Unsupervised Relation Extraction from Web

Bhavishya Mittal[1], Vempati Anurag Sai[2]
Advisor: Dr. Amitabha Mukerjee[1]
{bhavishy,vanurag,amit}@iitk.ac.in
[1] *Dept. Of Computer Science and Engineering,IIT Kanpur*
[2] *Dept. Of Electrical Engineering,IIT Kanpur*

October 22, 2013

1 Motivation

Information Extraction is a task of extracting structured information from heterogeneous text available on the web. This structured data can be used by the machines for understanding the natural language and making logical inferences from the data. The applications of such a system include populating entries in a knowledge base, document summarization and question answer systems. Since the data can involve complex relations, to make the task simpler most of the research involves identifying relation tuples like BornOn(Sachin, 24-04-1973), IsA(IBM, company) etc. Traditionally, such tasks extensively relies on human-supervision in terms of annotated training examples, hand-crafted rules, fixed set of prespecified relations [1]. The method we will be implementing shall rely on unannotated data and no fixed set of relations. Such a framework can handle heterogeneity in query language and will be domain independent.

2 Problem Statement

Extracting relation tuples from an unstructured corpus that is effective at noise removal. During the query process, given a partially filled tuple, our system will search for possible entries for the missing fields and rank the resulting tuples based on a probabilistic measure.

3 Our Approach

This framework includes three different blocks:

- Learning module
- Relation Extractor
- Evaluator

The learning module uses a dependency parser to identify Noun phrases (NP), Verb phrases (VP) and the interdependency. A set of tuples of the form (e_i,r_{ij},e_j) are built where e_i, e_j are noun phrases and r_{ij} is the verb phrases connecting them. A tuple is considered "trustworthy" if it satisfies some set of rules like,"dependency chain connecting e_i and e_j consists r_{ij} and is no longer than few cycles", "Neither e_i nor e_j is a pronoun" etc. Now for each tuple, a set of features are extracted. A classifier is learned on all such feature vectors that can evaluate tuple's

trustworthiness.

The extractor module traverses the entire corpus. For each sentence, it uses noun phrase chunker to extract candidate entities and the text in-between (now this makes it language dependent. Not valid for languages like hindi) them as a relation. The phrases are normalized, as in, intensifiers like many, definitely etc. are removed. If these tuples are classified as trustworthy by the classifier learned previously, they are retained.

The evaluator module runs over all the tuples and merges the ones with similar entity and relation fields. The count (occurences) of each tuple is used to assign a probabilistic measure to it. During the query, the missing filed is filled by the entity that maximizes the resulting tuple's probability.

4 Improvisations over previous methods

Identify synonyms among the entities and relations while grouping the tuples. So, relations like "assassinated by", "killed by", "massacred by" etc. get grouped together.If time permits, add location and temporal entities to the tuple as well.

5 Evaluation

Since it extracts totally new relations from the web, there is no gold set of correct instances of relations.Instead, we can approximate precision(can't calculate recall)

- Draw random sample of relations from output, check precision manually

$$P = \frac{\text{Number of correctly extracted relations in the sample}}{\text{Total number of extracted relations in the sample}}$$

6 Dataset

We will be using articles from *Wikipedia* as our resource. *Wordnet* will also be used to group certain relations under one category.

References

[1] Banko, Michele, et al. "Open Information Extraction from the Web." IJCAI. Vol. 7. 2007.

[2] Fader, Anthony, Stephen Soderland, and Oren Etzioni. "Identifying relations for open information extraction." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.