

Information extraction with HMMs and shrinkage

“Honey, I shrunk the HMM”

Dayne Freitag and Andrew McCallum
AAAI '99 workshop on machine learning
for information extraction
Presented by Greg Hamerly
5/14/2001



Where we are going

- Information extraction
- Hidden Markov model
- Difficulty: sparse training data
- Probability smoothing via shrinkage
- Experiments



Information extraction (IE)

- We want to extract specific information from text documents.
- For example, from web pages about presentations, we might want:
 - the start time
 - the presenter
 - the location

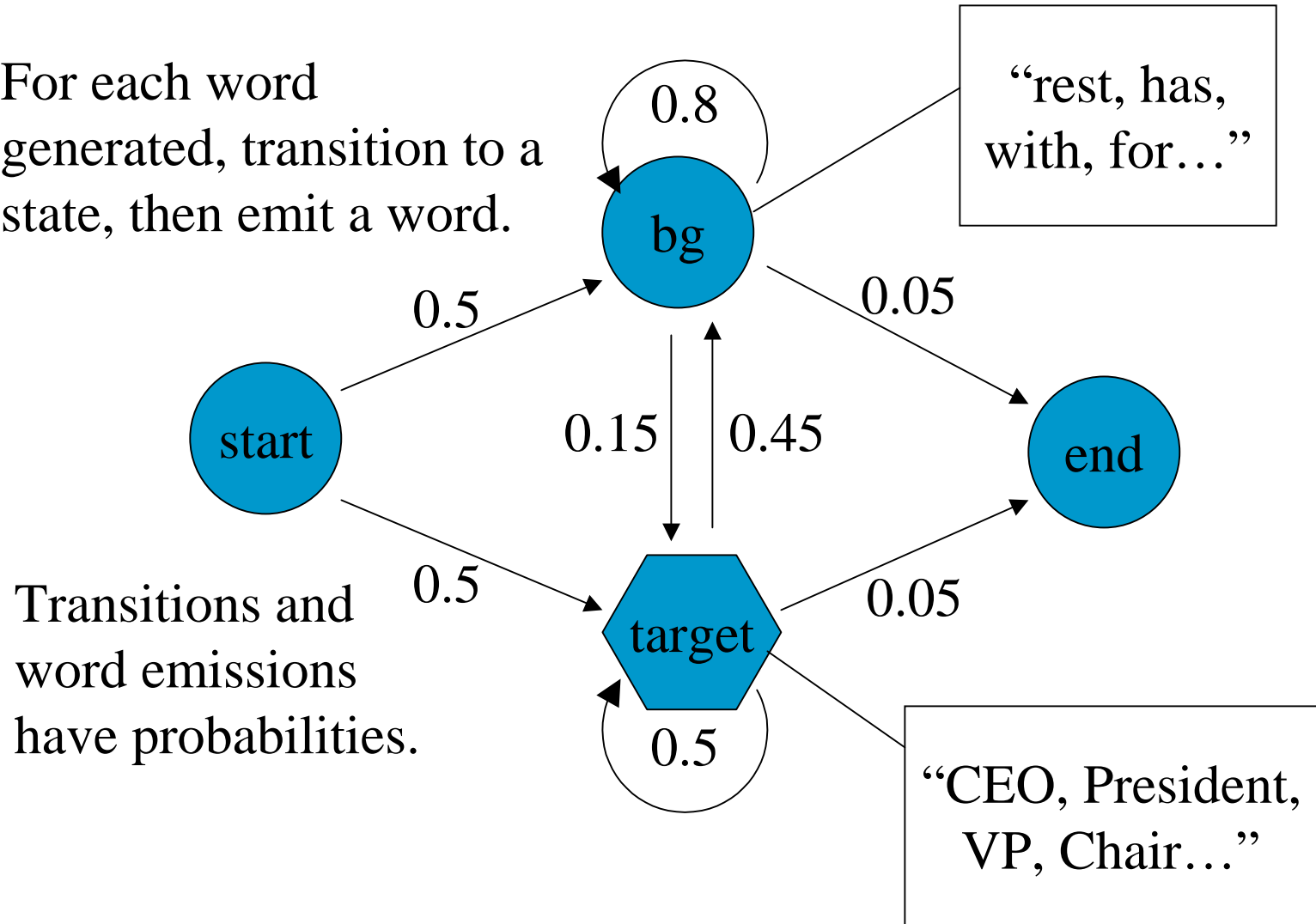


Hidden Markov model (HMM)

- An HMM is a generative data model.
- HMMs are commonly used in speech recognition.
- An HMM is good at using context.
- In this work, a single HMM is used to model a whole document.

