Word Prediction With Suffix Trie HMM's

Bernhard Bauer <bauerb@in.tum.de>

Jul 4, 2008

The task

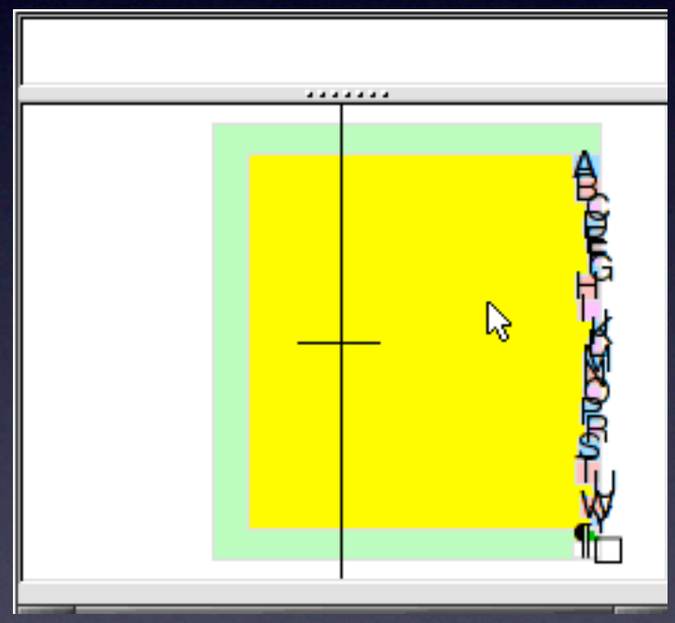
- Statistical word prediction
- Online learning

Applications

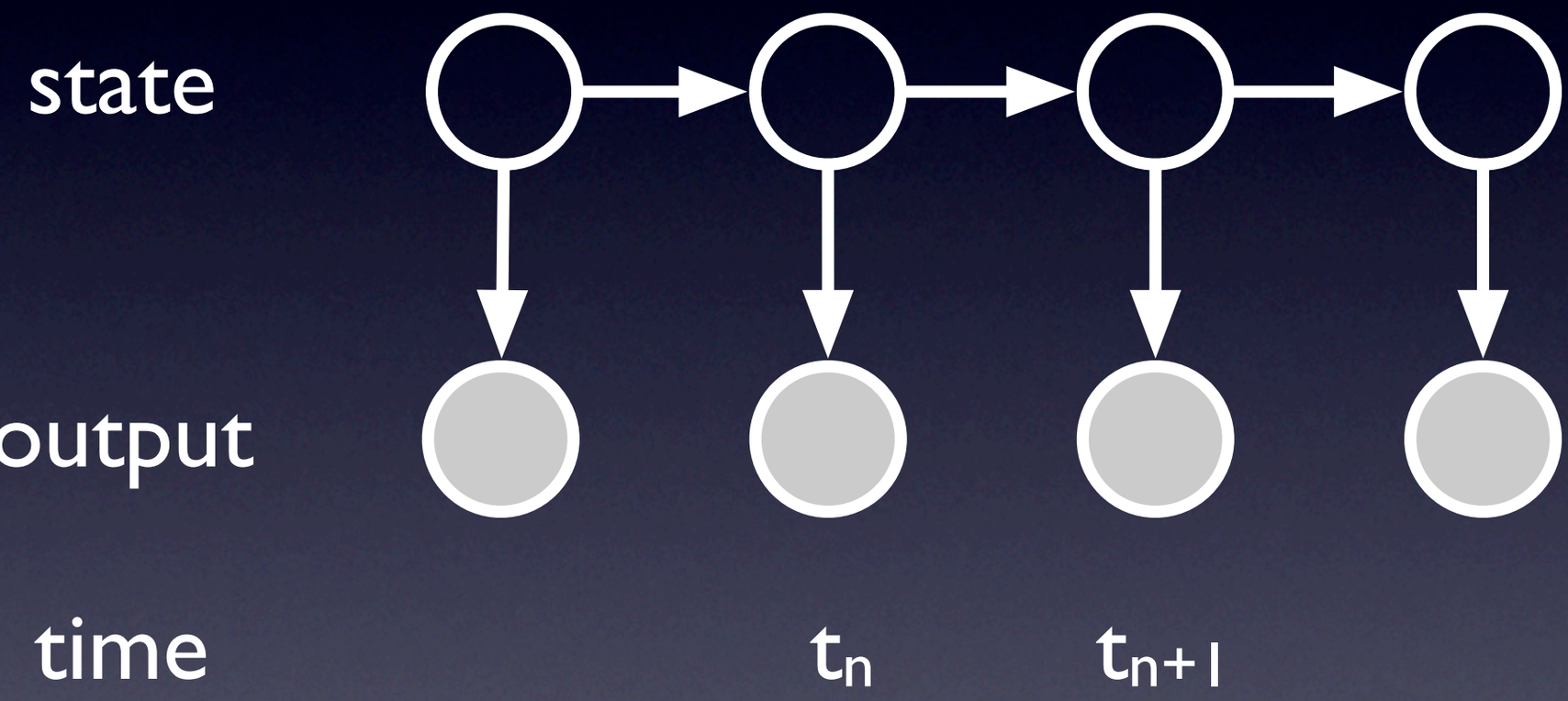
- Mobile devices (T9)
- Dasher (http://www.inference.phy.cam.ac.uk/dasher/)

Applications

- Mobile devices (T9)
- Dasher (http://www.inference.phy.cam.ac.uk/dasher/)

