Constructing knowledge graph from unstructured text

Kundan Kumar
Siddhant Manocha

CS365A: ARTIFICIAL INTELLIGENCE

Instructor: Dr. Amitabh Mukherjee

1 Abstract

The objective of constructing a knowledge graph is to construct structured graphical representation of semantic knowledge and relations extracted from unstructured text. In this project, we aim to extract concepts, entities and relationships from unstructured text and populate the knowledge graph. The resulting knowledge graph can be combined with existing knowledge bases as freebase, YAGO, DBpedia etc. to make efficient question answering systems. We further aim to investigate the effectiveness of word vector embeddings in capturing syntactic and semantic regularities in English language and further find a mapping to capture the corresponding relations in Hindi language.

2 Motivation

The world wide web is a vast repository of knowledge, but extracting information in a structured machine readable format is still a very difficult task. Obtaining formal structured knowledge from data repositories is both difficult and expensive [5]. Unfortunately, much of human knowledge still exist in unstructured form. Last decade has witnessed significant progress in the field of information retrieval and information extraction, including Never Ending Language Learning (NELL) project, OpenIE, YAGO, and efforts at Google. These projects have proposed various methodologies to extract new structured information from the web, which represent the various entities and the relationships rather than normal strings. The concepts extracted are interlinked and are thus known as knowledge graphs. Motivated by these ongoing projects, we aim to explore various Natural Language Processing and Machine Learning techniques approaches for language modelling and extract meaningful semantic information from unstructured text.

3 Existing Knowledge Bases

- Freebase: "Freebase is a large collaborative knowledge base consisting of structured data harvested from many sources, including individual, user-submitted wiki contributions." [Source:Wikipedia]
- YAGO: YAGO is a huge semantic knowledge base, derived from Wikipedia WordNet and other sources
- Dbpedia : "DBpedia is a project aiming to extract structured content from the information created as part of the Wikipedia project.It enables users to semantically query relationships and properties associated with Wikipedia resources, including links to other related datasets"[Source:Wikipedia]

4 Related Work

Never Ending Language Learning (NELL) project at CMU had done signifcant work in learning relations using some seed examples. [?]. Initially, they specify ontology of the domain like, country has a capital city,etc. Then, they provide some seed examples for each of the ontology relations. Then, unstructured web is scraped and the instances of the specified entities and relations are found out using seed examples. They use multi-view learning paradigm, where they use a network of multiple independent hypotheses for learning ontology relations. A similar model can be learnt for Hindi Language. Work on semantic relation extraction with dependency parser and other linguistics techniques have been done at Stanford University. Apart from language dependent parsing techniques, there have been various approaches to capture semantic relations via distributed word representations. As noted in [2], semantic relations exist and can be precisely learned using distributed word representation. This approach can be utilised to learn word embeddings on english and hindi text corpus having similar context and learn a mapping between them. This can help to verify if the semantic relations captured via English word vector representation is similar to corresponding Hindi relations.

5 Methodology

There are two major steps involved in the journey from an unstructured text document to a structured knowledge graph.

1. Entities and Relations Extraction
2. Combining relations to construct a knowledge graph

5.1 Entities and Relations Extraction

Two broad category of entity and relation extraction algorithms are present in literature:

- Techniques based of linguistics concepts such as dependency parser and Named Enitity recognition [3] [4]
- "Self-supervised learner": A classifier based approach where we label possible entity-relation tuples as trustworthy or not. An extractor employing rules based on POS (parts of speech) tagging is used to extract tuples from the sentences. The set of labels classified as trustworthy are finally retained. This system can be further improved by imposing certain syntactic and lexical constraints. [4] [1]

5.2 Combining relations to construct a knowledge graph

We investigate the semantic and syntactic regularities of the word vector embedding space and aim to learn relationship among entities other than those learned through the previous approach.We apply k-means clustering in the space to learn distinct class of entities in the space.Each relationship is characterized by a relation-specific vector offset. As a result, we obtain relationship for the entity of the same class using the corresponding vector offset.

We further construct a knowledge graph where the entities serve as nodes and the relationships between entities serve as the edges. Meta data can be added to entities and relationships using existing knowledge bases as YAGO and Freebase. The entire graph is stored in Cayley, an open source graph database which can be used for efficiently querying relevant information.

References

[1] *TextRunner: Open Information Extraction on the Web*, Rochester, New York, USA, April 2007. Association for Computational Linguistics.

[2] *Linguistic Regularities in Continuous Space Word Representations.*, 2013.

[3] *The Stanford CoreNLP Natural Language Processing Toolkit*, 2014.

[4] Association for Computational Linguistics. *Identifying relations for open information extraction*, 2011.

[5] J. Fan, A. Kalyanpur, D. C. Gondek, and D. A. Ferrucci. Automatic knowledge extraction from documents. *IBM Journal of Research and Development*, 56(3):5, 2012.