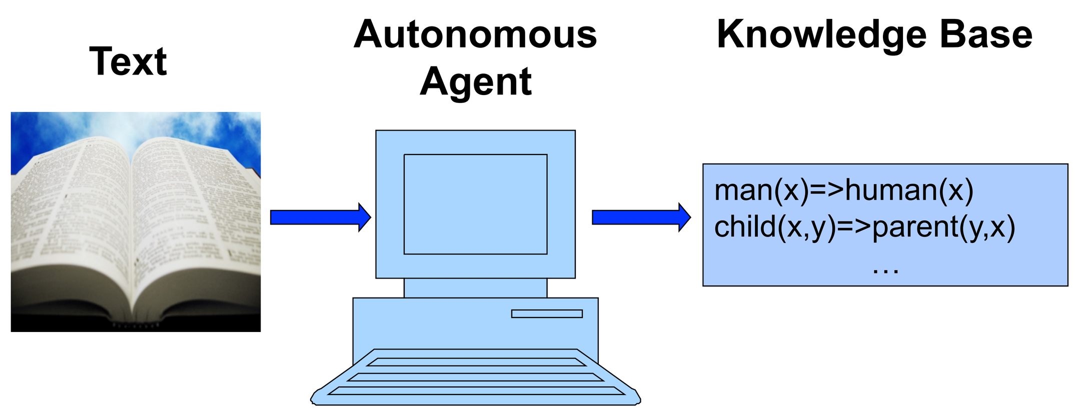
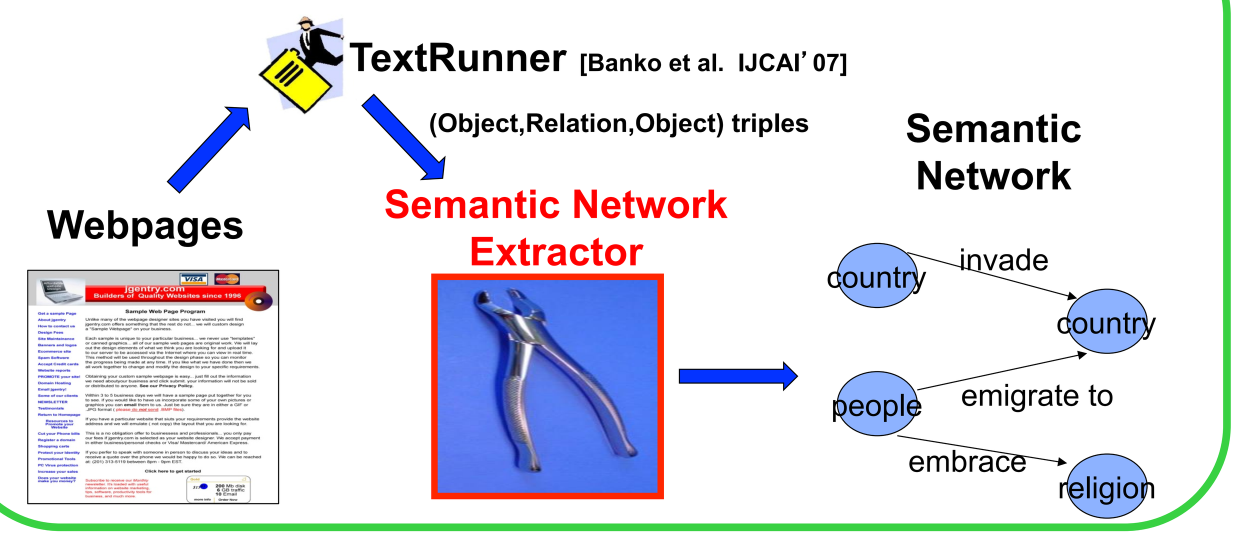
Extracting Semantic Networks From Text via Relational Clustering

Stanley Kok & Pedro Domingos
Dept. of Computer Science and Engineering
University of Washington, Seattle, USA