Example HMM for job titles

For each word generated, transition to a state, then emit a word.



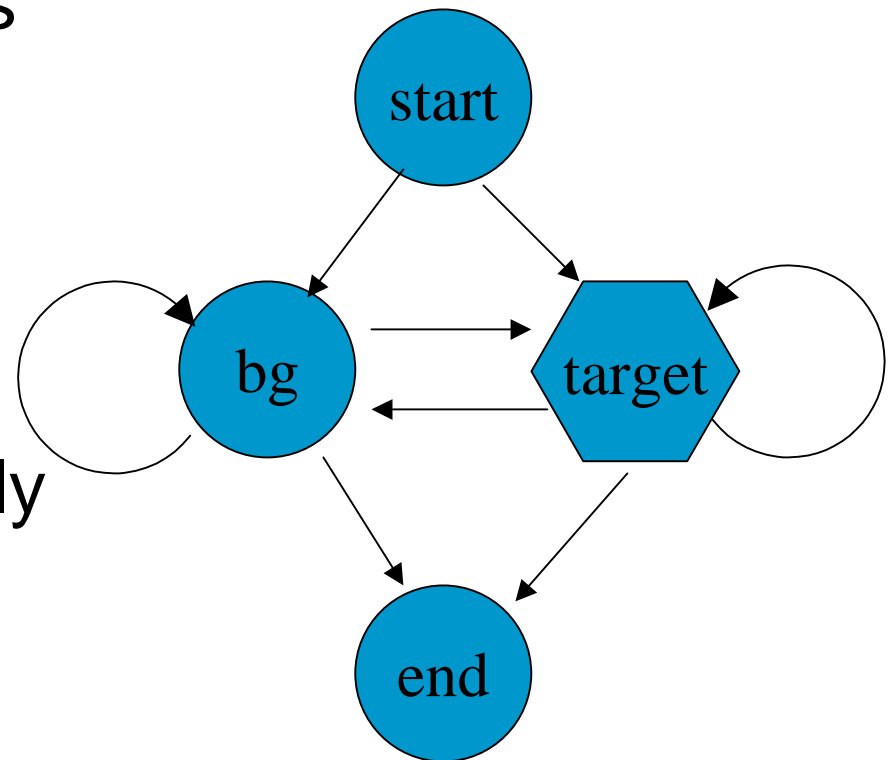


HMMs for information extraction

- Target states model the text of interest.
- Other states model background language, including the prefix and suffix of the target state.
- The Viterbi algorithm can extract information from documents modeled by an HMM.

