**How to build KNowledge Graph for search**

**What is Knowledge Graph?**

To store,represent and search real world information or fact based data knowledge graphs can be used.Knowledge graph (KG) uses the triples to describe the facts in the real world. Triples contains head entity,relation and tail entity or in simple words subject,relation and object.Knowledge graphs can be used in search engine,recommendations,chatbots.In this blog we discuss how to build knowledge graph for search.

**Steps to build knowledge Graph for search:**

1.Generating triples from given text

2.Storing Triples in Graph Database

3.Querying the graph

**Step1:Generating triples from given text:**

Extraction of triples from documents contains mainly two steps:

**1.Entity Extraction:**

Named entity extraction is a popular technique used in information extraction which extracts the entities from text based on predefined classes.Different NER systems to generate entities from given text:

* spacy NER,Stanford NER
* Pretrained models like BERT,Elmo
* Machine learning algorithms like conditional Random fields
* Deep learning algorithms like Bidirectional LSTM-CRF,LSTM-CNN

**2.Finding Relation between entities:**

Once the entities are extracted from text,next we have to find the relation between these entities . It is the task of extracting semantic relations between two or more entities .For example: Tajmahal is in Agra .Here **Tajmahal** is head entity,**is in** is relation ,**Agra** is Tail Entity.This can be represented as triple(Tajmahal,is in,Agra)

There are different methods for doing Relation Extraction,few of them are

* Rule-based RE
* Supervised RE
* Unsupervised RE

alternatively we can use below mentioned tools to extract triples from documents

1.open source python module stanford-openie <https://pypi.org/project/stanford-openie/>

2.IBM Watson NLU service <https://developer.ibm.com/patterns/build-a-domain-specific-knowledge-graph-from-given-set-of-documents/>

In this article i'm going to explain how to generate triples using stanford-openie python module:

Open information extraction (openie) was released by the stanford natural processing group.This software is a Java implementation of an open IE system.

**Installation Steps:**

1.Since it is written in Java, make sure java 1.8 is installed in your system .To check whether java is installed or not run below command in terminal

**java -version**

2.If OS is windows system ,download and install the java from <https://www.oracle.com/in/java/technologies/javase-downloads.html> and set the path

3.If OS is linux system,install the java using below commands

**sudo add-apt-repository ppa:webupd8team/java**

**sudo apt-get update**

**sudo apt-get install oracle-java8-installer**

4.Install stanford openie python package using below command:

**pip install stanford-openie**

**pip install stanford-openiepip install stanford-openie**

The openie is supporting only 100000 characters,so the length of the text must be below 100000 characters ,you can know the length of the text by using below python syntax

**print(len(text))**

5.**Execution:**Run the below code by passing input text and file name of csv to store triples

|  |
| --- |
| from openie import StanfordOpenIE  import csv  def triple\_generation(text,file\_name):  subject=[]  relation=[]  objects=[]  with StanfordOpenIE() as client:  for triple in client.annotate(text):  subject.append(triple['subject'])  relation.append(triple['relation'])  objects.append(triple['object'])  with open(file\_name, 'w') as f:  writer = csv.writer(f)  writer.writerows(zip(subject,relation,objects))  return "Triples Generated and stored in given csv file name"  ## sample test case  print(triple\_generation ("Barack Obama was born in Hawaii.","output.csv")) |

**Step2:Storing triples in Graph Database:**

Graph databases are designed to store nodes and their relations(edges).Graph databases give first priority to relationships.Unlike other database management systems, in graph databases connected data is equally important to individual data.

**Applications of Graph databases:**

* Knowledge Graph
* Recommendation Engine
* Fraud Detection

**Available graph databases are:**

* ArangoDB
* Neo4j
* OrientDB
* Amazon Neptune
* FlockDB
* DataStax
* Cassandra
* Cayley

Now lets see how to store generated triples triples in Neo4j

Neo4j is available in two editions:

* **Community edition**:It is fully functional and designed for single instance deployments .
* **Enterprise edition:**It has all the features of community edition and has extra features like clustering and online backup facility .

Installation steps for community edition:

* In windows, download neo4j from <https://neo4j.com/download/neo4j-desktop/?edition=desktop&flavour=unix&release=1.2.9&offline=true> and install it
* In ubuntu, java has to be installed to install neo4j.Follow the steps in below link to install neo4j in ubuntu <https://dzone.com/articles/installing-neo4j-on-ubuntu-1604>

Once the installation is completed neo4j can be accessible in the browser also,the default username and password for the browser version is neo4j.

Install py2neo package using below command:

**pip install py2neo**

|  |
| --- |
| from py2neo import Graph, Node, Relationship  import pandas as pd  #give the ip address and port on which neo4j is running  # auth =(user\_name,password), both neo4j in this case  graph = Graph("http://localhost:7474", auth=("neo4j", "neo4j"))  df = pd.read\_csv("output.csv")  ##The header of the csv file must be subject ,relation,object  for index, row in triples\_df.iterrows():  tx = graph.begin()  a = Node('Subject',name = row['subject'])  tx.create(a)  b = Node('Object',name = row['object'])  ab = Relationship(a,row['relation'],b)  tx.create(ab)  tx.commit() |

Now give the output.csv (which is generated from the previous step) as input to the above code to store triples in neo4j Database.

Now you can visualise knowledge graph in neo4j database

**Step3: Querying neo4j or knowledge Graph:**

Go to the neo4j web browser,there you can start searching nodes

Sample query to start with:

**MATCH (n:Subject) RETURN n LIMIT 25**

- Matches the nodes with label Subject and returns 25 results as a limitation is mentioned in the query.

- You can select the label from Subject and Object and query the graph using the query.

or you can access the neo4j graph with credentials using python script.

**Conclusion:**

We are using knowledge graphs in our daily life without knowing it.For example we use google search daily but many people don't know that it is using knowledge graph in backend.Many advanced concepts are there to build a knowledge graph using artificial intelligence,In this blog i have explained a basic way to generate a knowledge graph using stanford openie and neo4j with python .Further ,we can predict the missing relations between the entities using link prediction or relation prediction concepts.