Grand Goal in AI



We Take a Step Towards Goal



State of the Art

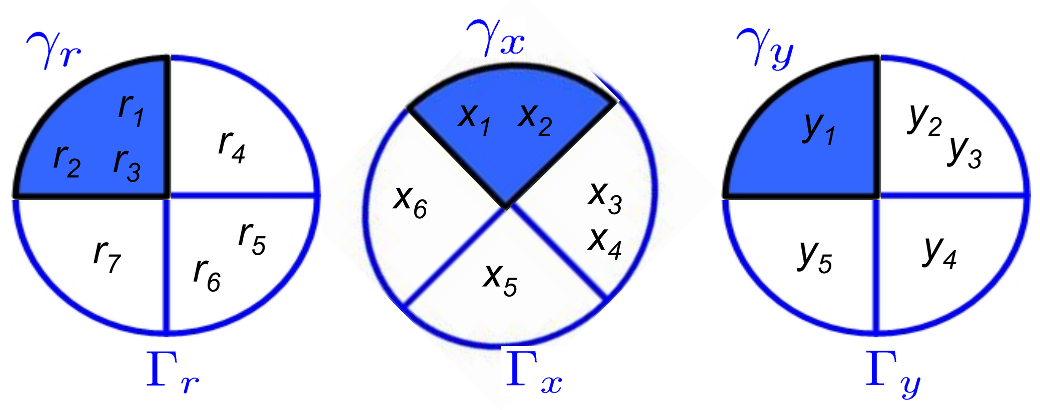
- Supervised approaches
  - Manual annotation of training data
  - Not scalable to Web
  - e.g., Semantic Parsing [Wong & Mooney, ACL' 07]
- Unsupervised approaches
  - Extracts noisy & sparse ground facts
  - No high-level knowledge that generalizes ground facts
  - e.g., TextRunner [Banko et. al., IJCAI' 07]

Our Approach: Semantic Network Extractor

- Unsupervised, domain-independent
- Scales to the Web
- Based on Markov logic
- Input: triples ***r(x,y)*** from TextRunner
- Output: simple semantic network
- Clusters objects and relations simultaneously
- Number of clusters need not be specified in advance
- Cluster relations by objects they relate and vice versa
- First step towards learning full-blown logical representation from Web text

Symbols

- Cluster: γ_r γ_x γ_y
- Clustering: Γ_r Γ_x Γ_y
  i.e, partitionings of relation and object symbols
- Atoms: $r \in \gamma_r$ $x \in \gamma_x$ $y \in \gamma_y$ $r(x,y)$
- Cluster combination: $(\gamma_r, \gamma_x, \gamma_y)$
  i.e., clusters to which corresponding symbols belongs



SNE Rules

- Four rules
- Each symbol belongs to exactly one cluster

$$\infty \quad \forall x\, \exists^1 \gamma\;\; x \in \gamma$$

- Exponential prior on #cluster combinations

$$-\lambda \quad \forall \gamma_r, \gamma_x, \gamma_y\; \exists r, x, y \;\; r \in \gamma_r \wedge x \in \gamma_x \wedge y \in \gamma_y$$

- Most symbols tend to be in different clusters

$$\mu \quad \forall x, x', \gamma_x, \gamma'_x \;\; x \in \gamma_x \wedge x' \in \gamma'_x \wedge x \neq x' \Rightarrow \gamma_x \neq \gamma'_x$$

- **Atom prediction rule:** Truth value of atom is determined by cluster combination it belongs to

$$w \quad \forall r, x, y, +\gamma_r, +\gamma_x, +\gamma_y \;\; r \in \gamma_r \wedge x \in \gamma_x \wedge y \in \gamma_y \Rightarrow r(x,y)$$

Wt of rule is log-odds of atom in its cluster combination being true

$$= \log \frac{t+\alpha}{f+\beta}$$

Smoothing parameters

#true & #false atoms in cluster combination

Learning SNE Model

Learning consists of finding

- Weights of atom prediction rules
- Cluster assignment $\Gamma=(\Gamma_r,\Gamma_x,\Gamma_y)$: assignment of truth values to $r \in \gamma_r$, $x \in \gamma_x$ and $y \in \gamma_y$ atoms

that maximize log-posterior probability

$$\log P(\Gamma|R) \propto \log P(\Gamma) + \log P(R|\Gamma)$$

vector of truth assignments to all observed ground atoms *r(x,y)*

first three rules

atom prediction rule

Log-Posterior

- Can be computed in closed-form

$$\log P(\Gamma|R) = \sum_{k\in K}\left[\overbrace{t_k \log\left(\frac{t_k+\alpha}{t_k+f_k+\alpha+\beta}\right)}^{\text{prob. atom is true}} + \overbrace{f_k \log\left(\frac{f_k+\beta}{t_k+f_k+\alpha+\beta}\right)}^{\text{prob. atom is false}}\right] - \lambda m_{cc} + \mu d + \mathcal{C}$$