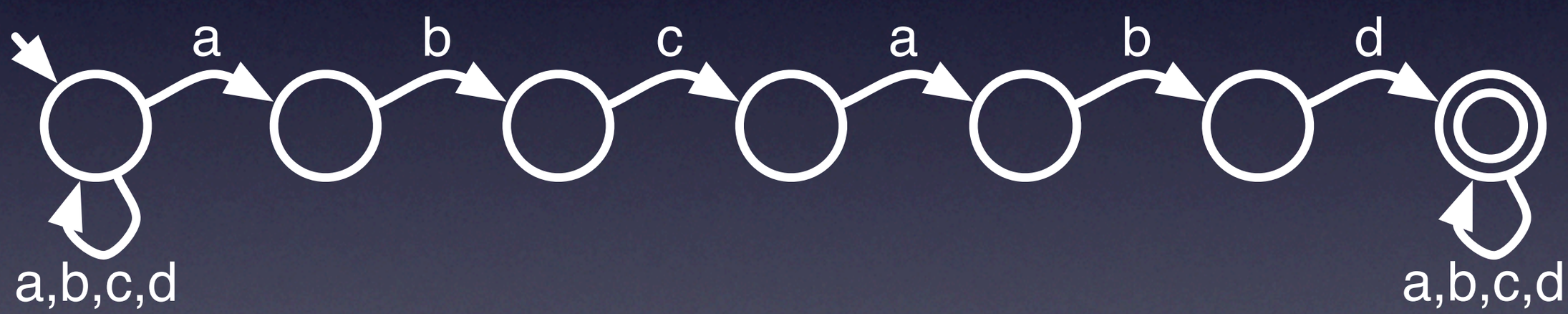


HMM's



String searching

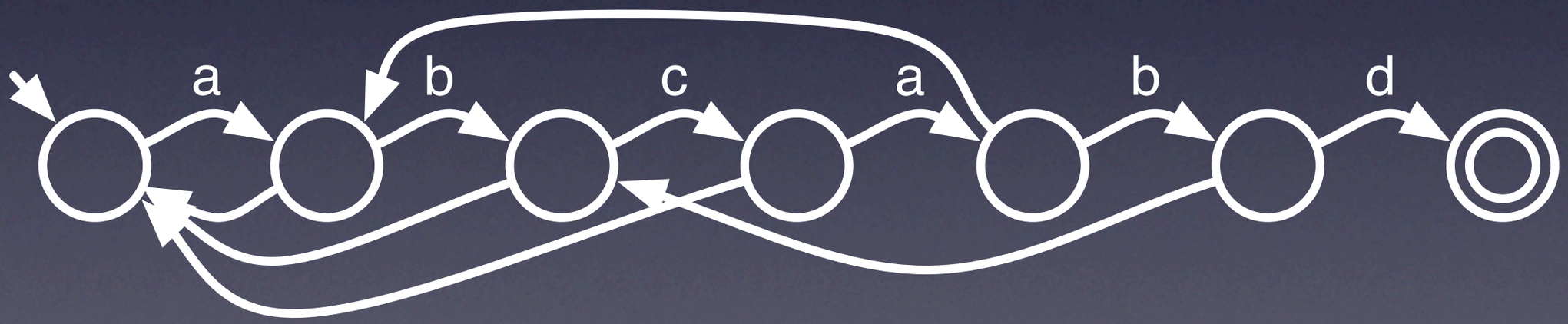
Finite automaton



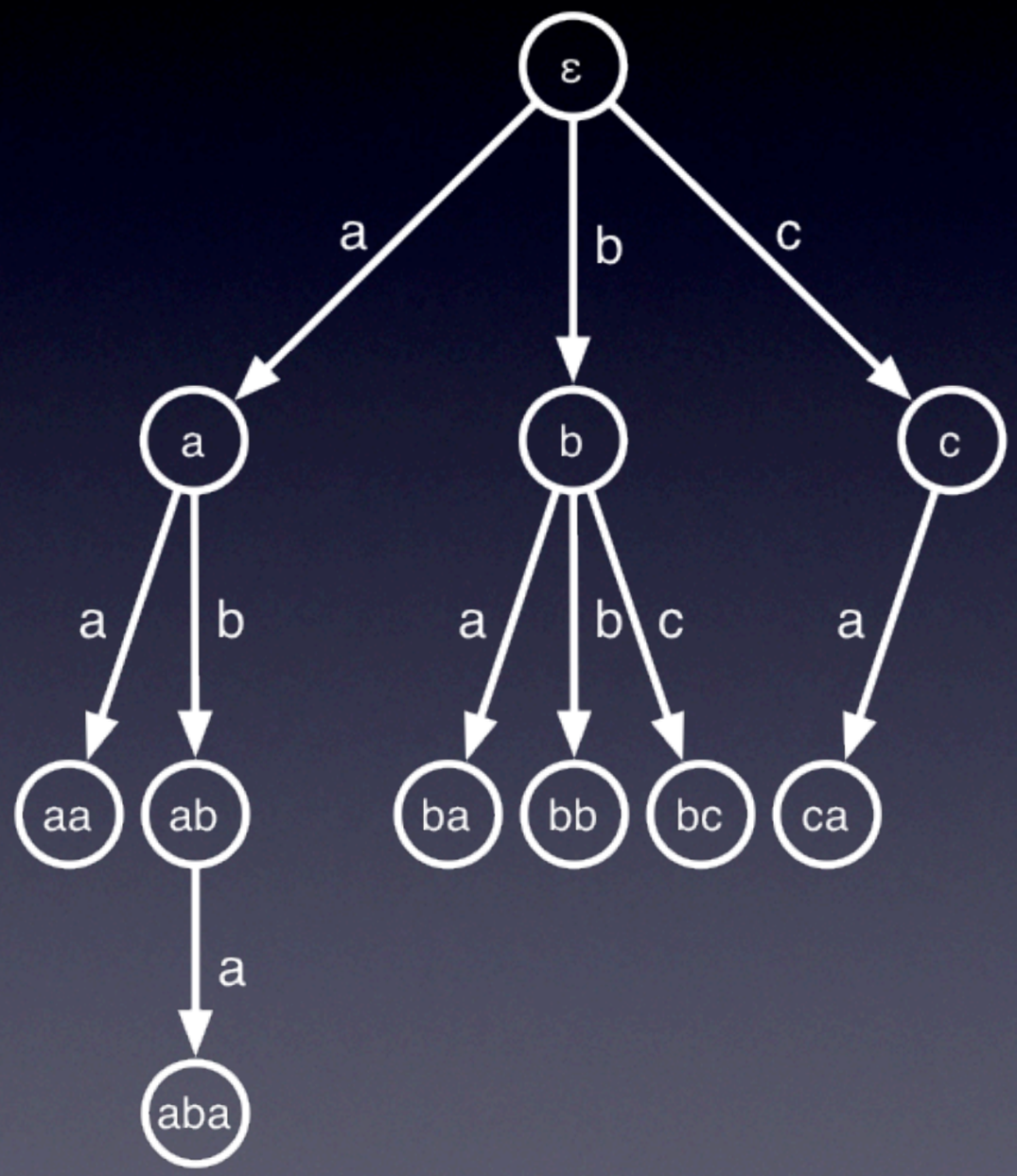
String searching

Knuth-Morris-Pratt

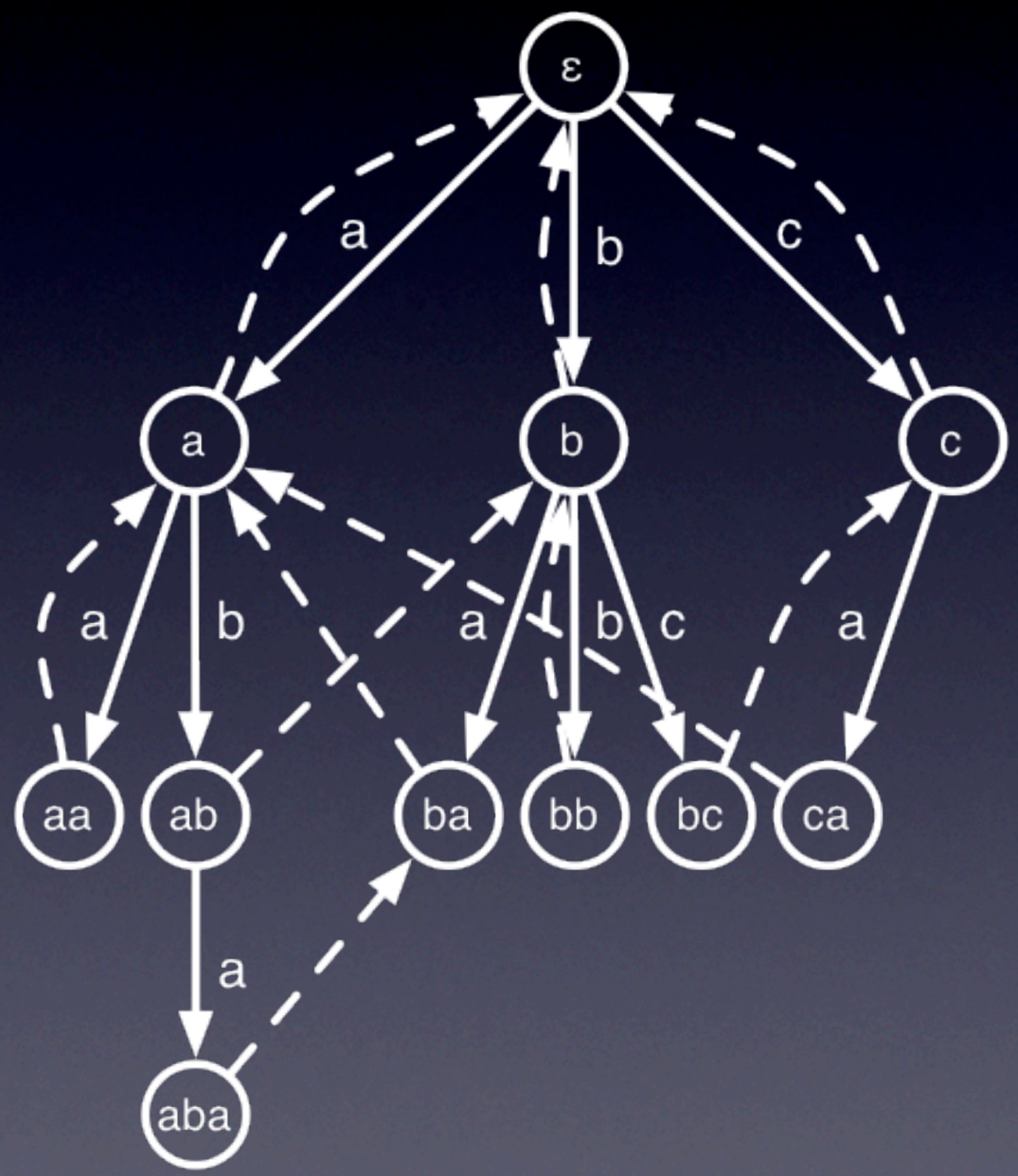
A	B	C	A	B	D
-1	0	0	0	1	2