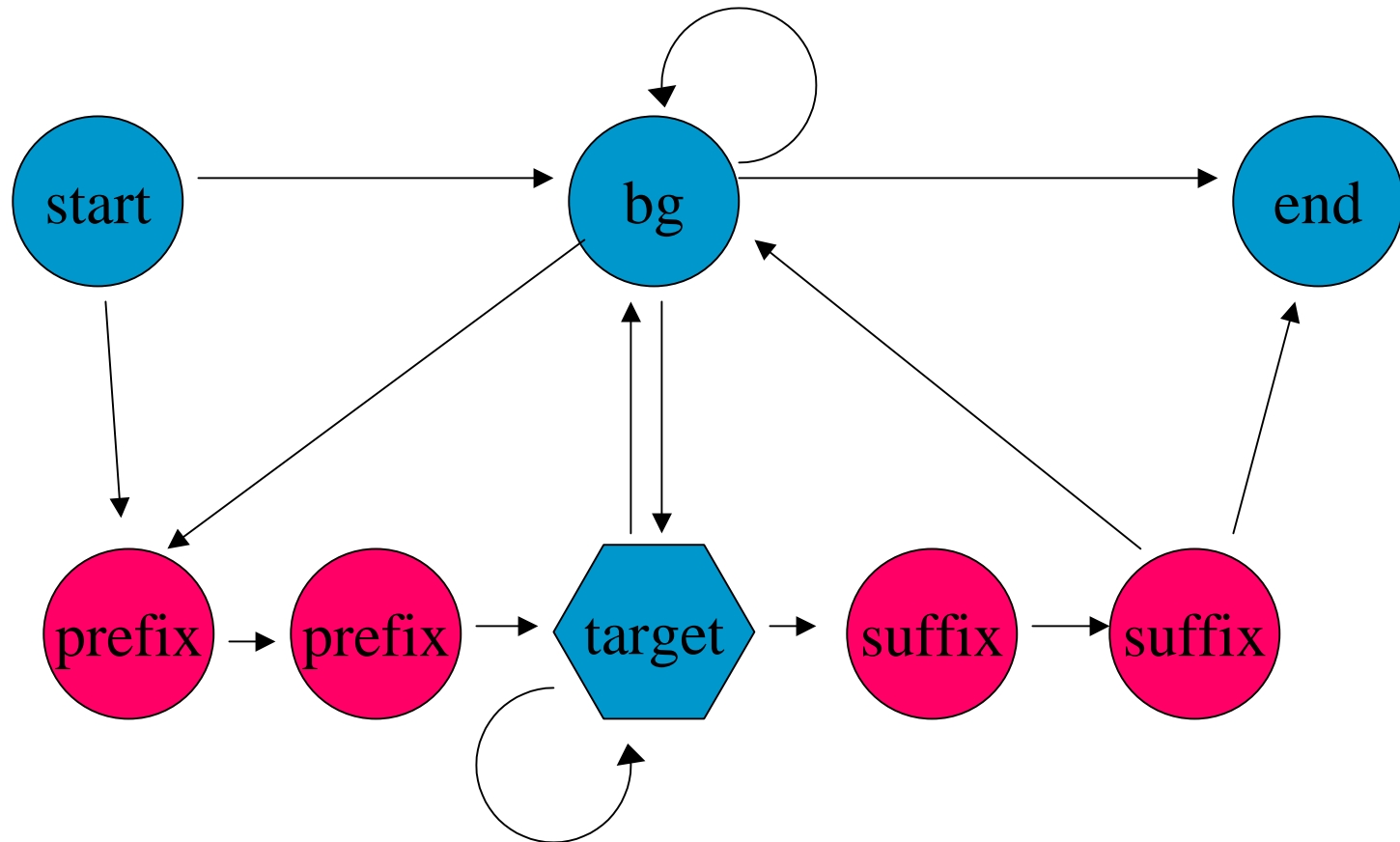
Constructing HMMs for IE

- Each HMM extracts one field.
- The HMM has background and target states.
- The HMM is not fully connected.



Structural options

- A “window” of prefix and suffix states may capture context of the target.





The Viterbi algorithm for HMMs

- Given a document, the algorithm finds a **maximum-probability path** through an HMM.
- We can use this path to extract parts of the document of interest.
- The algorithm is fast, running in time $O(ds^2)$, where d = length of document, s = number of HMM states



Learning for HMMs

- HMMs can be learned from annotated documents. Annotations label each word as a target or a non-target word.
- We could learn HMM structure, probabilities, or both.
- In this paper, the authors fix structure, and learn the transition and emission probabilities.



Improving probability estimates

- The objective of learning is to give high probabilities to training documents.
- The result of learning is estimated probabilities for vocabularies and transitions.
- **Difficulty:** sparse training data causes poor probability estimates. **Unseen words have emission probabilities of zero.**



Probability smoothing

- We need to smooth the vocabulary probabilities. The authors cite a variety of methods:
 - Laplace smoothing
 - Absolute discounting
 - Shrinkage



Laplace smoothing

- Laplace smoothing adds pseudo-counts to all word frequencies. All estimates move towards the uniform distribution.
 - $N(w,s)$: # of times state s has seen word w
 - $N(s)$: # of word occurrences seen in state s
 - V : entire vocabulary (all unique words)
- $P(w|s) = (N(w,s) + 1) / (N(s) + |V|)$
- All unseen words have equal, non-zero probability