Set of cluster combinations
Intractable!
#cluster combinations
#pairs of symbols in different clusters
constant

- Assume atoms in cluster combinations with only false atoms all belong to single 'default' cluster combination
- Only sum over cluster combinations with ≥ 1 true atom (the number of such combinations is at most the number of triples in the data)

$$\log P(\Gamma|R) = \sum_{k\in K^+}\left[t_k \log\left(\frac{t_k+\alpha}{t_k+f_k+\alpha+\beta}\right) + f_k \log\left(\frac{f_k+\beta}{t_k+f_k+\alpha+\beta}\right)\right] + \left(|S_r||S_x||S_y| - \sum_{k\in K^+}(t_k+f_k)\right)\log(p_{false}) - \lambda m^+_{cc} + \mu d + \mathcal{C}$$

Set of cluster comb. with $\geq$ 1 true *r(x,y)* atom

S_i **set of symbols of type *i***

#false atoms in cluster comb. with only false atoms

Pr(atom=false)

#cluster comb. with $\geq$ 1 true *r(x,y)* atom

Search Algorithm

- Approximation: Hard assignment of symbols to clusters
- Searches over cluster assignments, evaluate each by its posterior prob.
- Agglomerative clustering
  - Start with each *r, x, y* symbols in own cluster
  - Merge pairs of clusters in bottom-up manner
- Canopies
  - e.g., merge relations *r1* and *r2* if arguments in common; *r1(x,y)* and *r2(x,y)*
- Change in log-posterior in merging two clusters can be computed efficiently (see paper)

Experimental Data

- 2.1 million triples extracted in Web crawl by TextRunner
  - e.g., *named_after(Jupiter,Roman_god)*, *upheld(Court,ruling)*
  - 15,872 *r* symbols, 700,781 *x* symbols, 665,378 *y* symbols
- Only consider symbols appearing $\geq$ 25 times
  - 10,214 *r* symbols, 8942 *x* symbols, 7995 *y* symbols
  - 2,065,045 triples contain at least one such symbol

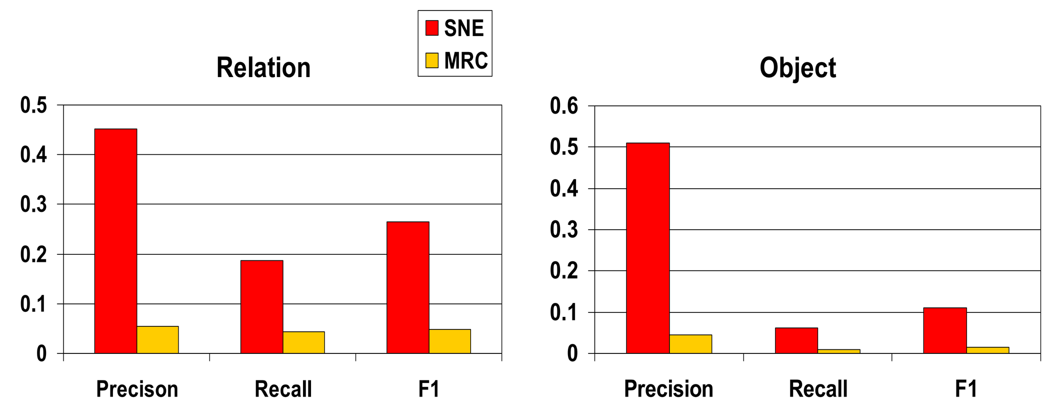
Comparison Systems

- Multi-Relational Clustering (MRC) [Kok & Domingos, ICML' 07]
- Information-Theoretic Co-clustering (ITC) [Dhillon et al. , KDD' 03]
- Infinite Relational Model (IRM) [Kemp et al., AAAI' 06]