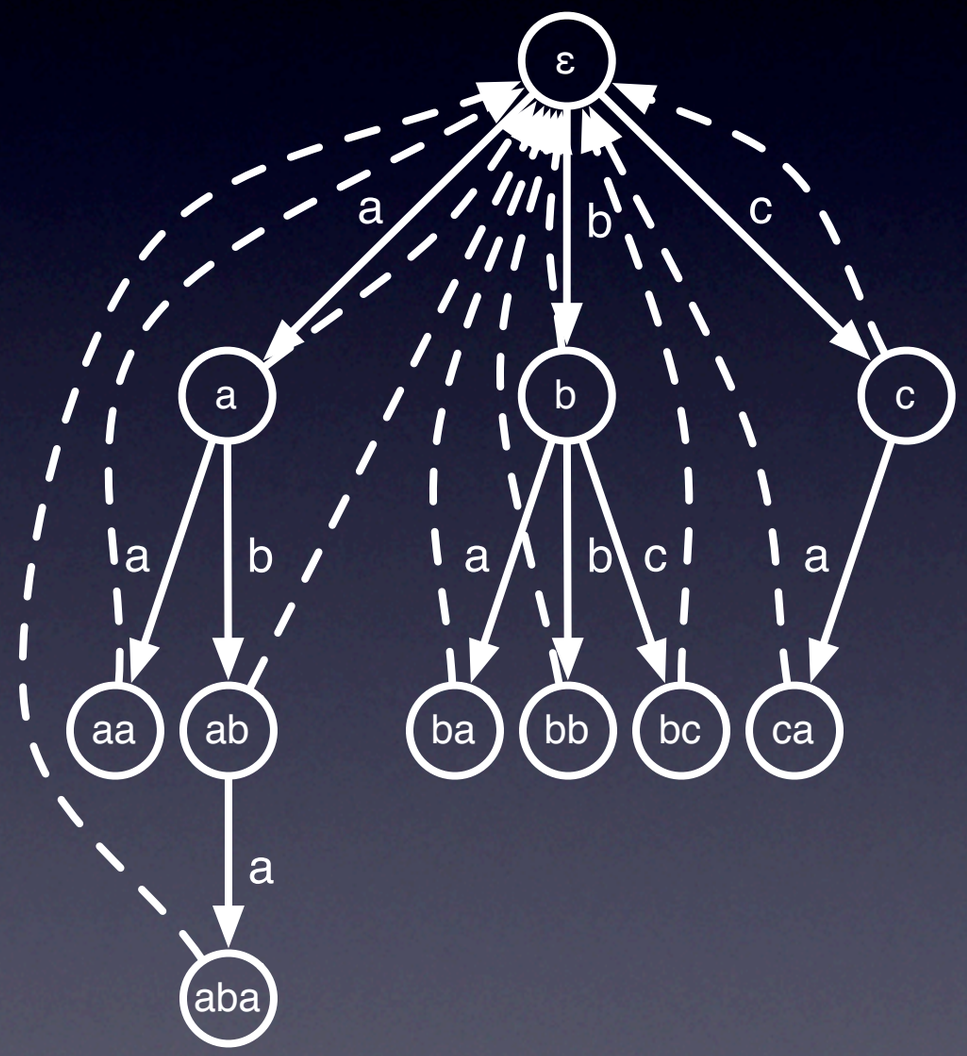
Suffix trie



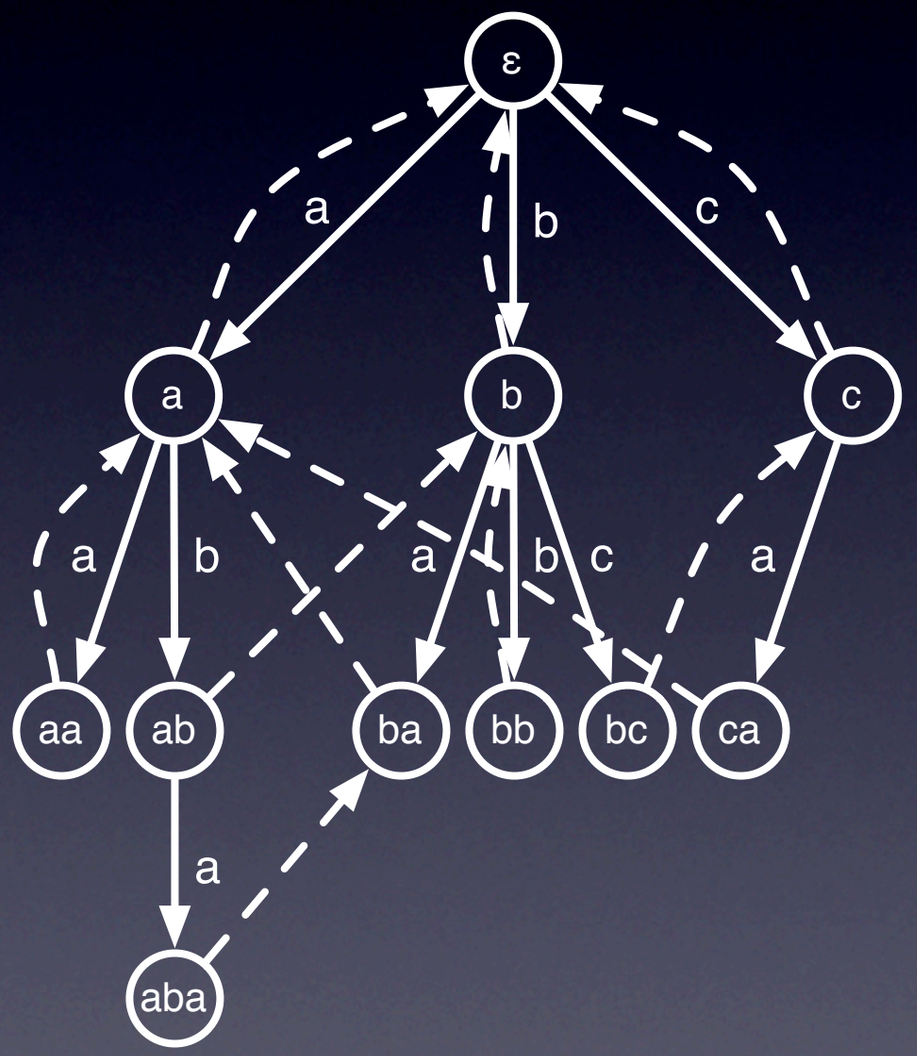
Suffix trie



State model

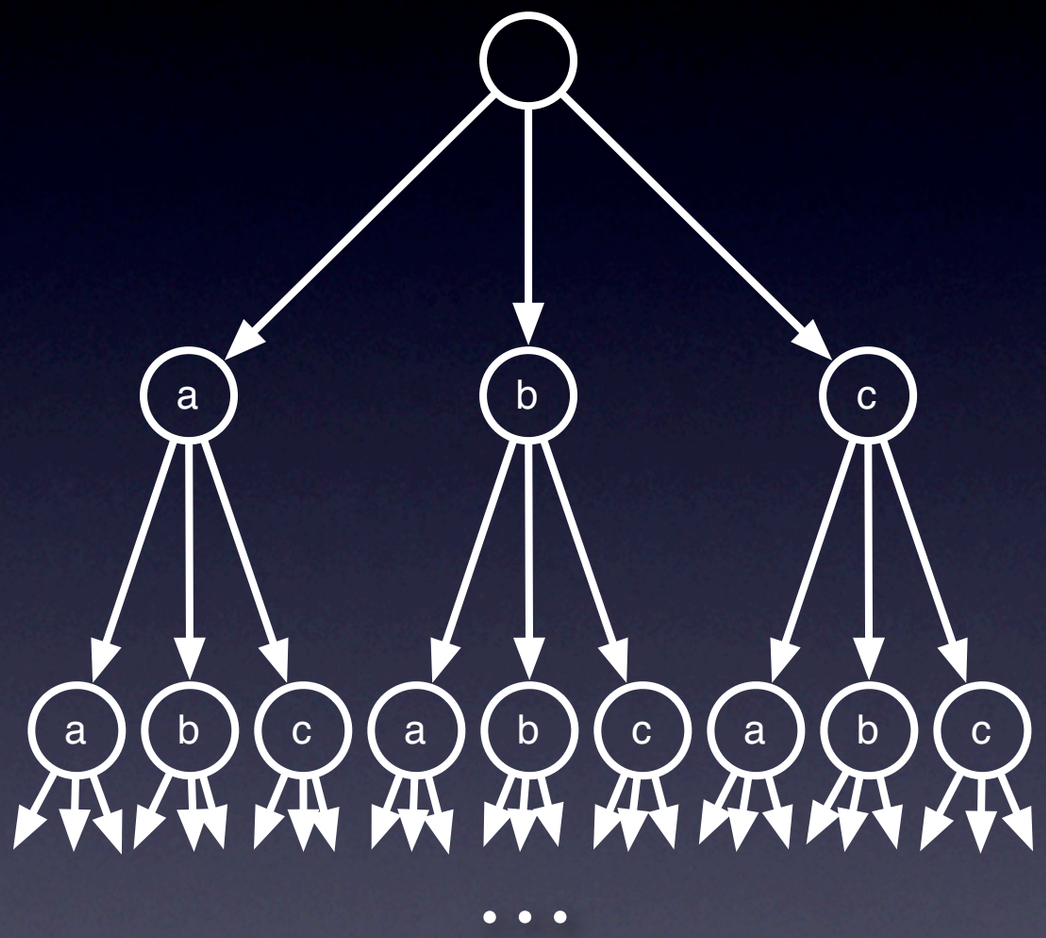


Full backlinks



Partial backlinks

Output model



Space complexity

- $O(n^2 \cdot m)$
  - *n*: word length
