Using Markov Logic to Refine an Automatically Extracted Knowledge Base

Introduction

Motivation: A number of information extraction (IE) projects such as NELL[2] and TextRunner seek to build a usable knowledge base (KB) from the rapidly growing amount of information on the web.

However, these solutions use heuristic approaches to reasoning rather than sound probabilistic inference, and thus face the problem of error propagation.

In this paper, we present a method based on Markov logic[1] for cleaning an automatically extracted knowledge base using only the confidence values and ontological constraints of the original system.

To achieve scalability, we introduce a neighborhood grounding method that only instantiates the part of the network most relevant to the given query.

In experiments on NELL’s KB, our method improves both F1 and AUC.

Background

Never Ending Language Learner

NELL[2] starts from a small number of “seed instances” of each category and relation in the seed ontology.

It uses natural language processing and information extraction techniques to extract candidate instances from a large web corpus, using the current facts in the knowledge base as training examples.

It has four subcomponents to extract candidates: Pattern Learner, SEAL, Morphological Classifier, and Rule Learner.

A major problem of NELL is that the accuracy of the knowledge it acquires gradually decreases as it continues to operate. After the first month, NELL had a precision of 0.9; after two more months, precision had fallen to 0.7;

Coupled training and periodic human supervision are used to alleviate this problem. However, they cannot prevent it entirely.

Methodology

Problem Setting

We’re given a set of ontological constraints and confidences from an IE system such as NELL. They’re encoded with the notations in Table I, e.g.,

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CandCat</td>
<td>new category instance</td>
</tr>
<tr>
<td>CandRel</td>
<td>new relation instance</td>
</tr>
<tr>
<td>PromCat</td>
<td>promoted category instance</td>
</tr>
<tr>
<td>PromRel</td>
<td>promoted relation instance</td>
</tr>
</tbody>
</table>

Our goal is to produce a cleaned up set of category and relation facts, i.e., find the truth assignments of CandCat and CandRel.

This is done by running inference with the MLN formulas and the evidence.

MLN Formulas

Ontological constraints

- Sub(cl,c2) ∧ CandCat(c1) ⇒ CandRel(cl,c2).
- Sub(cl,c2) ∧ CandRel(x,y,r) ⇒ Rel(x,y,r).
- Mut(cl,c2) ∧ CandRel(x,y,r) ⇒ ¬CandRel(cl,c2).
- Mut(cl,c2) ∧ Rel(x,y,r) ⇒ ¬Rel(x,y,r).
- Inv(r1,r2) ∧ Rel(x,y,r) ⇒ Rel(y,x,r).
- Domain(r,c) ∧ Rel(x,y,r) ⇒ CandCat(x,c).
- Range(r,c) ∧ Rel(x,y,r) ⇒ CandRel(x,y,r).

Prior confidence of instances

- conf CandCat(x,c,conf) ⇒ CandCat(x,c).
- conf CandRel(x,y,r,conf) ⇒ CandRel(x,y,r).
- conf PromCat(x,c,conf) ⇒ PromCat(x,c).
- conf PromRel(x,y,r,conf) ⇒ PromRel(x,y,r).

- 0.2 (¬EXISTS conf: CandCat(x,c,conf)) ⇒ ¬CandCat(x,c).
- 0.2 (¬EXISTS conf: CandRel(x,y,r,conf)) ⇒ ¬CandRel(x,y,r).

Seed instance

- CandCat(x,c) ⇒ CandCat(x,c).
- CandRel(x,y,r) ⇒ CandRel(x,y,r).
- PromCat(x,c) ⇒ PromCat(x,c).
- PromRel(x,y,r) ⇒ PromRel(x,y,r).

Experiments

Dataset

- Take the 165th iteration of NELL as a snapshot for the experiments.
- Test on 6 relations and 4 categories.

Models for comparison

- NELL: proposed facts of NELL.
- MLN-0: Markov logic with the information of candidate and promoted facts, but without the ontological constraints.
- MLN-1: Markov logic with the ontological constraints.

A concrete example

- “Los Angeles county” is extracted as a candidate instance for both City and County. Although the former is wrong, it was extracted before the latter and got promoted by NELL since it had strong supporting evidence at that time.

The latter was not promoted because it violated the mutual exclusion rule of the two categories (i.e., Mut(City,County)). Our approach is able to smartly reason about contradictory instances using all available information and therefore corrects this mistake.

References