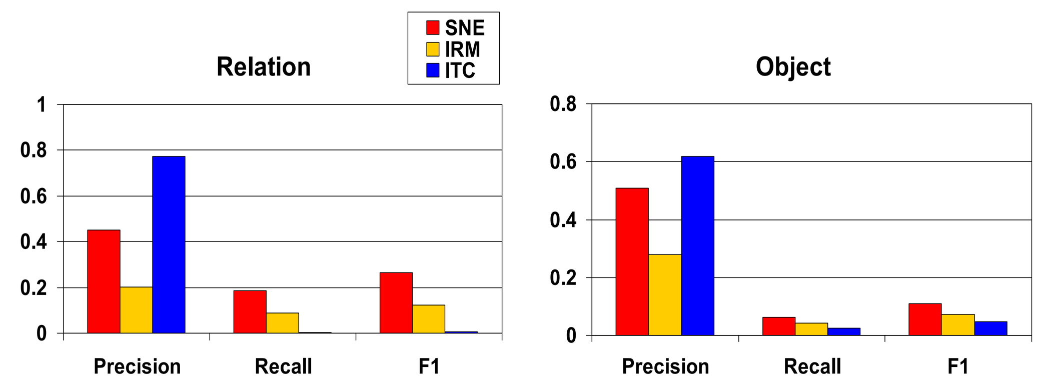
Evaluation

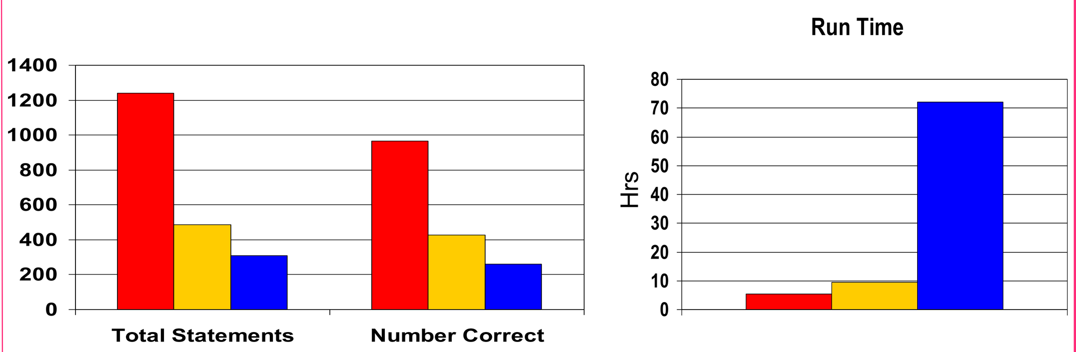
- Pairwise precision, recall, & F1 against manually created gold standard
  - 2688 *r* symbols, 2568 *x* symbols, 3058 *y* symbols assigned to non-unit clusters
  - 874 *r* clusters, 511 *x* clusters, 700 *y* clusters
  - Remaining symbols assigned to unit clusters
- Correct semantic statements
  - Cluster combinations with $\geq$ 5 true ground *r(x,y)* atoms