  - *m:* number of words
- Less if words share common prefixes
- Suffix *tr*ees can achieve $O(n \cdot m)$, but are not applicable

Inference

Inference

- $O(n^2 \cdot m)$ potential states, but most of them have probability zero

Inference

- $O(n^2 \cdot m)$ potential states, but most of them have probability zero
- Sparse representation

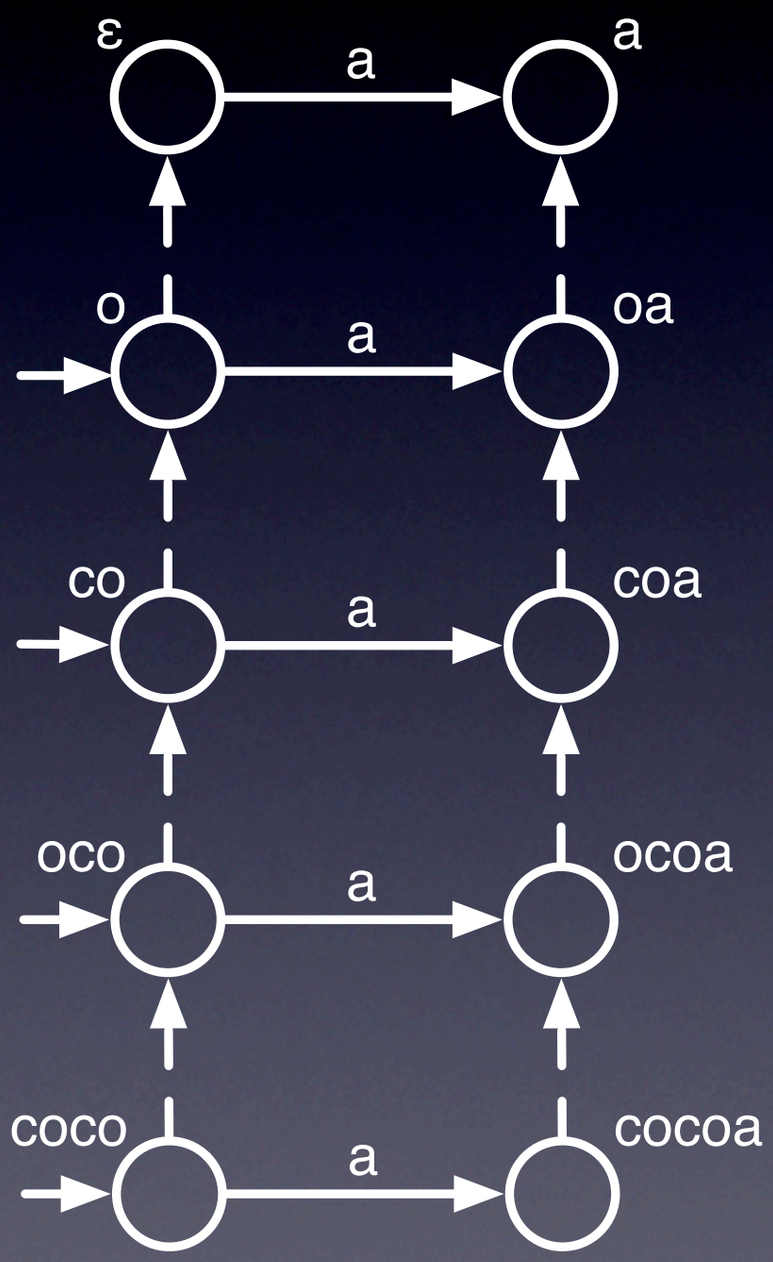
Inference

- $O(n^2 \cdot m)$ potential states, but most of them have probability zero
- Sparse representation
- Possible states are suffixes of word read so far