Absolute discounting

- Subtract a fraction $0 < d < 1$ from seen words to give to unseen words.
 - $V(s)$: vocabulary (unique words) of state s
 - $Z(s) = V - V(s)$
- $P(w|s) = (N(w,s) - d) / N(s)$ if $N(w,s) > 0$
- $P(w|s) = |V(s)| d / |Z(s)|$ if $N(w,s) = 0$



Shrinkage

- Shrinkage smoothes the distribution of a state towards that of states that are more data-rich.
- It uses a linear combination of probabilities.
- $P(w|s) = \lambda_1 P(w|s_1) + \lambda_2 P(w|s_2) + \dots$
- (where s_1 = original state, s_2 = larger context, etc.)



The range of smoothing influence

- Laplace smooths all state vocabulary distributions towards the same uniform distribution.
- Absolute discounting smooths the word distribution within a state.
- Shrinkage uses *context* to smooth distributions towards those in states that are similar but have more data.



Shrinkage, structural complexity

- Simple HMMs have robust probability estimates, but are poor at discrimination.
- Complex HMMs can learn a concept more precisely, but are more prone to overfitting.
- Shrinkage tries to balance these two by smoothing: a complex HMM can have advantages of a simple one.



Smoothing alternatives

- We compare four types of smoothing. Three are variants of shrinkage.
 - Absolute discounting
 - Uniform shrinkage
 - Global shrinkage
 - Hierarchical shrinkage



Uniform shrinkage

- All state vocabulary distributions are shrunk towards the uniform distribution.
- Similar to Laplace smoothing, but the λ 's are learned via EM.

