Profiler: Knowledge Schemas at Scale

Cognitive Computation Group, University of Illinois

Goal and Description

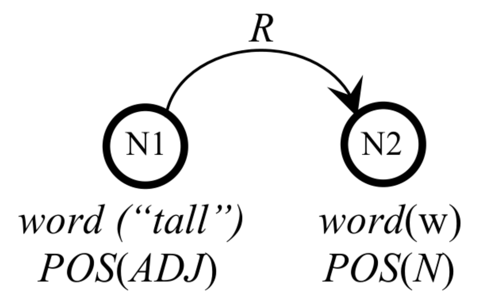
◆Issue

Textual Inference needs additional knowledge. Therefore there is a need to induce external knowledge in NLP tasks.

◆Examples

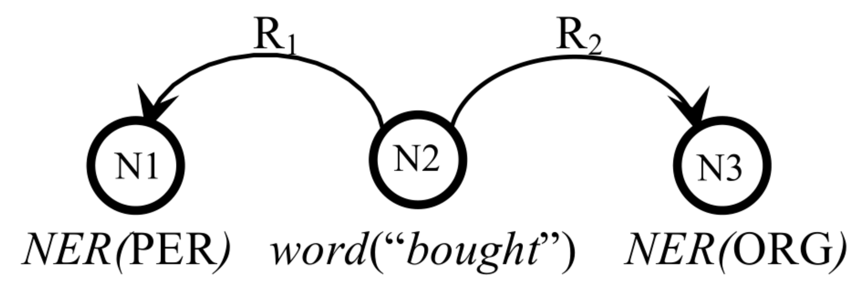
◆Co-reference resolution:

"I chopped down [the tree] with my [axe] because [it] was tall."



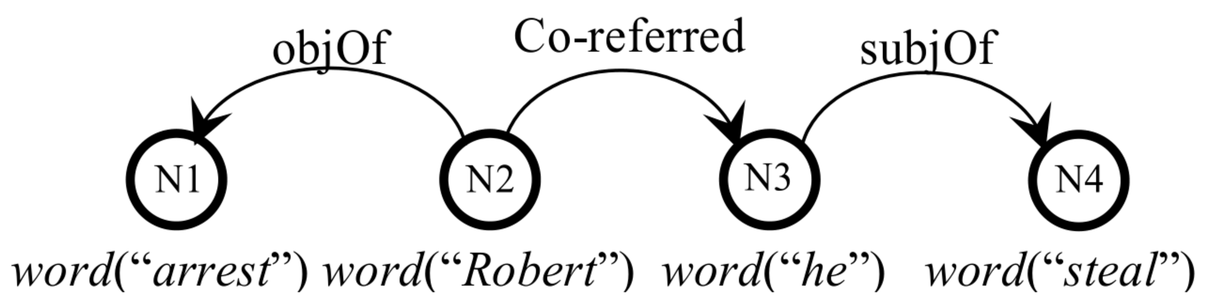
◆Named Entity resolution:

"[Larry Robbins], founder of Glenview Capital Management, bought shares of [Endo International Plc] ..."



◆More intricate co-reference resolution:

"[Jimbo] attacked [Bobbert] because [he] stole an elephant from the zoo."



◆Goal

Creating a knowledge base with

- ◆Knowledge schemas with different patterns
- ◆Extracted automatically and efficiently
- ◆Patterns contain multiple abstraction levels
- ◆Easily extendible to new knowledge patterns

Knowledge Schemas

◆ Feature Description Logic

◆ Generalization of Description Logic (Cumby&Roth,2003)

Attributes: $\mathcal{A} = \{a_1, a_2, \dots\}$

Values: $\mathcal{V} = \{v_1, v_2 \dots\}$

Relations: $\mathcal{R} = \{r_1, r_2, \dots\}$

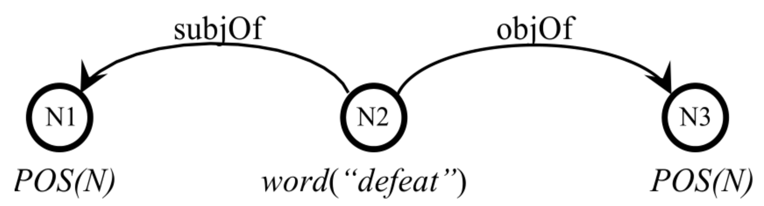
1. For an *attribute* $a \in \mathcal{A}$ and a value $a \in \mathcal{V}$, $a(v)$ is a description, and it represents the set $x \in \mathcal{X}$ for which $a(x, v)$ is $\texttt{True}$.
2. For a description D and a role $r \in \mathcal{R}$, $(r\ D)$ is a role description. Such description represents the set $x \in \mathcal{X}$ such that $\text{r}(x, y)$ is $\texttt{True}$, where $y \in \mathcal{Y}$ is described by D.
3. For given descriptions $D_1, \dots, D_k$, then $(AND\ D_1, \dots, D_k)$ is a description, which represents a conjunction of all values described by individual descriptions.

Describing Knowledge Schema

Given a concept graph, the goal is to *describe* the set of all tuples (containing nodes of the graph), which are compatible with the given graph.

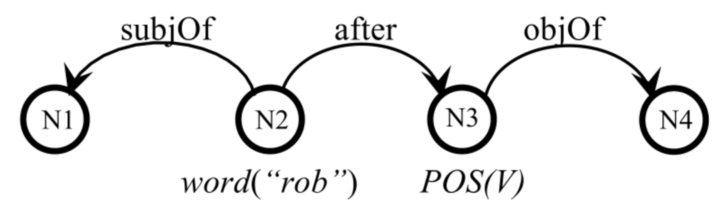
- D_i : the description of node i, i.e. the set of 1-tuples
- $D_{i_1,\dots,i_k}$: the description of nodes $i_1, \dots, i_k$, i.e. the set of k-tuples.

