Language Models

Generative model

One-state Finite Automaton

Language Model

- If each node has a probability distribution over generating different terms, we have a language model.
- A language model is a function that puts a probability measure over strings drawn from some vocabulary.

- model M over an alphabet $\sum$:
- $\sum P(s) = 1$
  - $s \in \sum^*$
  - Probabilistic Finite Automaton
  - $\sum P(t) = 1$
  - $t \in V$

Example

P(frog said that toad likes frog) =

(0.01 × 0.03 × 0.04 × 0.01 × 0.02 × 0.01)

×(0.8 × 0.8 × 0.8 × 0.8 × 0.8 × 0.8 ×

0.2)

≈ 0.000000000001573

Types of Lang Models

- Unigram
- Bigram
- ....
- Similarly, there can be higher n-gram models
- Unigram is experimentally

found to work good for IR

Language Model

- Multinomial Distribution

. L_d = length of doc in #terms

Query Likelihood Model

- We need

- We use

- where

Estimating Probabilities

- Maximum Likelihood Estimate

- Smoothing if $tf_{t,d} = 0$, then

- Linear Internpolation

- where $0 < \lambda < 1$ and M_c is the model built from entire document collection.

- Ranking is done using

Bayesian Smoothing

- Small Lambda or large alpha means more smoothing
- Low smoothing good for short queries, high smoothing good for long queries

Example

Suppose the query is *revenue down*.

Soln.