

# **Automatic Concept Map Generation from Text-Based Learning Material**

**Documentation**

**Submitted By**

**Ashish Malgawa 17CS60R81**

**Hussain Jagirdar 17CS60R83**

**Abhishek Prakash Singh 17CS60R48**

**Shah Smit Ketankumar 17CS60R72**

**Guided by**

**Prof. Plaban Kumar Bhowmick**

**Master of Technology**

**In**

**Computer Science**



**Department of Centre for Educational Technology**

**Indian Institute of Technology, Kharagpur**

**April 2018**

# Contents

- **Prerequisites**

- Python
- Java 8
- NLTK
- Scipy
- ElementTree

- **Downloads**

- Glove Dataset
- Solr 7.1.0
- Pysolr
- Pyspotlight
- Gephi
- Core NLP Package

- **Installation Manual**

- Solr Indexing
- Coreference Resolution and OpenIE
- Gephi Installation

- **User manual**

- **Evaluation Metrics**

# Downloads

## 1. Glove Dataset

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

Pre-trained word vectors.

**Download Link :** <http://nlp.stanford.edu/data/glove.6B.zip>

This data is made available under the Public Domain Dedication and License .

(For convenience extract this zip file under data directory)

## 2. Solr 7.1.0

Solr is the popular, blazing fast open source enterprise search platform from the Apache Lucene project. Its major features include powerful full-text search, hit highlighting, faceted search, dynamic clustering, database integration, rich document (e.g., Word, PDF) handling, and geospatial search. Solr is highly scalable, providing distributed search and index replication, and it powers the search and navigation features of many of the world's largest internet sites.

**Download Link :** <http://archive.apache.org/dist/lucene/solr/7.1.0/>

## 3. Pysolr

Pysolr is a lightweight Python wrapper for Apache Solr. It provides an interface that queries the server and returns results based on the query.

**Install:** \$ pip install pysolr

## 4. Pyspotlight

Pyspotlight is a thin python wrapper around DBpedia Spotlight .

**Install:** \$ pip install pyspotlight

## 5. Gephi

Gephi is an open-source network analysis and visualization software package written in Java on the NetBeans platform.

**Download Link :** <https://gephi.org/users/download/>

## **6. Core NLP Package**

Stanford CoreNLP provides a set of human language technology tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between entity mentions, get the quotes people said, etc.

**Download Link :**

***<http://nlp.stanford.edu/software/stanford-corenlp-full-2018-02-27.zip>***

# Installation Manual

## Solr Indexing

### Steps:

1. Follow this guide to install solr:

*[https://lucene.apache.org/solr/guide/7\\_1/installing-solr.html](https://lucene.apache.org/solr/guide/7_1/installing-solr.html)*

2. Open the folder in which you've installed solr.

3. Start solr

`>bin/solr start`

The line below indicates you've successfully installed solr

*Waiting up to 180 seconds to see Solr running on port 8983 [\\]*

*Started Solr server on port 8983 (pid=24451). Happy searching!*

4. Create a core named glove.

`> bin/solr create_core -c glove`

This will come in form of a success message

*WARNING: Using \_default configset. Data driven schema functionality is enabled by default, which is*