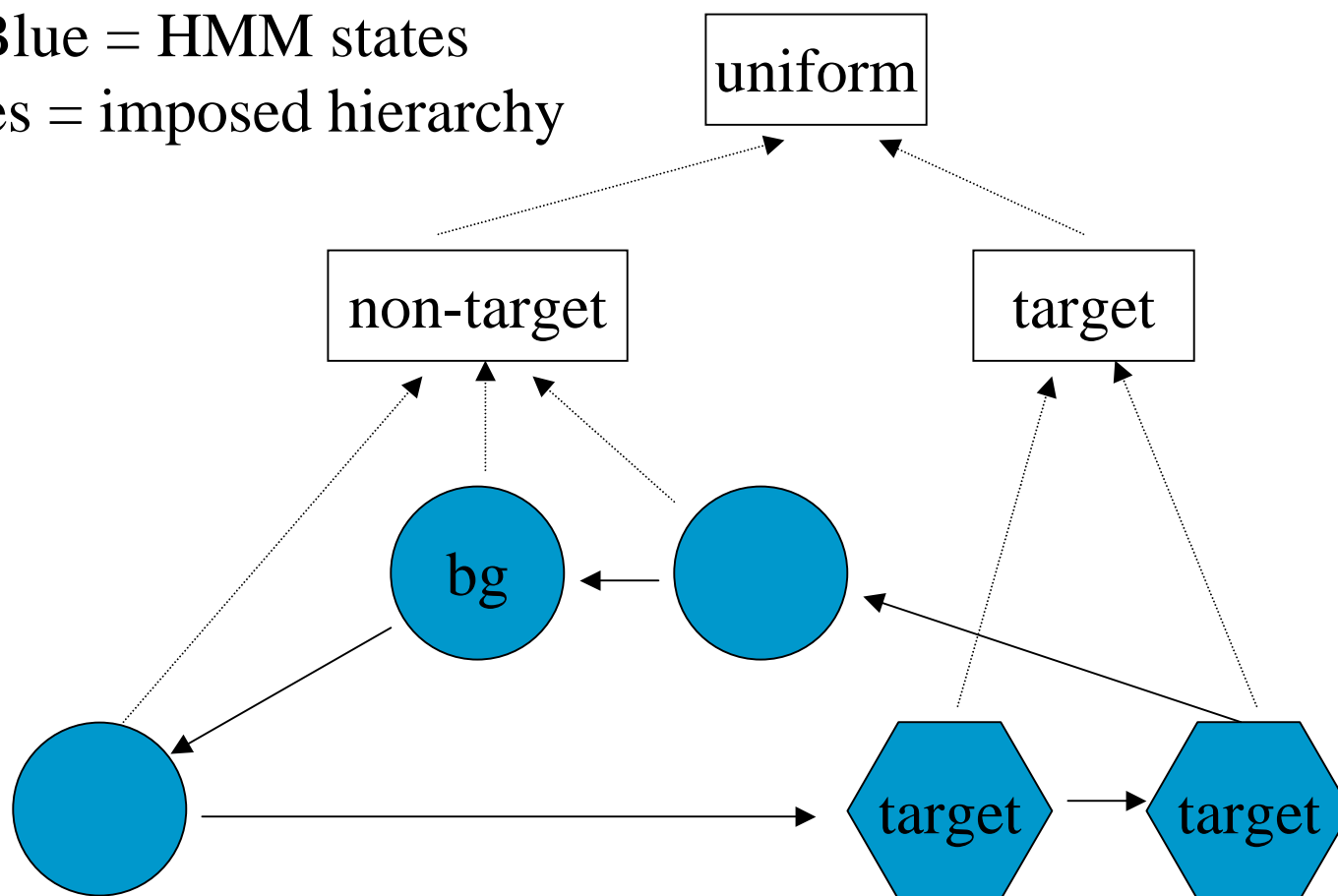


Global shrinkage

- The distributions of all target states are shrunk towards one common parent.
- Non-target state distributions towards another common parent.
- Presumably, the parent states are shrunk towards the uniform distribution.

Global shrinkage

Blue = HMM states
Boxes = imposed hierarchy





Hierarchical shrinkage

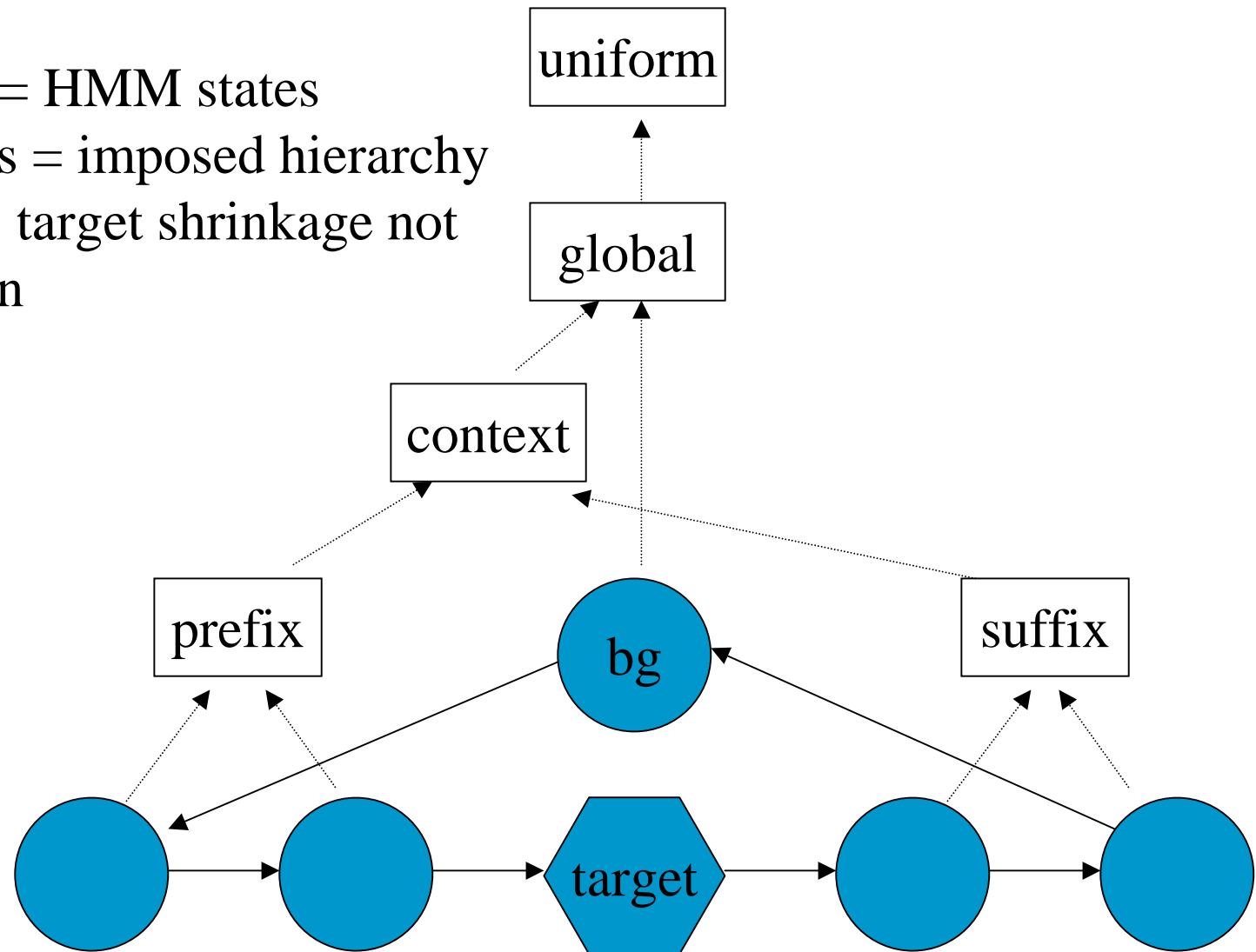
- Target distributions are shrunk towards one parent.
- Non-target distributions shrink according to an imposed hierarchy.
- All distributions are eventually shrunk towards the uniform distribution.