◆Example 1:



$$D_1 = (\textbf{AND}\,(\texttt{POS(N)})\,(\texttt{subjectOf word}(\text{"}defeat\text{"})))$$
$$D_2 = \{\texttt{word}(\text{"}defeat\text{"})\}$$
$$D_3 = (\textbf{AND}\,(\texttt{POS(N)})\,(\texttt{objectOf word}(\text{"}defeat\text{"})))$$
$$D_{1,2,3} = D_1 \otimes D_2 \otimes D_3$$

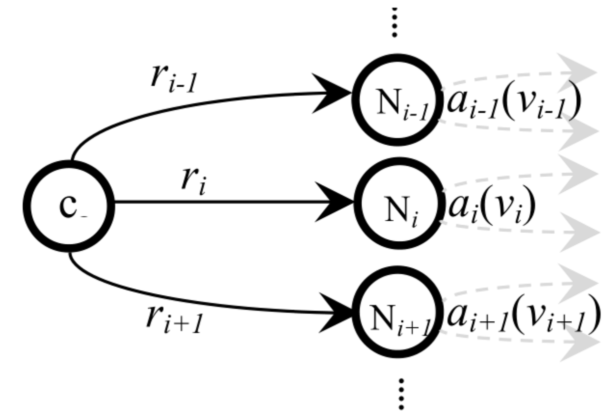
◆Example 2:



$$D_1 = (\texttt{subjectOf word}(\text{"}rob\text{"}))$$
$$D_2 = \{\texttt{word}(\text{"}rob\text{"})\}$$
$$D_3 = (\textbf{AND}\,(\texttt{POS(V)})\,(\texttt{after word}(\text{"}rob\text{"})))$$
$$D_4(w) = (\texttt{objectOf word}(w))\,, \forall w \in D_3$$
$$D_{3,4} = \bigcup_{w \in D_3} (\{w\} \otimes D_4(w))$$
$$D_{1,2,3,4} = D_1 \otimes D_2 \otimes D_{3,4}$$

A General Description of Knowledge Schemas

Given a concept graph, the goal is to give a general description of the elements that accord to the description of the graph.



1. Description of each based on its *parent* node:

$$D_i(c) = (\textbf{AND}\,(a_i(v_i))\,(r_i\ \ \texttt{word}(c))),\quad \forall c \in D_{\text{parent}}$$

2. Chaining description:

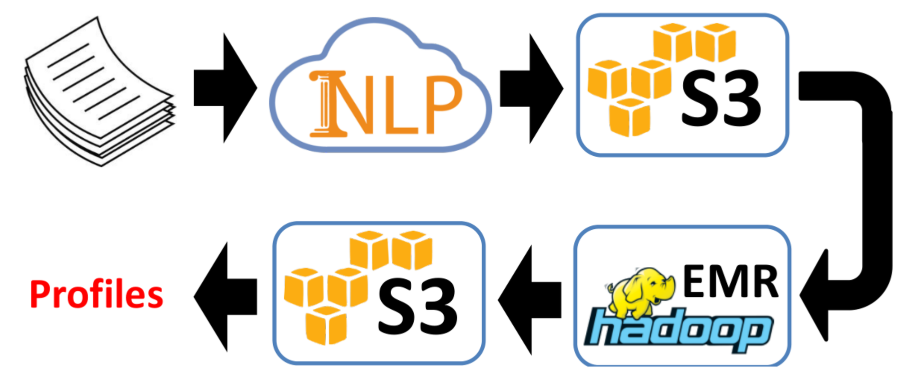
$$D_{\text{parent, child}} = \bigcup_{c \in D_{\text{parent}}} \left[\{c\} \otimes \left(\bigotimes_{i \in \mathcal{I}} D_i(c) \right) \right]$$

Acquisition Procedure

1. Process data with IllinoisCurator deployed on IllinoisCloudNLP
2. Store the data on S3, Amazon's scalable storage
3. Process the data using MapReduce on Amazon EC2
4. Store the result on Amazon S3
5. Import the results to MongoDB, a scalable dababase supporting flexible indexing

Annotated 4,019,936 Wikipedia documents with 1,455 GB size with 200 mid-end EC2 nodes in 3 hours, at a cost of $420.

The result has size 198 GB and it contains 3,636,263 profiles for Wikipedia entities and 313,156 profiles for Verbsense entities.



Experiments

Visualizing sample schemas

"Verb After" schema

Detaless Classification of Professions-People

- We create a labeled dataset of people-professions, using Wikipedia, such that for any entity its professions is labeled.
- For a given entity, we create a feature for it, based on a select set of schemas.
- For each profession, we average the feature vectors of a bunch of entities.
- Now given the feature vectors of professions, for an unseen entity, decide the profession of an unseen entity based on its profiler feature vector.
- **Result:** In 72.1% of the test cases, the correct answer is among the top-5 prediction.

Winograd Challenge

- We follow the setting in Peng et al [2015].
- We add extract information based on their setting from our schemas and add them as both constraints and features.

Ex.1 The [ball] e1 hit the [window] e2 and Bill repaired [it] pro.
Ex.2 The [ball] e1 hit the [window] e2 and Bill caught [it] pro.

Dataset	*Winograd*	*WinoCoref*
Metric	Precision	AntePre
Rahman *et al* [2012]	73.05	——
Peng *et al* [2015]	76.41	89.32
Our paper	77.16	89.77

References

- Cumby, Chad M., and Dan Roth. "Learning with feature description logics."*ILP*. Springer, 2003. 32-47.
- Peng, Haoruo, Daniel Khashabi, and Dan Roth. "Solving Hard Coreference Problems." *Urbana* 51: 61801.