*NOT RECOMMENDED for production use.*

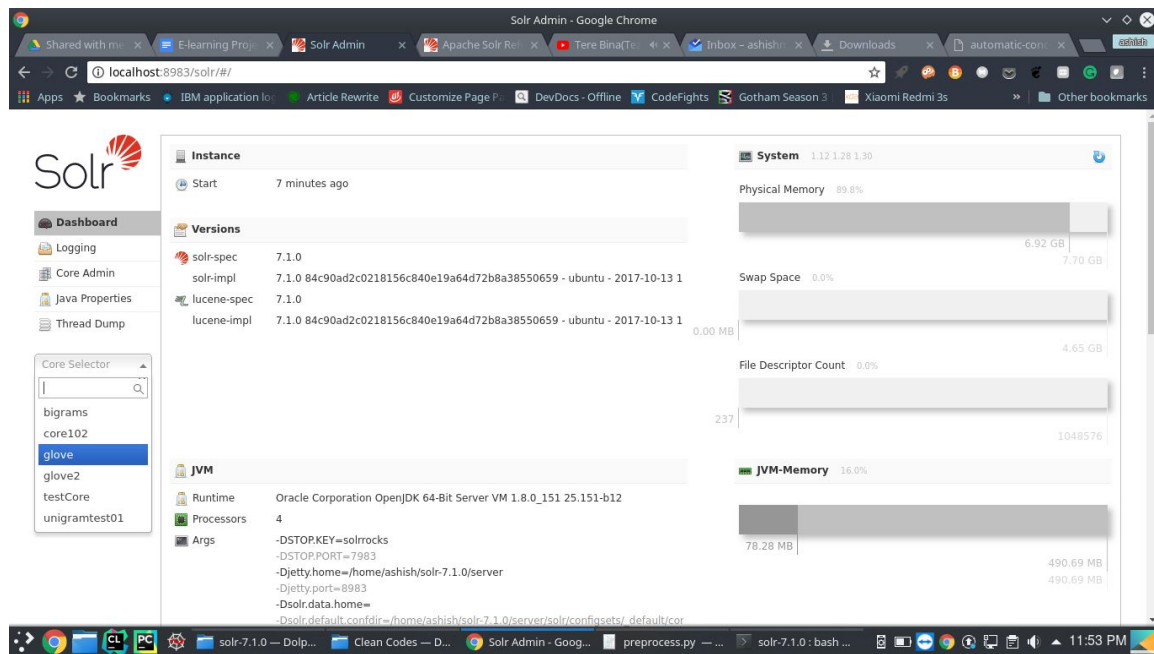
*To turn it off:*

*`curl http://localhost:8983/solr/glove2/config -d '{"set-user-property": {"update.autoCreateFields": "false"}}'`*

