Unsupervised Learning From Internet

**AIM**->TO create more precise search engine that give more relevant answer to user query.

**PROCESS**->1.User enter the query

2.Retrieve information related to the search query from specific website(STATIC) or Google Search the query and then retrieve information by scraping from the top results shown by Google search (DYNAMIC).

3.Store the scrap information in a format in neo4j database

4.Create a model/agent trained to convert query entered by user into corresponding cypher query which will extract the required (more specific) information and present to user in readable format.

**PROGRESS->**

**STEP1**

.Android smartphone database has been created from Wikipedia which contains Specification from more than 300 smartphones.

For Scraping data from various sites we used *Beautiful Soup* model .

Sample code to scrap text data from wikipedia page has been uploaded on github account <https://github.ibm.com/madhaka4/UnsupervisedLearning> with name ***wiki\_scraping.py***

It uses Google search for query written by user and for that I separately downloaded the google module as named it ***hhh.py*** and that is also uploaded on above github account.

**STEP1.2**

The next step is to setup neo4j environment to storing database and retrieving desire result/data from it using cypher queries.

Go to site <https://neo4j.com/download/> and download /install neo4j on your system .

Then I created a new project on it with database username as “neo4j” and password as” **shaktisingh**”

In order to connect to databse from python code

db = GraphDatabase(**"http://localhost:7474"**, username=**"neo4j"**, password=**"shaktisingh"**)

the following code is used .

I then crawled through all the android smartphone link given on

**"https://en.wikipedia.org/wiki/Xiaomi\_Redmi"**

And store the data in ***property : value*** format on neo4j

The code which achieve this is written in ***AndroidSmartPhone.py*** also available on above given github link.

**STEP2**->.A demo GUI has been created which retrieve query from user and based on some specific pattern extract information from database and present in required format. The code for this is given in [***Neo4jQA\_enhance.py***](https://github.ibm.com/madhaka4/UnsupervisedLearning/blob/master/Neo4jQA_enhance.py)

Here I used wamp server to display CGI created using python to take query from user in natural language and give answer.

Download and install the server from <https://sourceforge.net/projects/wampserver/>

The Files (**Neo4jQA\_enhance.py and second.py**)to be access by server are put inside ***www*** folder under the installation directory of wamp.

Type <http://localhost/second.py> after running wamp server and neo4j database ,it will show all the mobiles whose database is stored .Then enter user query in text field it will redirect to **Neo4jQA\_enhance.py script** which then connect to neo4j database and perform relative cypher query and return the result press clear button to again give some query.

**STEP3**->.A database with more than 500 pair of Natural language questions and their corresponding cypher query has been created to train a seq2seq model which will translate the user query into cypher query and retrieve the results.

The database was created by taking question from flipkart site under Question and Answer section.

The training set is stored in “.txt format” file with name Database.txt and is also uploaded on <https://github.ibm.com/madhaka4/UnsupervisedLearning> .

Then to create a seq2seq model so that more questions asked by user in natural language can be answer we trained the above training set by using sample seq2seq model.

**Seq2Seq Sample->**

1.This model have enjoyed great success in a variety of tasks such as machine translation, speech recognition, and text summarization.

2. It consists of two RNNs : An Encoder and a Decoder. The encoder takes a sequence(sentence) as input and processes one symbol(word) at each timestep. Its objective is to convert a sequence of symbols into a fixed size feature vector that encodes only the important information in the sequence while losing the unnecessary information.

3. Each hidden state influences the next hidden state and the final hidden state can be seen as the summary of the sequence. This state is called the context or thought vector, as it represents the intention of the sequence.

4.  The decoder generates another sequence, one symbol(word) at a time. Here, at each time step, the decoder is influenced by the context and the previously generated symbols.

The Sample I took was of twitter chatbot

To install this tutorial, you need to have TensorFlow installed on your system. To install TensorFlow follow these instructions-:

* This is for installing on windows
* Installing with native pip

**pip3.6 install --upgrade tensorflow**

**pip3.6 install --upgrade tensorflow-gpu**

First method is used to install tensorflow without gpu support while second one is for installing with gpu support.

* Installing with Anaconda

Go to Anaconda site to download and install Anaconda

Create and Activate Conda environment by entering following command

C:> **conda create -n tensorflow pip python=3.5**

C:> **activate tensorflow**

(tensorflow)C:> # Your prompt should change

(tensorflow)C:> **pip install --ignore-installed --upgrade tensorflow**

For any confusion go to site <https://www.tensorflow.org/install/install_windows>

The sample model we first tried was twiiter bot based on twitter chat log borrowed from Marsan-Ma

<http://suriyadeepan.github.io/2016-12-31-practical-seq2seq/>

Above is the link of the trained model

We tried above model to run but there were tensorflow version compatible issues coming like

OP\_REQUIRES failed at save\_restore\_v2\_ops.cc:184 : Not found: Key decoder/embedding\_rnn\_seq2seq/embedding\_rnn\_decoder/rnn\_decoder/output\_projection\_wrapper/bias not found in checkpoint

As the model is trained using tensorflow1.0 while I was having tensorflow 1.8

Then I used the untrained model downloaded from

<https://github.com/tensorlayer/seq2seq-chatbot>

It took 7-8 days for it to be half trained but the results were promising

Here are the several output it achieved during its 25 iteration on given training sample.

**From**

**Query**-> donald trump won last nights presidential debate according to snap online polls

**Answer**-> The the aa iii ….

**To**

**Query->** Donald trump won last nights presidential debate according to snap online polls

**Answer**-> thought he was a joke

**Query**-> happy Birthday

**Answer**-> thank you very much

Then we created our own training set with around 500 human questions translated to cypher queries and trained the data on above code

But error were there as data processing for above code doesn’t fit our requirement for cypher queries as it need to have certaing brackets and commmas to be included in it .