTM

1. Go to your AWS account, Amazon SageMaker, and click on Notebook. Click Create notebook instance



1. Give Notebook instance name and select Notebook instance type: ml.t3.medium

Graphical user interface, application

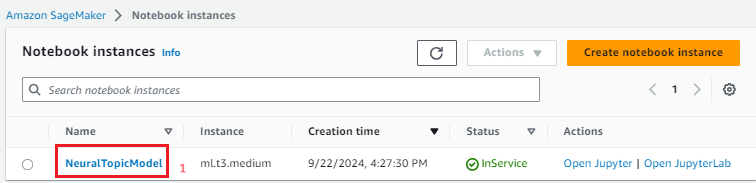
Description automatically generated

1. Follow the steps to create an IAM role. Then select the **Create notebook instance** button at the end of the page.

Graphical user interface, text, application, email

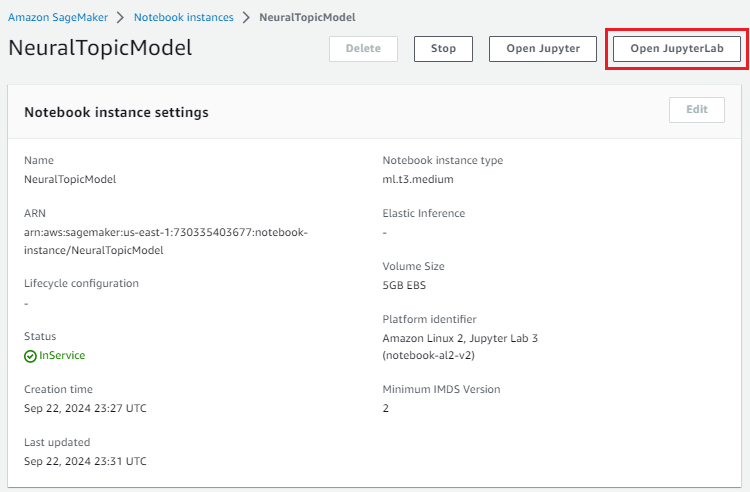
Description automatically generated

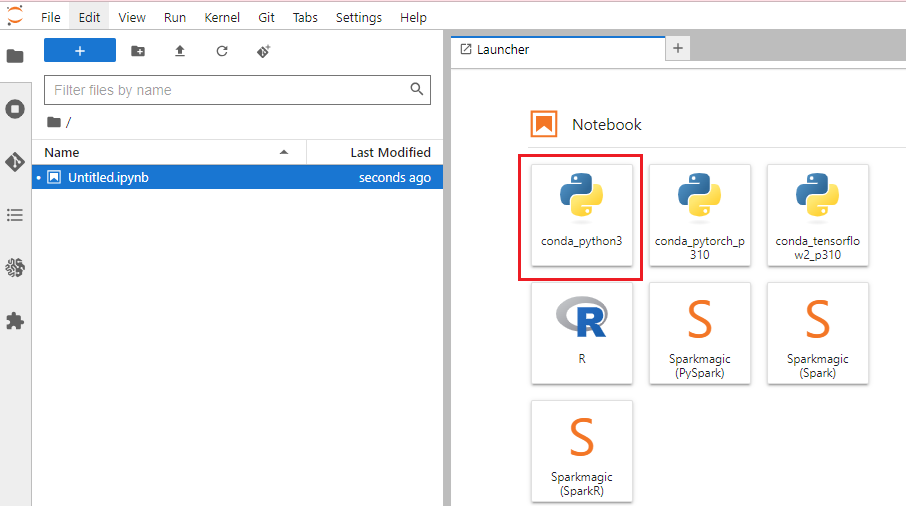
Wait until the notebook instance’s Status changes to InService. This can take a little while.



Click Open JupyterLab > conda\_python3 to create a new notebook

Your assignment repo comes with a starter notebook. You can upload this notebook to complete the assignment.





# Training NTM in SageMaker

## Fetching Data Set

First let’s define the folder to hold the data and clean the content in it which might be from previous experiments.