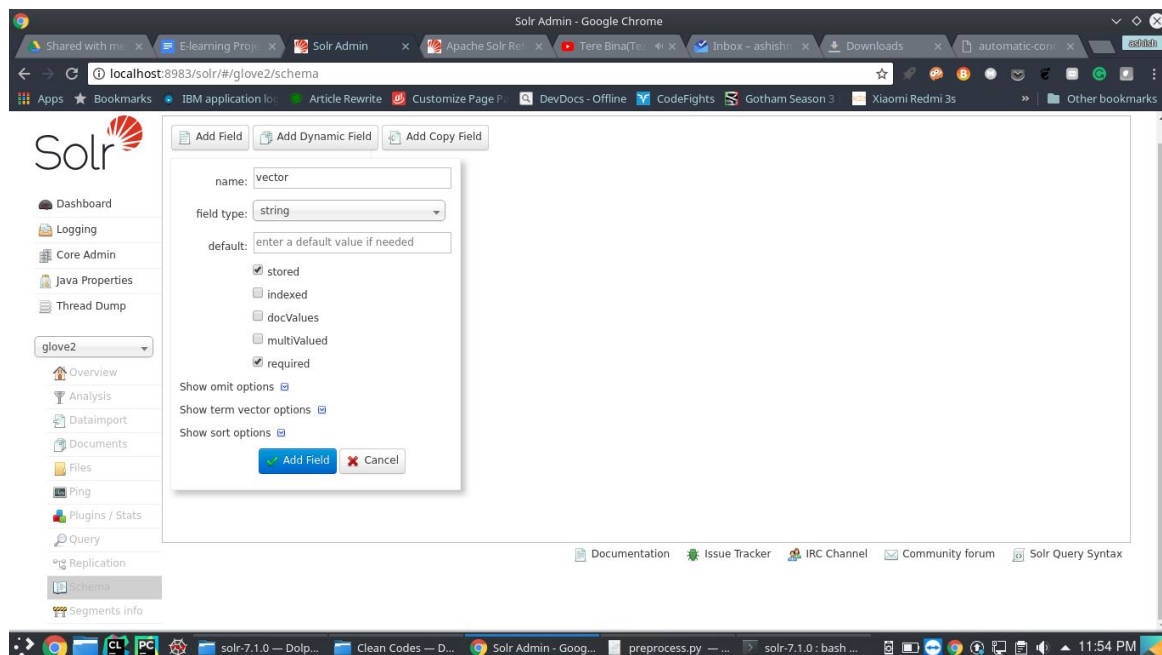
*Created new core 'glove'*

5. Open solr admin panel.

- a. Go to this address localhost:PORT\_NUMBER\_OF\_SOLR/solr/



6. Add a string named “vector” field under the schema section.



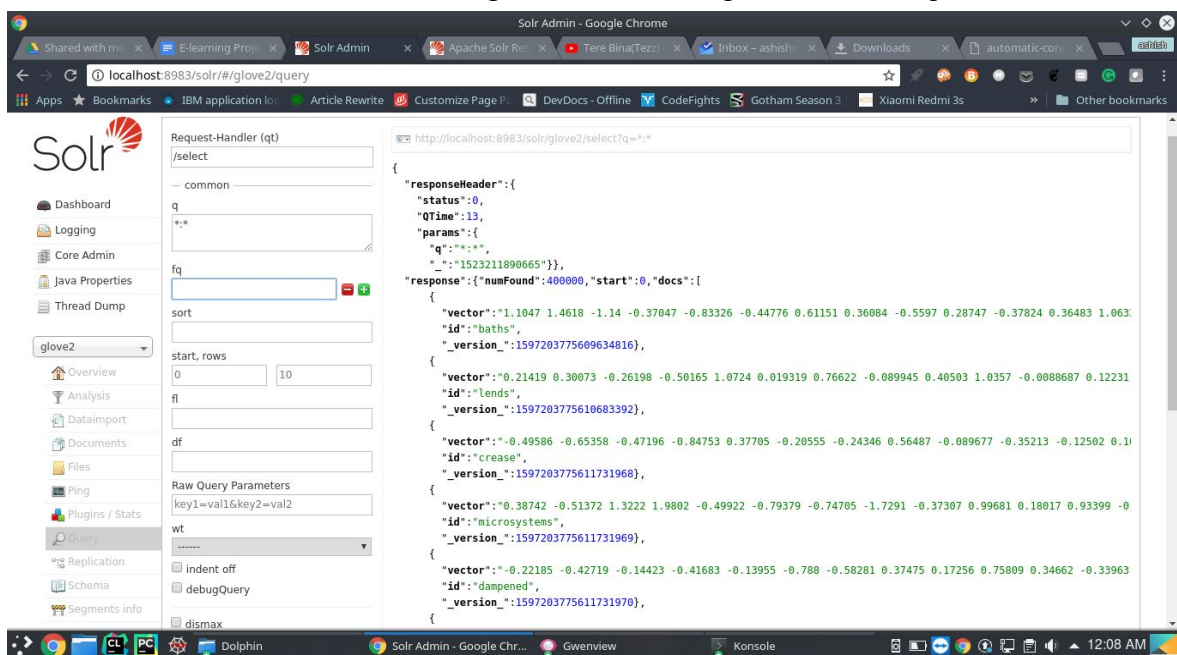
7. Install pysolr
  - > pip install pysolr
8. Run solr\_indexer.py with following format
 

```
python solr_indexer.py path_to_glove_dataset PORT_NO CORE_NAME
```