SNE vs. MRC

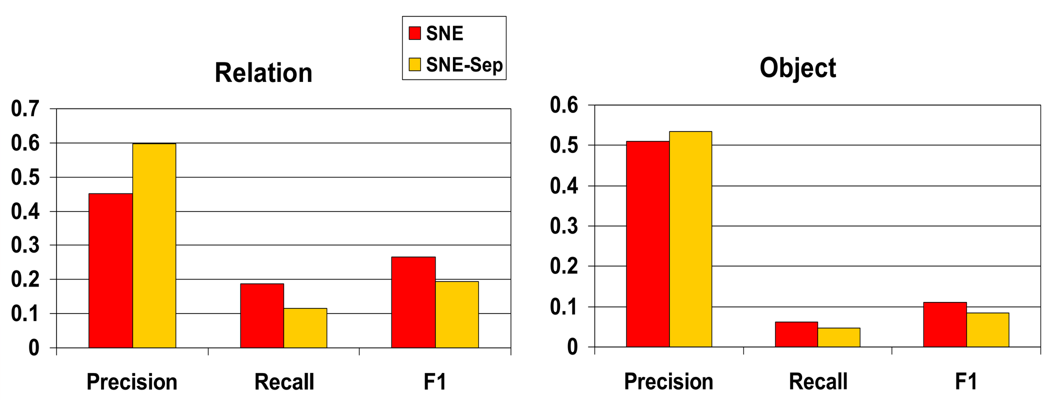


SNE vs. IRM vs. ITC

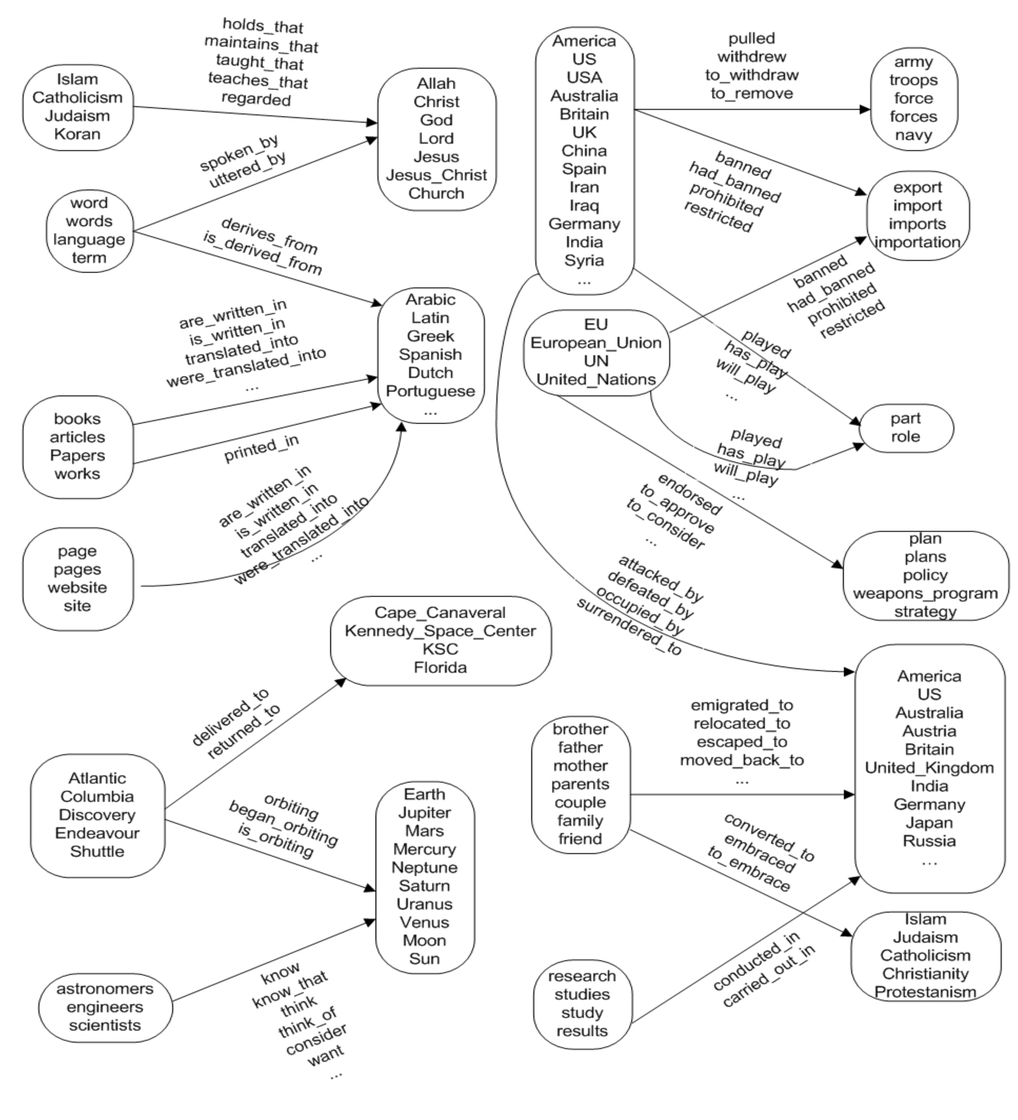




SNE Full Joint Model vs. Separate Clustering



Snippet of Semantic Network Learned



Future Work

- Integrate tuple extraction into SNE
- Learn richer semantic networks
- Learn logical theories
- Etc.