Using gensim for NLP in big data

Long wen

CS 410: University of Illinois at Urbana-Champaign

November 11st, 2020

1. Introduction

The increase in social media with multiply E-Commerce transactions has resulted in the explosion of data, and time intervals that doubling the global data has been more and more shorting. Meanwhile, nearly 80%[1] of the data organized as unstructured data and most of them are picture, video, and text. In order to deal with the situation, we need more efficient tool and powerful compute platform to solve this issue.

Gensim is a very useful natural language processing (NLP) python library, which could support many NLP tasks:

a. Topic modeling.
b. Converting word to vectors
c. Converting document to vectors
d. Finding text similarity
e. Text Summarization

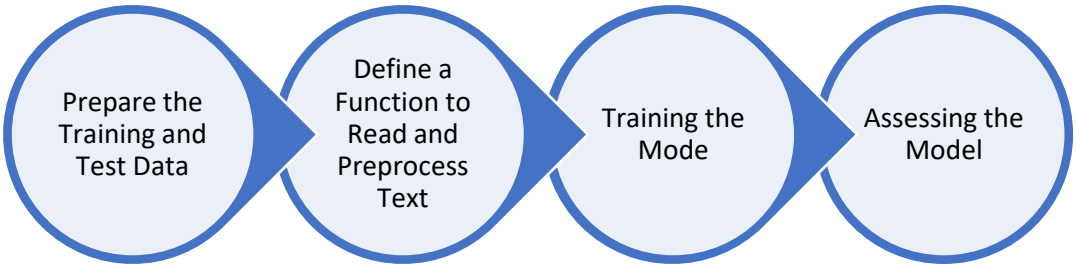
Spark is a powerful and flexible compute platform to execute the data manipulation not only in data pipeline but only interactive query, below is the advantage of spark[2].

This article focuses on use case in my project to use gensim library in spark(this use case use azure synapse which bring limitless enterprise data warehousing) to resolve the issue of data processing.

This use case need to analyze the email between customer and customer support, which is about 100G with nearly 500K emails.

2. Case of Study: Create Topic Model

In the case, we need to analyze all the email interaction transactions to generate the top topic that the customer's main pain points. Below is the step to apply the top model with genism[3].



The use case tests the performance in different configuration, the parameters include executors, executor size.

Prepare the Training and Test Data

Synapse could link to different data source, like data blob, azure data late storage, etc. with very simple script, and save to spark table with high performance. Below is sample query:

```
%%pyspark
IncidentInteraction_raw = spark.read.options(header='True')\
    .text('adl://cp-bizops-
c15.azuredatalakestore.net/local/users/lowen/AzureOps/IncidentInteraction.csv')
interaction=IncidentInteraction_raw.\
selectExpr("cast(split(value,'\t')[0] as string) IncidentInteractionId",
"cast(split(value,'\t')[1] as string) IncidentId",
"cast(split(value,'\t')[3] as string) Direction",
"cast(split(value,'\t')[13] as string) text"
)
interaction.select(['IncidentId','text']).write.mode("overwrite").saveAsTable("d
efault.Interaction")
```

Define a Function to Read and Preprocess Text

The second step is to tokenize the text into individual word, remove stop word and normalize the words, etc. so we use the simple_preprocess and doc2vec to process step, below is script:

```
import smart_open

def read_corpus(text, tokens_only=False):
    i=0
    for value in text:
        tokens = gensim.utils.simple_preprocess(str(value))
        #print(tokens)
        if tokens_only:
            yield tokens
        else:
            yield gensim.models.doc2vec.TaggedDocument(tokens, [i])
            i=i+1
```

Training the Model

The data is very big and we have about 500k document, we need to make use of the synapse platform, so set the vector_size to 200, and epochs to 50.

```
model = gensim.models.doc2vec.Doc2Vec(vector_size=200, min_count=3, epochs=50)
model.build_vocab(train_corpus)
model.train(train_corpus, total_examples=model.corpus_count, epochs=model.epochs)
```

Assessing the Model

To assessing the model, we just need to compare the inferred the training doc

```
for doc_id in range(len(train_corpus)):
    inferred_vector = model.infer_vector(train_corpus[doc_id].words)
    sims = model.dv.most_similar([inferred_vector], topn=len(model.dv))
    rank = [docid for docid, sim in sims].index(doc_id)
    ranks.append(rank)
```

3. Conclude

To compare the benefit of the spark's ability of parallel compute and memory optimized, this test case did some test on different configuration (show as below), based on the test, the genism doesn't utilize the benefit of the parallel of spark.

Parameters	Result
1 Small Executors	15 mins
20 Medium Executors	12 mins
10 Medium Executors	12 mins
1 Medium Executors	12mins

4. Reference and Resource

[1] https://datacrops.com/blogs/7-steps-extract-insights-unstructured-data/#:~:text=Unstructured%20Data%20Extraction%20The%20increase%20in%20digitization%20of,data%20to%20double%20in%20very%20short%20time%20intervals.

[2] https://www.quora.com/What-are-the-advantages-of-using-Apache-Spark

[3] https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html