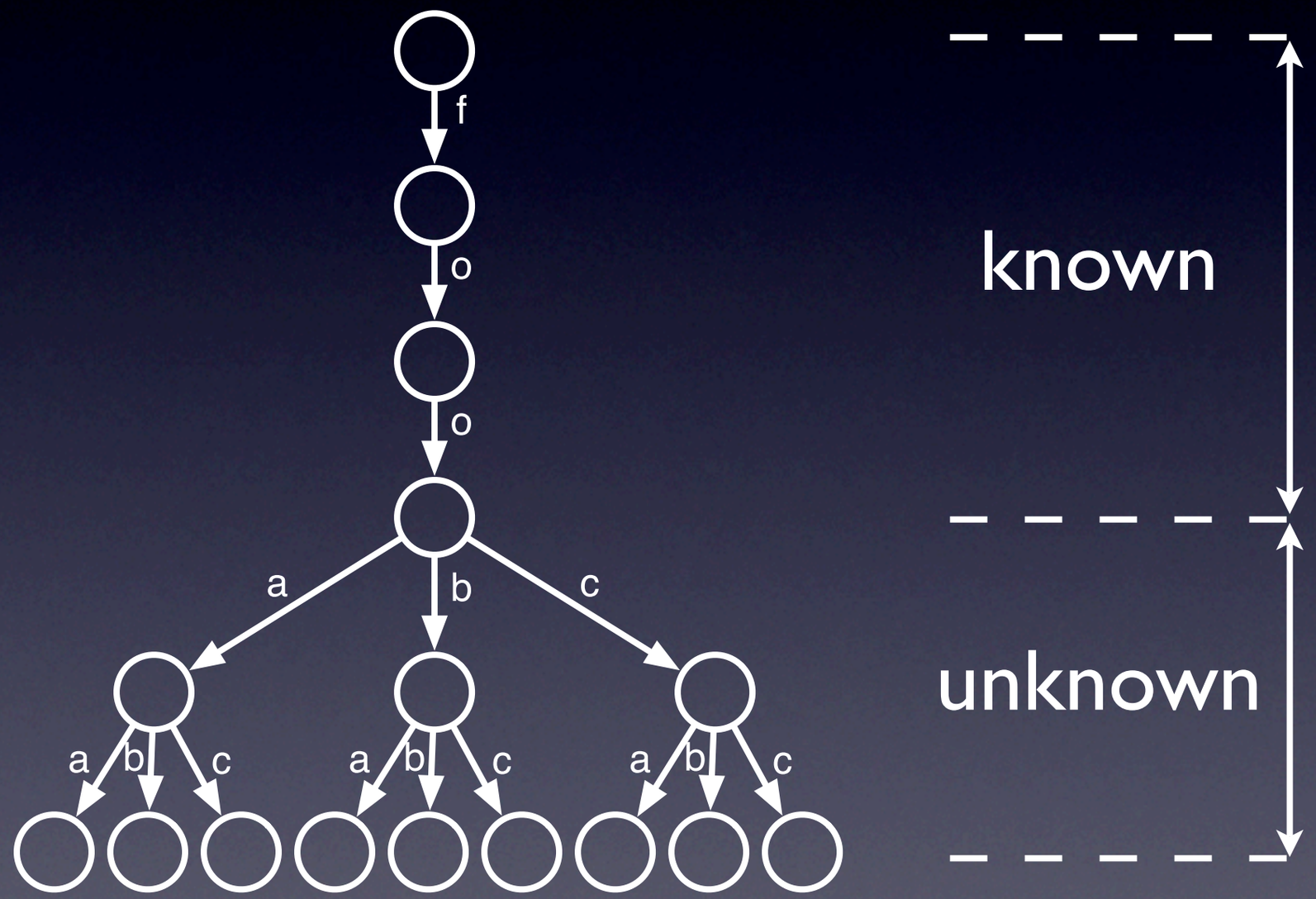
Inference

- $O(n^2 \cdot m)$ potential states, but most of them have probability zero
- Sparse representation
- Possible states are suffixes of word read so far
- Store only longest suffix and list of probabilities

Inference



Word completion



Parameter estimation

Parameter estimation

- Online version of Baum-Welch algorithm:

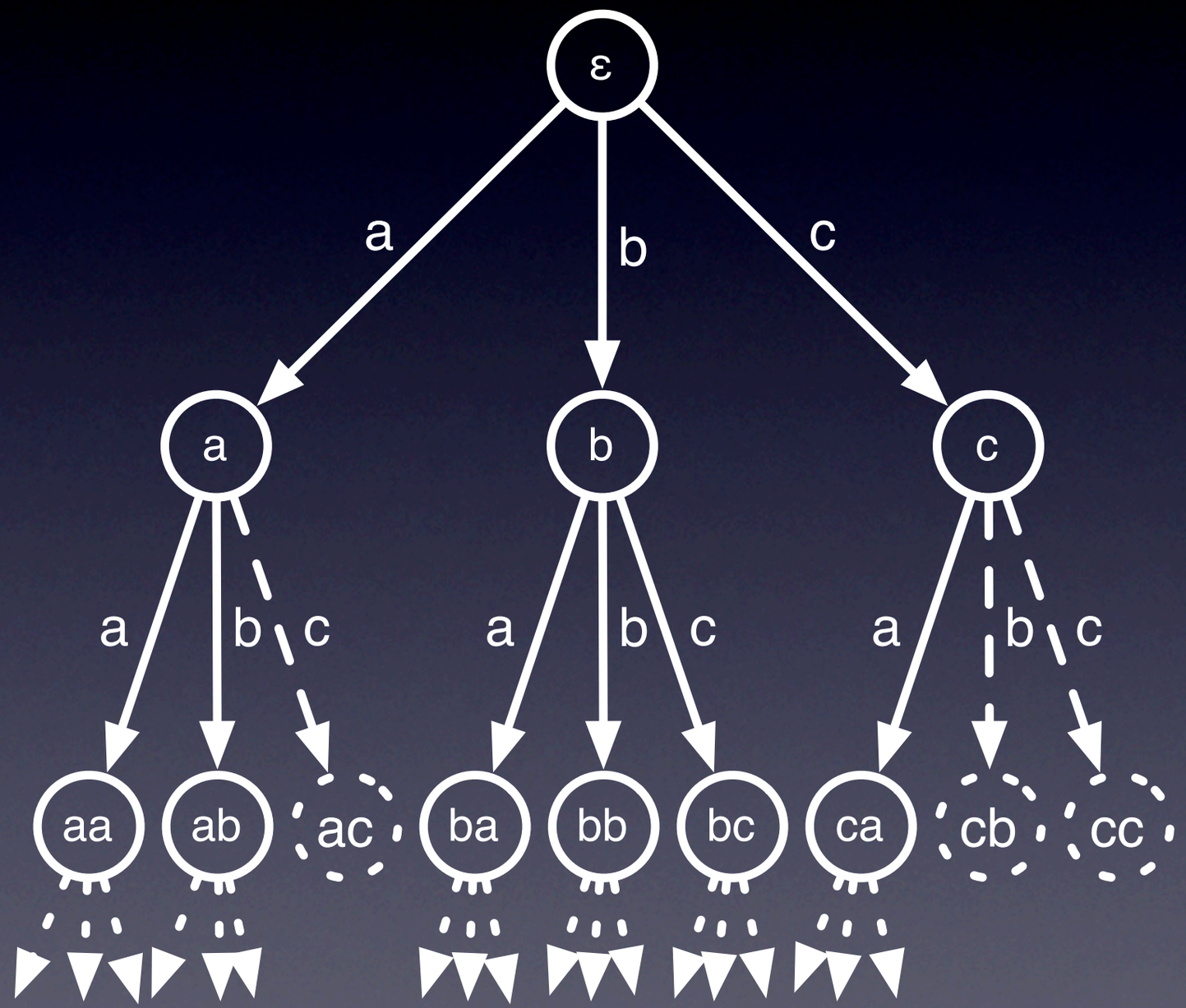
Parameter estimation

- Online version of Baum-Welch algorithm:
  - Baum-Welch algorithm counts estimated transitions