Graphical user interface, text

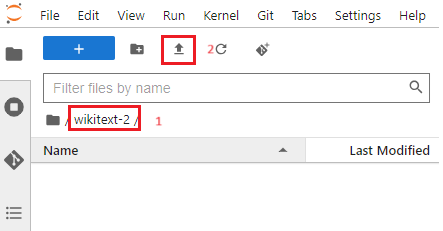
Description automatically generated

Let’s download the wikitext-2 data from [Kaggle](https://www.kaggle.com/datasets/bestwater/wikitext-2-v1) and unzip it to your local computer.

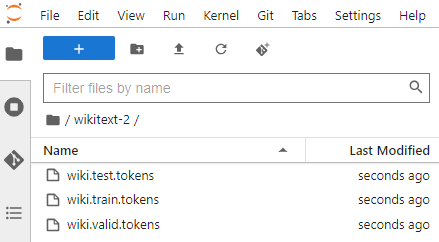
There should be 3 files:

* wiki.train.tokens
* wiki.val.tokens
* Wiki.test.tokens

Upload these files to the SageMaker notebook. On the Jupyter notebook, select the wikitext-2 folder and select the upload button. Upload all 3 files to this folder



Your wikitext-2 should look like this:



## Preprocessing

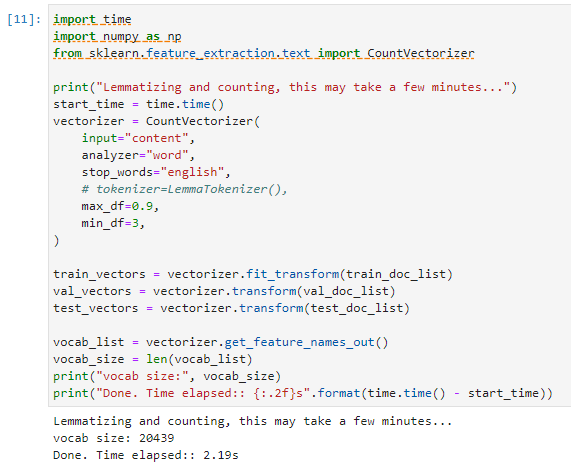
Let’s first parse the input files into separate documents. We can identify each document by its title in level-1 handling.



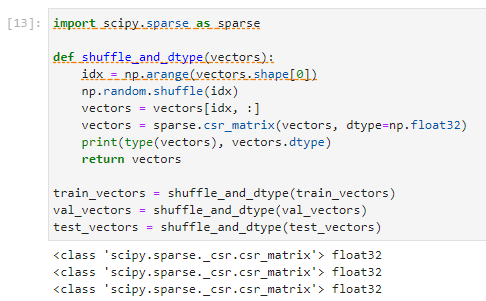
Let’s install and import nltk.



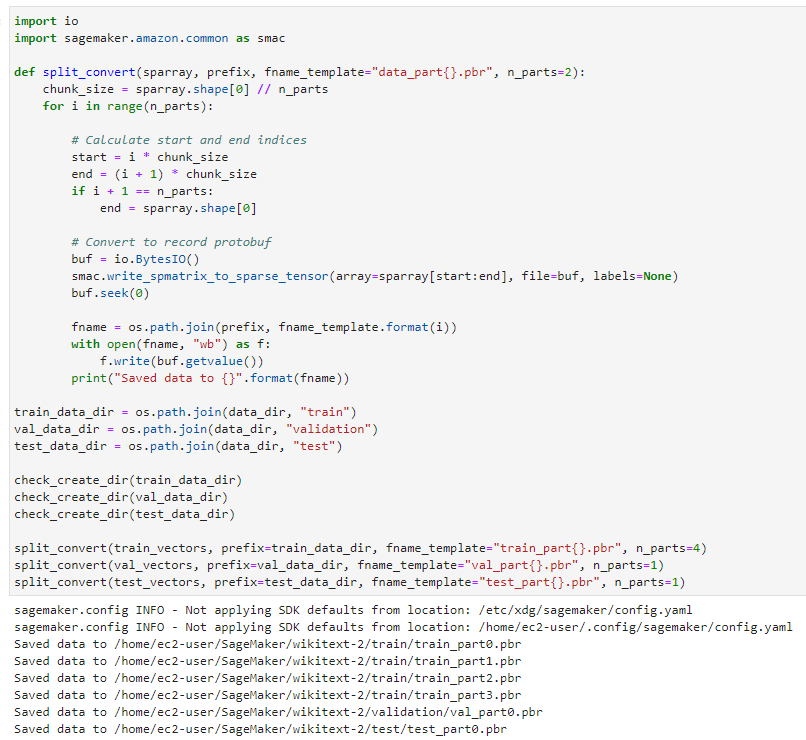
Let’s perform lemmatizing and counting next.



