Learning Random Walk Models for Inducing Word Dependency Distributions

Kristina Toutanova, Christopher Manning, Andrew Ng

Proceedings of the 21st International Conference on Machine Learning, 2004

Erin R. Rhode
University of Michigan
EECS 767
March 15, 2006

Outline

- Markov Chains
- Prepositional Phrase Attachment Problem
- Random Walks

Markov Chains

- Markov Distribution
  - States $\mathcal{S}$
  - Initial distribution: $p_0(S)$ over all $S \in \mathcal{S}$
  - State transition probabilities: $p(S_t|S_{t-1})$

Markov Chains

- Generating a Markov Chain
  - Generate initial state, S_0, based on p_0
  - Generate remaining states, S_t, based on transition probabilities

- Stationary distribution (if the limit exists):

$$\pi(s) = \lim_{t \to \infty} P(S_t = s)$$

Markov Chains

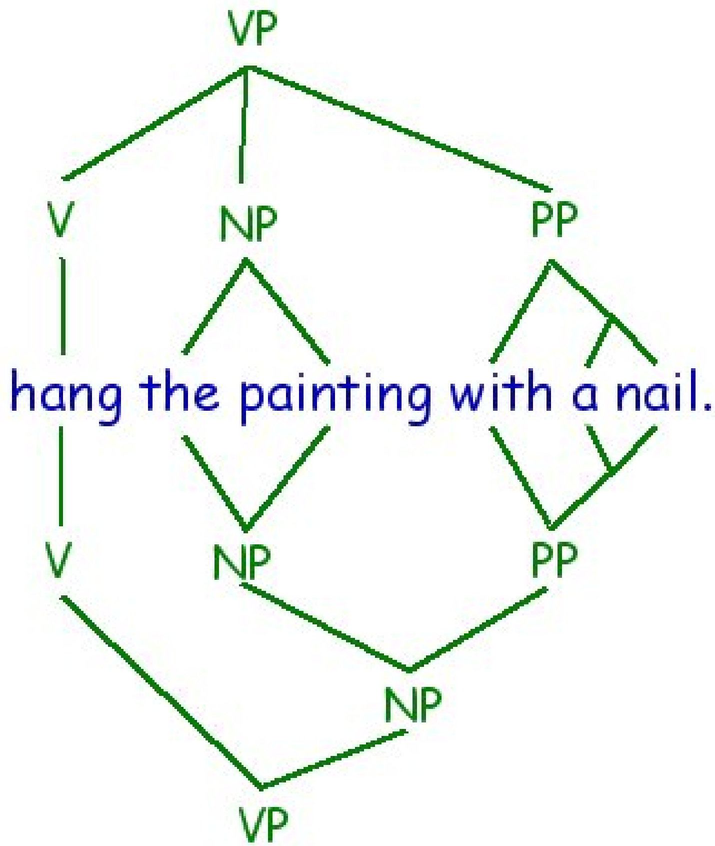
- How to ensure a stationary distribution?
  - Include probability $\gamma > 0$ of resetting to initial distribution p_0
  - New state transition distribution:

$$p(S_t|S_{t-1}) = \gamma p_0(S_t) - (1-\gamma)p'(S_t|S_{t-1})$$

  - Stationary distribution (can be approximated) becomes:

$$\pi(s) = \gamma \sum_{t=0}^{\infty} (1-\gamma)^t P(S_t = s)$$

PP Attachment Problem



PP Attachment Problem

- ⋆ v:hang n_1:painting p:with n_2:nail

- ⋆ $P(V, N_1, P, N_2, Att) \Rightarrow P(Att|V, N_1, P, N_2)$

  - ⋆ $Att = va$ (verb attachment) or na (noun attachment)

  - ⋆ $P(v, n_1, p, n_2, va) =$

    $$P(p, va)P(v|p, va)P(n_1|p, va, v)P(n_2|p, \mathbf{v}, va)$$

  - ⋆ $P(v, n_1, p, n_2, na) =$

    $$P(p, na)P(v|p, na)P(n_1|p, na, v)P(n_2|p, \mathbf{n_1}, na)$$

Random Walks

- $\star$ But how to generate the probabilities?
  - $\star$ Random Walks!