 Eg. : python solr\_indexer.py ../data/glove.6B.50d.txt 8983 glove2
 On success:

```
Pre-trained_word_vectors: bash — Konsole <2>
File Edit View Bookmarks Settings Help
ashish@ashish-Inspiron-3543: /media/ashish/Studies/IIT Kgp/Assignments/NLP/Project/Pre-trained_word_vectors$ python solr_indexer.py glove.6B.50d.txt 8983 glove2
10000 processed
20000 processed
30000 processed
40000 processed
50000 processed
60000 processed
70000 processed
80000 processed
90000 processed
100000 processed
110000 processed
120000 processed
130000 processed
140000 processed
150000 processed
160000 processed
170000 processed
180000 processed
190000 processed
200000 processed
210000 processed
220000 processed
230000 processed
240000 processed
250000 processed
260000 processed
270000 processed
280000 processed
290000 processed
300000 processed
310000 processed
320000 processed
330000 processed
340000 processed
350000 processed
360000 processed
370000 processed
380000 processed
390000 processed
400000 processed
ashish@ashish-Inspiron-3543: /media/ashish/Studies/IIT Kgp/Assignments/NLP/Project/Pre-trained_word_vectors$
```

To further check the indexed data set go solr admin and press enter on fq



```
Request-Handler (qt)
/select
common
q
fq
sort
start, rows
0 10
fl
df
Raw Query Parameters
key1=val1&key2=val2
wt
indent off
debugQuery
dismax

http://localhost:8983/solr/glove2/select?q=*:*
{
  "responseHeader": {
    "status": 0,
    "QTime": 13,
    "params": {
      "q": "*",
      "fq": "glove2"
    }
  },
  "response": {
    "numFound": 400000, "start": 0, "docs": [
      {
        "vector": "1.1047 1.4618 -1.14 -0.37047 -0.83326 -0.44776 0.61151 0.36084 -0.5597 0.28747 -0.37824 0.36483 1.063",
        "id": "baths",
        "_version_": "1597203775609634816",
      },
      {
        "vector": "-0.21419 0.38073 -0.26198 -0.50165 1.0724 0.019319 0.76622 -0.089945 0.40503 1.0357 -0.0088687 0.12231",
        "id": "lends",
        "_version_": "1597203775610683392",
      },
      {
        "vector": "-0.49586 -0.65358 -0.47196 -0.84753 0.37705 -0.20555 -0.24346 0.56487 -0.089677 -0.35213 -0.12502 0.1",
        "id": "crease",
        "_version_": "1597203775611731968",
      },
      {
        "vector": "-0.38742 -0.51372 1.3222 1.9802 -0.49922 -0.79379 -0.74705 -1.7291 -0.37307 0.99681 0.18017 0.93399 -0",
        "id": "microsystems",
        "_version_": "1597203775611731969",
      },
      {
        "vector": "-0.22185 -0.42719 -0.14423 -0.41683 -0.13955 -0.788 -0.58281 0.37475 0.17256 0.75809 0.34662 -0.33963",
        "id": "dampened",
        "_version_": "1597203775611731970",
      },
    ]
  }
}
```

9. Indexing Complete. Happy Searching!

## Coreference Resolution and OpenIE

Steps -

1. Download Stanford CoreNLP package from here.

Or use the following commands:

```
wget http://nlp.stanford.edu/software/stanford-corenlp-full-2018-02-27.zip
```

Or get using curl:

```
curl -O http://nlp.stanford.edu/software/stanford-corenlp-full-2018-02-27.zip
```

2. Unzip the release:

```
unzip stanford-corenlp-full-2018-02-27.zip
```

Store the unzipped file in **data** directory of the project.

3. Enter the newly unzipped directory:

```
cd stanford-corenlp-full-2018-02-27
```

4. Set up your classpath. If you're using an IDE, you should set the classpath in your IDE. If you are using bash or a bash-like shell, the following will work.