Parameter estimation

- Online version of Baum-Welch algorithm:
  - Baum-Welch algorithm counts estimated transitions
  - Use intermediate transition counts for inference

Suffix trie construction



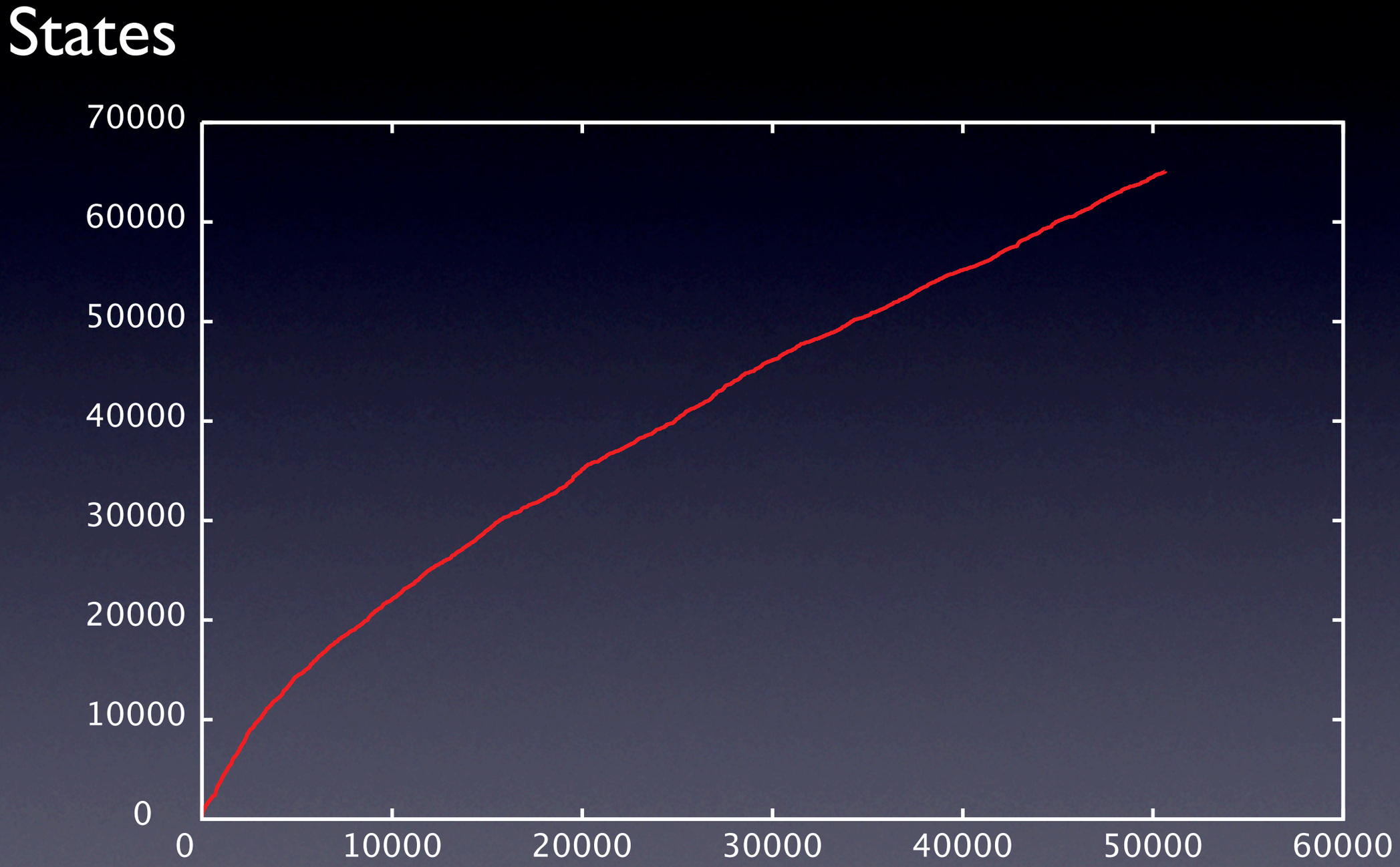
Suffix trie construction

- Problem: probabilities stay zero
  ⇒ Smoothing
- Value: current state probability
- Total transition counts increase by 1
  (like during update-step)
- Uses simple estimation of new values for inference, then updates estimation