- $\star$ $\mathcal{S} = \mathcal{W} \times \{0, 1\}$
  - $\star$ $\mathcal{W}$ = all words, 0 = head state, 1 = dependent state
  - $\star$ nail = {nail, 1}, hang = {hang, 0}

- $\star$ $p_0(verb, 0) = 1$

Random Walks

- ⋆ Separate graphs for each probability distribution (e.g. $P(N_2|p, v, va)$)

- ⋆ Separate transition matrics (graphs) for each p

- ⋆ "Short" walks away are likely related; "long" walks are not as likely

Transition types

- ⋆ Empirical transitions: What was seen in the training set

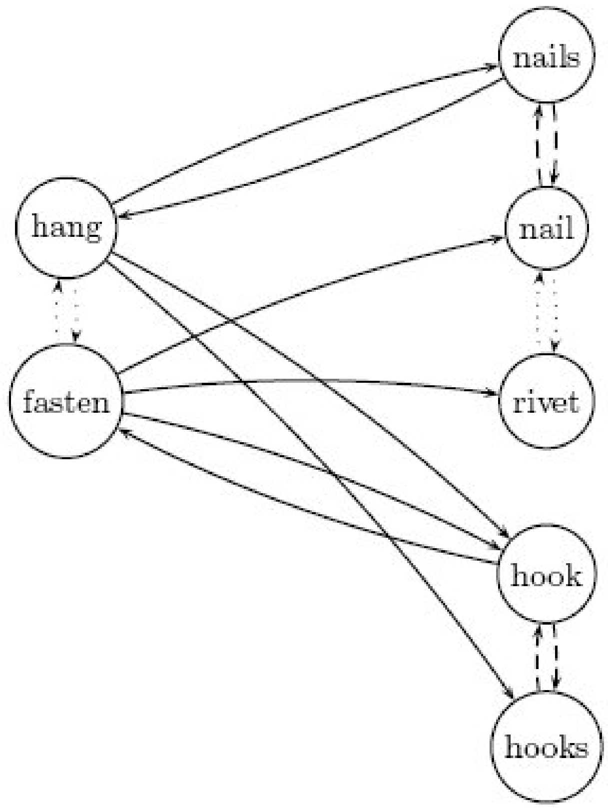
- ⋆ Morphology: Words with same root and same marker (nail vs. nails)

- ⋆ WordNet Synsets: Semantically similar words (nail vs. rivet)

Transition types

- External Corpus: Similar to empirical transition, but includes noisy data (from a statistical parser)
- $V \rightarrow V$: Verbs that take similar dependents (hang vs. fasten)
- $N \rightarrow V$: Nouns with similar heads (nail vs. rivet)

Random Walks



State space for learning $P_{with}(n_2, v)$

Experiments

- ⋆ Penn Treebank WSJ for training and test set
- ⋆ BLLIP corpus for noisy data (1.8 million sentences; Charniak parser)
  - ⋆ 567,582 ambiguous PP tuples extracted
- ⋆ Run the Markov chain d time steps (degree)
- ⋆ Try different types of links

Results

Model	Degree	Accuracy
Baseline	1	85.86%
Baseline+stem verbs	1	86.02%
Morph Verbs	2,3	86.08%
Morph Verbs and Nouns	2,3	86.18%
Morph & Synonyms	2,3	86.53%
Sim_{JS_β}	2,3	86.44%
Final	2,3	87.54%

⋆ Baseline: run chain for one step (linear interpolation)

Conclusions

- Random walk models provide general framework for similarity based smoothing

- Random walk of step 1 is just linear interpolation (baseline)

- Gives high performance in prepositional phrase attachment
  - Can be applied to other problems as well