Hierarchical shrinkage

Blue = HMM states

Boxes = imposed hierarchy

Note: target shrinkage not shown





Shrinkage probability estimates

- The probability for a word is now given by a linear combination.
- Lambda denotes the shrinkage prior.
- j denotes the state, i denotes the shrinkage ancestor.

$$\hat{P}(w|s_j) = \sum_{i=1}^k \lambda_j^i P(w|s_j^i)$$



Expectation-maximization

- Now we have extra free parameters (the lambdas). We use EM to learn them.

- Initialize: $\lambda_j^i = \frac{1}{k}$

- E-step:
$$\beta_j^i = \frac{\lambda_j^i P(w_t | s_j^i)}{\sum_m \lambda_j^m P(w_t | s_j^m)}$$

- M-step:
$$\lambda_j^i = \frac{\beta_j^i}{\sum_m \beta_j^m}$$



Determining mixture weights

- Use all the data for EM training, employing a holdout vocabulary set in the E-step.
- Use each word as a test to determine the predictive power of the model.



Experiments

- Two corpora:
 - university seminar announcements
(extract: speaker, location, start, end)
 - corporate acquisitions from Reuters
(extract: acquired, purchaser, acquabr,
dollar amount, status)
- Experiments using
 - various smoothing techniques
 - various prefix/suffix window sizes



Metric of performance: “F1”

- C = number of documents with items extracted correctly
- Precision (P) = C / number of predicted extractions made
- Recall (R) = C / number of documents annotated with the item to extract
- $F1 = 2 / (1/P + 1/R)$ (harmonic mean)



Prefix/suffix window size results

Window Size	Speaker	Location	Stime	Etime
1	0.431	0.797	0.943	0.771
4	0.460	0.653	0.960	0.716
10	0.363	0.558	0.967	0.746

It is not clear which window size is best, though size = 1 appears to work well and is simplest.