Define train, test, and val to train the algorithm.



The NTM algorithm accepts data in RecordIO Protobuf format. Inside this helper functions we use write\_spmatrix\_to\_sparse\_tensor function provided by SageMaker to convert scipy spare matrix into RecordIO Protobuf format.



Let’s save the text file with the name vocab.txt in the auxiliary directory.

A picture containing text

Description automatically generated

## Store Data on S3

Specify data locations and access roles. The S3 bucket and prefix that you want to use for training and model data. The IAM role is used to give training and hosting access to your data.

Text

Description automatically generated

Upload the input directories to S3

Text

Description automatically generated

Graphical user interface, text, application, email

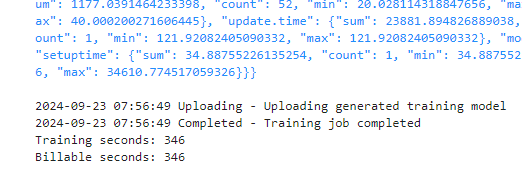
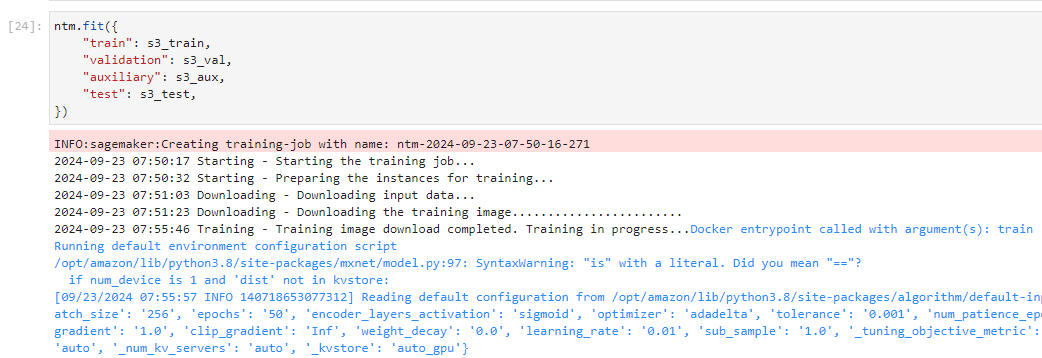
Description automatically generated

## Model Training

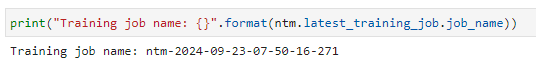
Let’s configure a SageMaker training job to use the NTM algorithm on the data we prepared. SageMaker uses Amazon Elastic Container Registry (ECR) docker container to host the NTM training image. \*\* in below screen show( instance\_type = ‘ml.c4.xlarge’)



Now, it is ready to run the training job. Again, we will notice in the log that the top words are printed together with the WETC and TU scores.



Once the job is completed, you can view information about and the status of a training job using the AWS SageMaker console.



**HOS submission instructions:**

1. Please install the GitHub Desktop: <https://cityuseattle.github.io/docs/git/github_desktop/>

2. Clone, organize, and submit your work through GitHub Desktop: <https://cityuseattle.github.io/docs/hoporhos>