Automatic Concept Map Generation from Text-based Learning Material

Plaban Kumar Bhowmick

March 8, 2018

1 Introduction

Concept map is an effective tool to present summary of a learning material. Concept map is represented by a set of concepts and their relationships. They are different from formal ontology in the sense that concepts maps do not consider standard vocabulary to represent the relations. The concepts and relations are formed using free text. An example concept map for *evolution* topic in biology is presented in Fig 1.

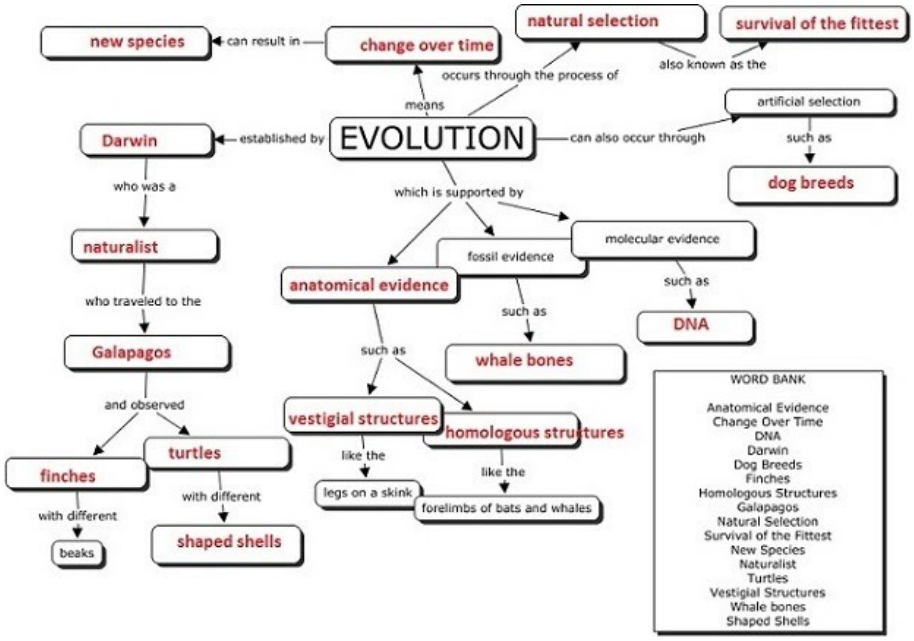

Figure 1: A concept map for evolution topic in Biology

Concept map for a given learning material is not unique. A concept map is represented as a graph $G = (V, E)$ where V is the set of vertices representing different concepts and E is the set of edges representing the relations. An edge $e \in E$ can be represented as a tuple $\rangle l, w \langle$ where l is a string that represents the relation name and $w \in \mathbb{R}$ represents the strength of the relation.

2 Problem Statement

Given a learning material (from varying domains)

- identify important concepts that are discussed in the LM
- extract relations among the identified concepts
- determine weights of the extracted relations
- present a visual representation of extracted concept map

3 Representative Steps and Tools

The project can be developed using the following steps. However, you may ignore the steps fully and partially and come up with your own methodology given that the desired outcome is obtained.

1. **Concept Extraction:** This can be done in an unsupervised way by spotting the keywords based measures like tf-idf and likes. However, you may directly use Dbpedia spotlight (Python API[1]). This API will help you to mark the important concepts in a text and link it to appropriate Dbpedia entity (e.g., Darwin[2]) You may also consider the noun phrases as candidate concepts.

2. **Concept Similarity:** Compute concept pair-wise similarity. To find similarity, represent each word as a vector. Download pre-trained word vectors (Wikipedia2014+Gigaword 5)[4] from GloVe and either index it into Solr or hash index. Similarity between a pair of concepts is distance (like cosine similarity) between the respective word vectors. A concept may involve multi-word expression. In that case, apply sum or averaging over the individual vectors to obtain the vector for multi-word concept.

3. **Filtering Unrelated Pairs:** Threshold over the similarity values to keep relations that associates concepts having higher similarity value. You may also use proximity threshold to filter out pairs that are distant apart in document. Constraint such as the concepts have to in the same sentence in order to be related.

4. **Extract Relations:** Use OpenIE tool to parse the sentences involving a pair and extract <subject, predicate, object> triple. For example. OpenIE output of "Barack Obama was born in Hawaii." would be (Barack Obama; was born in; Hawaii). Use the predicate name as relation.

5. **Computation of Relation Strength:** The relation strength might be function of similarity between two associated concepts.

6. **Visualization Interface:** Finally, the generated concept graph has to be stored against the input document and can be visualized. There are several algorithms for graph/network visualization. GephiStreamer[5], a python library for Gephi, one of the renowned graph analysis tools, is one of them.

4 Dataset

NCERT books in Social studies and Science for lower grade levels (VI-VII). You might also use khan academy videos for lower grades and have transcripts.

5 Delivery Mode

This will be a group activity. Form a group of four and distribute the work among the group members. Different forms of improvisations in regard to the presented steps are highly solicited and will be rewarded.

Each group has to make a presentation and demonstration of the work.The tentative presentation schedule is 07.04.2018.

[1] pyspotlight: https://pypi.python.org/pypi/pyspotlight/0.7.1
[2] Dbpedia Entry of Charles Darwin: [3]
[4] Pre-trained word vectors: http://nlp.stanford.edu/data/glove.6B.zip
[5] https://pypi.python.org/pypi/GephiStreamer