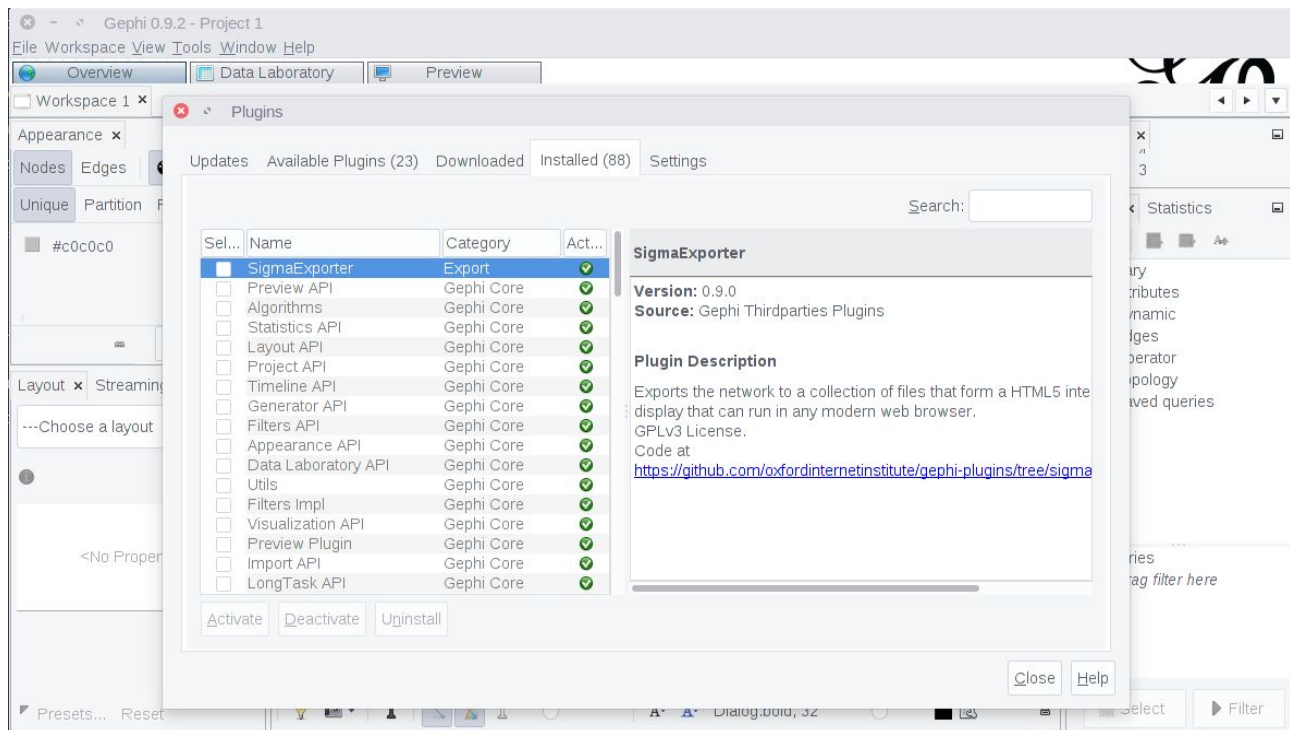
```
for file in `find . -name "*.jar"`; do export  
CLASSPATH="$CLASSPATH:`realpath $file`"; done
```



# Gephi Installation

Steps:

1. Follow this guide to install Gephi <https://gephi.org/users/quick-start/>
2. Unzip the folder.
3. Go to the bin folder and run './gephi' command to launch the gephi tool. (for ubuntu)



4. Go to tools menu and select Plugins.
5. Download Graph Streamer Plugin.

# User Manual

- All code should be present in src folder, data in data folder and output in output folder respectively.
- Run makefile.sh with version number (1.0, 2.0, 2.1, 2.2) as first argument and filename (path : data/<filename> ) as second argument.
- Now output would be a gephi file, stored in the output folder.
- Open that gephi file in Gephi tool.
- To display the labels, go to the data laboratory panel (edges) and copy 'hase' field to 'Label' field.
- For other stuff like animation and changing the UI, you can refer the quick guide link. <https://gephi.org/users/quick-start/>
- Algorithms are explained in the presentation with results.

## Evaluation metrics

1. We considered time taken to create a concept map considering number of words in the document. (To check the scalability of the model)
2. **Precision:** We can have a standard hand drawn concept map, and we can compare how close our concept map is from the standard one.
3. **Concept Density:** We can also calculate the concept density for different documents with same length.