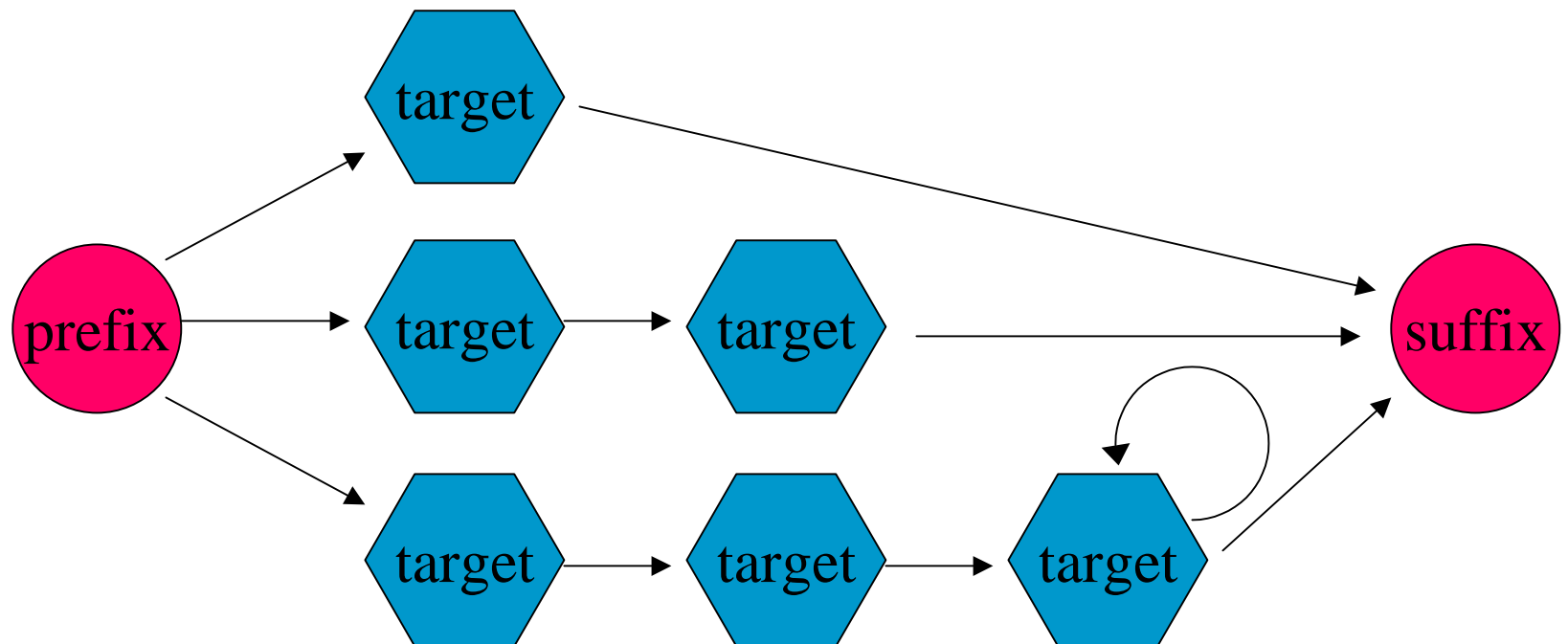
Shrinkage results (single target)

Shrinkage	Speaker	Location	Stime	Etime
Abs. Disc.	0.460	0.960	0.960	0.716
Uniform	0.499	0.660	0.971	0.840
Global	0.558	0.758	0.984	0.589
Hier.	0.531	0.695	0.976	0.565

Again, it is not clear which method is best. Shrinkage usually outperforms absolute discounting.

Multiple target paths

- Several parallel target paths can be used to capture a disjunctive concept.





Four parallel target paths

Shrinkage	Speaker	Location	Stime	Etime
Abs. Disc.	0.513	0.735	0.991	0.814
Uniform	0.614	0.776	0.991	0.933
Global	0.711	0.839	0.991	0.595
Hier.	0.672	0.850	0.987	0.584

Again, it is not clear which method is best. The authors point out the improvement in the Speaker field extraction.



Mixture weights for prefix path

Dist.	Speaker	Location	Stime	Etime
1	0.84	0.84	0.92	0.95
2	0.81	0.90	0.98	0.98
3	0.73	0.80	0.85	0.95
4	0.65	0.74	0.86	0.93

Prefix states closer to the target generally have larger weights, indicating greater importance in the extraction task.



Comparison with SRV

	Speaker	Location	Stime	Etime		
SRV	0.703	0.723	0.988	0.839		
HMM	0.711	0.839	0.991	0.595		
	Acquired	Purchaser	Acqabr	Dlramt	Status	
SRV	0.343	0.429	0.351	0.527	0.380	
HMM	0.309	0.481	0.401	0.553	0.467	

SRV is a “consistently strong rule-learning algorithm”

HMM uses window = 4, 4 target paths, global shrinkage



Critiques of this paper

- Shrinkage is a fairly complex method.
- It does not consistently improve upon absolute discounting.
- The authors don't report error bars.