



Automatic Concept Map Generation from Text-based Learning Material

Under the Guidance of:
Prof. P.K. Bhowmick

Presented by:
Suyog Chadawar 17CS60R57
Ashwin Bhoyar 17CS60R35
Akshay Patil 17CS60R53



Problem Statement

Given a learning material generate visual representation of concept map.

- Identify important concepts in given text file.
- Extract relations among the identified concepts.
- Determine weights of the extracted relations.
- Present a visual representation of extracted concept map.



Modules

Preprocessing:

- By using StanfordCoreNLP coreference resolution has been done.
- This clean data is sentence tokenized using NLTK sentence tokenizer.

Concept Extraction:

- In this step we find the important concepts in the given text by using Dbpedia spotlight.
- Dbpedia spotlight API of python is time consuming.
- That's why, we have used threading to find concepts in different sentences parallelly.



Modules

Extract Relations:

- The sentences obtained from preprocessing step is feed to OpenIE tool.
- The output of OpenIE is *<subject, predicate, object>*

```
Total Relations found 35
(u'Wind', u'has', u'turn')
(u'Sun', u'has', u'smile')
(u'you', u'are', u'divided')
(u'thief', u'is in', u'house')
(u'boy', u'Now shouted', u'Lion')
(u'faith', u'is in', u'god')
(u'one', u'should always do things on', u'time')
(u'pond', u'full of', u'fishes')
(u'reciting', u'name of', u'god')
(u'foxes', u'are', u'cunning')
(u'LORD GANESHA', u'OF STORY is', u'Goddess Parvathi')
(u'men', u'finally came upon', u'elephant')
(u'It', u'lift higher', u'it')
(u'It', u'lift', u'it')
(u'We', u'can soar above', u'storm')
(u'God', u'has', u'power')
(u'hole', u'is in', u'ground')
(u'accident', u'life of', u'child')
(u'They', u'would toss', u'coin')
(u'Albrecht', u'has', u'etchings')
(u'bones', u'is in', u'finger')
```



Modules

Concept Similarity:

- Pretrained GloVe word vectors are indexed in Solr.
- Cosine similarity is calculated between *<subject, object>*

Filtering Unrelated Pairs:

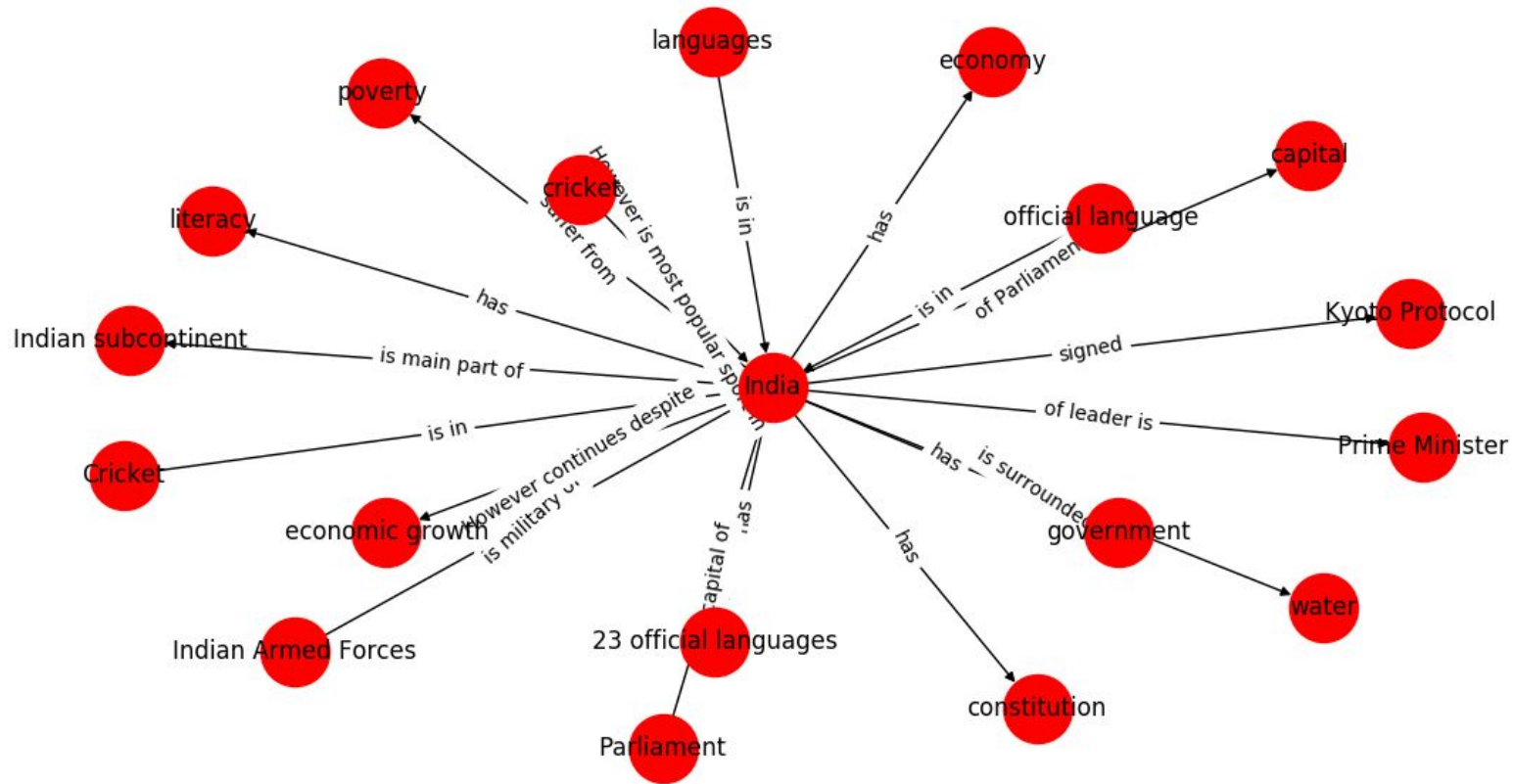
- In this step the *<subject, object>* pair having cosine similarity less than the **threshold** value is filtered out.

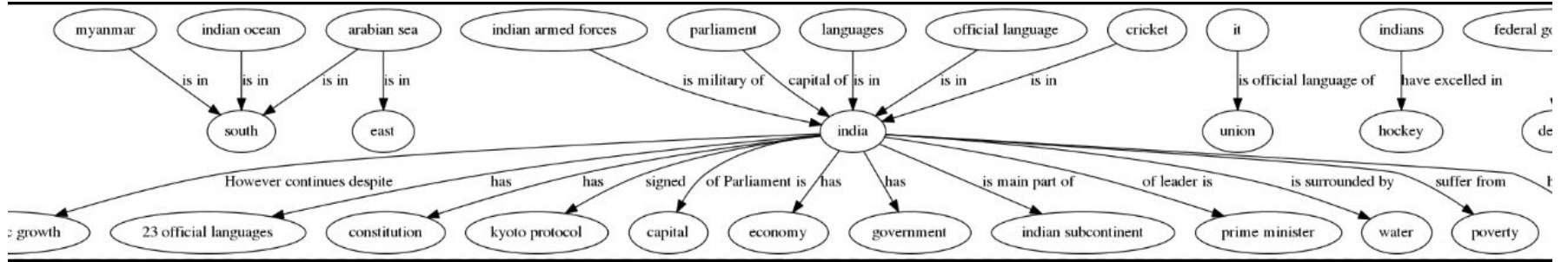


Modules

Visualization Interface:

- The filtered tuples *<subject, predicate, object>* are visualised.
- For visualisation we have used NetworkX with Matplotlib.
- We have also visualised the concept graph using Graphviz.







Evaluation of the output

- Input data is given to the human annotator and is asked to evaluate *<subject, predicate, object>* tuples.
- Given this tuples and the tuples from our program we can calculate the precision and recall.



Thank you