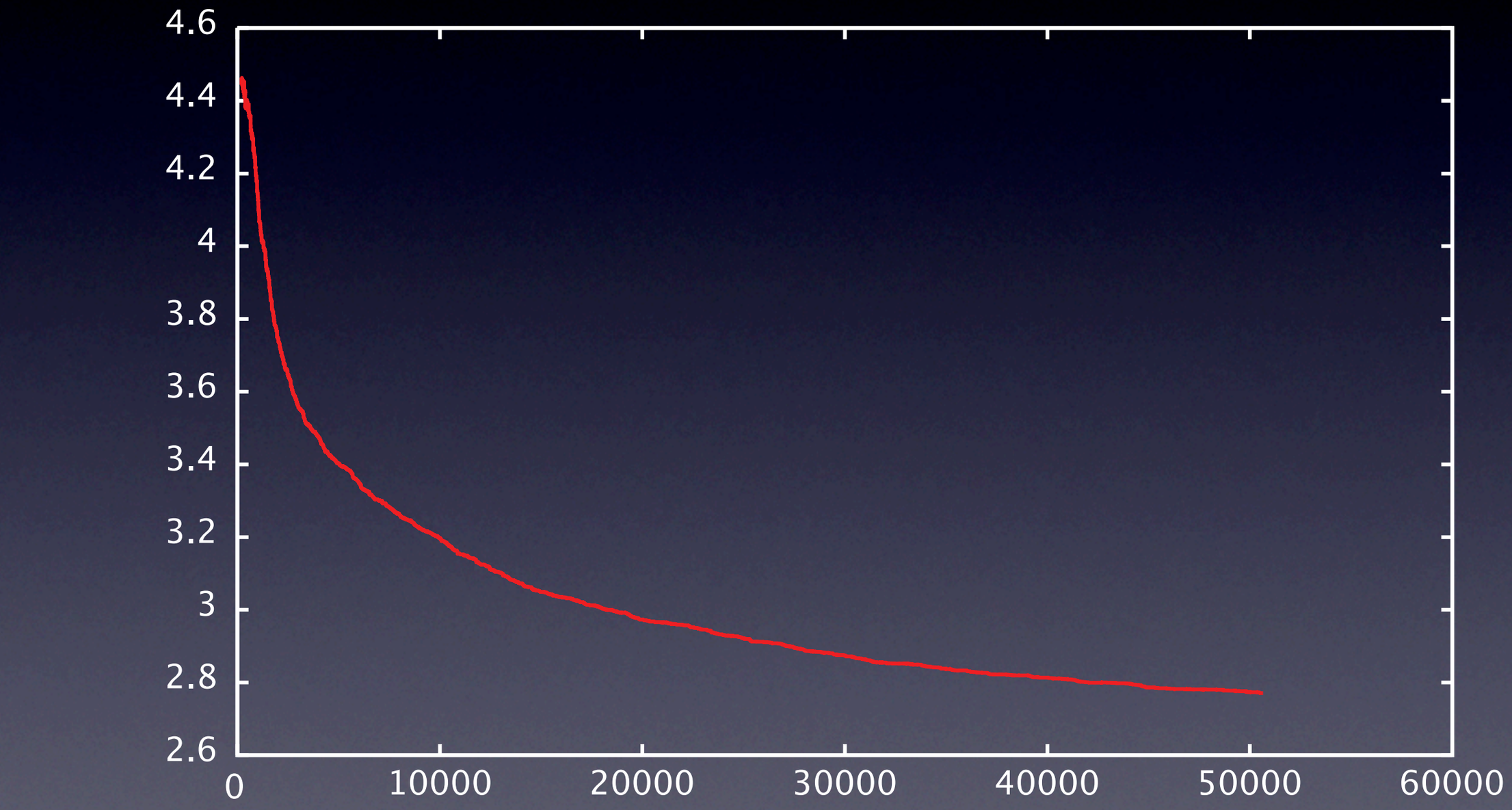
Evaluation

- English text corpus from Dasher, ≈56,000 words, 300 kb
- 90% training, 10% testing
- Measuring number of states and average perplexity of test set

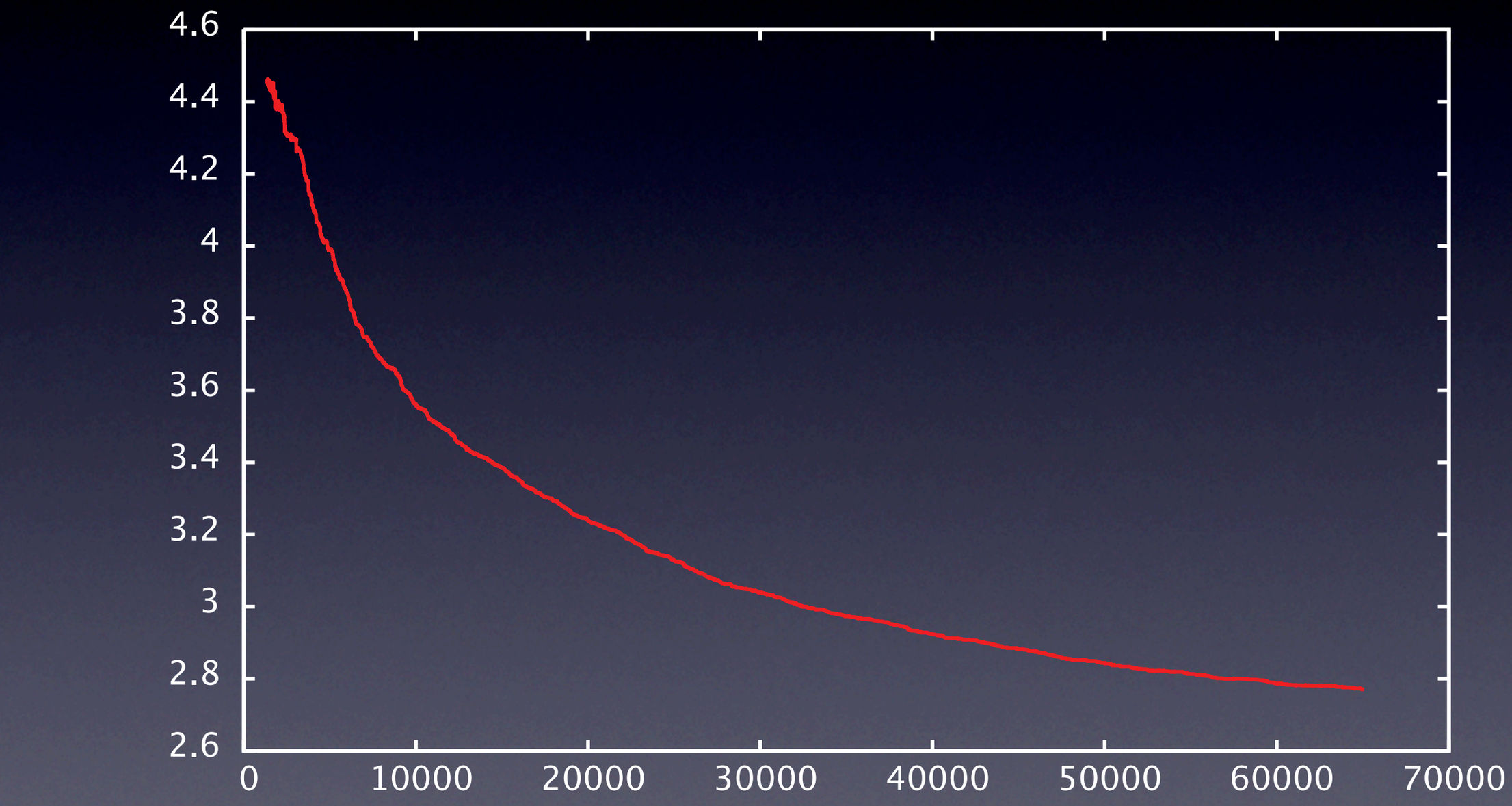
Results


States
70000
60000
50000
40000
30000
20000
10000
0
0
10000
20000
30000
40000
50000
60000
Words learned

Results


Avg. Perplexity
[bits/char]
4.6
4.4
4.2
4
3.8
3.6
3.4
3.2
3
2.8
2.6
0
10000
20000
30000
40000
50000
60000
Words learned

Results


Avg. Perplexity
[bits/char]
4.6
4.4
4.2
4
3.8
3.6
3.4
3.2
3
2.8
2.6
0
10000
20000
30000
40000
50000
60000
70000
States

An example

- T9-like text input:

  http://home.in.tum.de/~bauerb/t9/