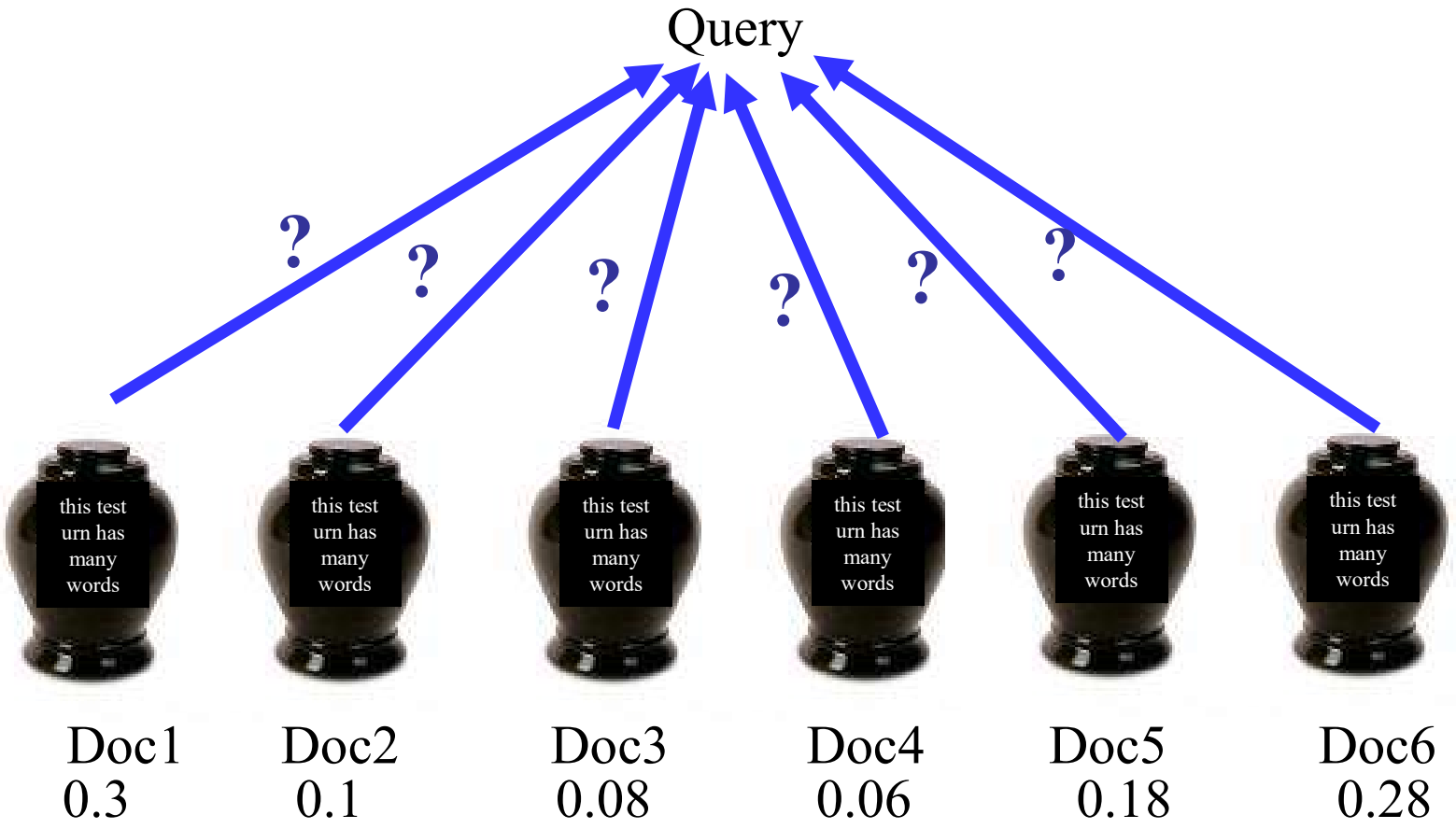
# Probabilistic Language-Model Based Document Retrieval

# Naïve Bayes for Retrieval

- Naïve Bayes can also be used for ad-hoc document retrieval.
- Treat each of the *n* documents as a category with only one training example, the document itself.
- Classify queries using this *n*-way categorization.
- Rank documents based on the posterior probability of their category.
- Effectively using Naïve Bayes as a simple1-gram language model for each document.

# Generative Model for Retrieval




**Ranked Retrievals:**

Doc 1 0.3
Doc 6 0.28
Doc 5 0.18
Doc 2 0.1
Doc 3 0.08
Doc 4 0.06

# Smoothing

- Proper smoothing is important for this approach to work well.
- Laplace smoothing does not work well for this application.
- Better to use ***linear interpolation*** for smoothing.

# Linear Interpolation Smoothing

- Estimate conditional probabilities P($X_i$ | $Y$) as a mixture of conditioned and unconditioned estimates:

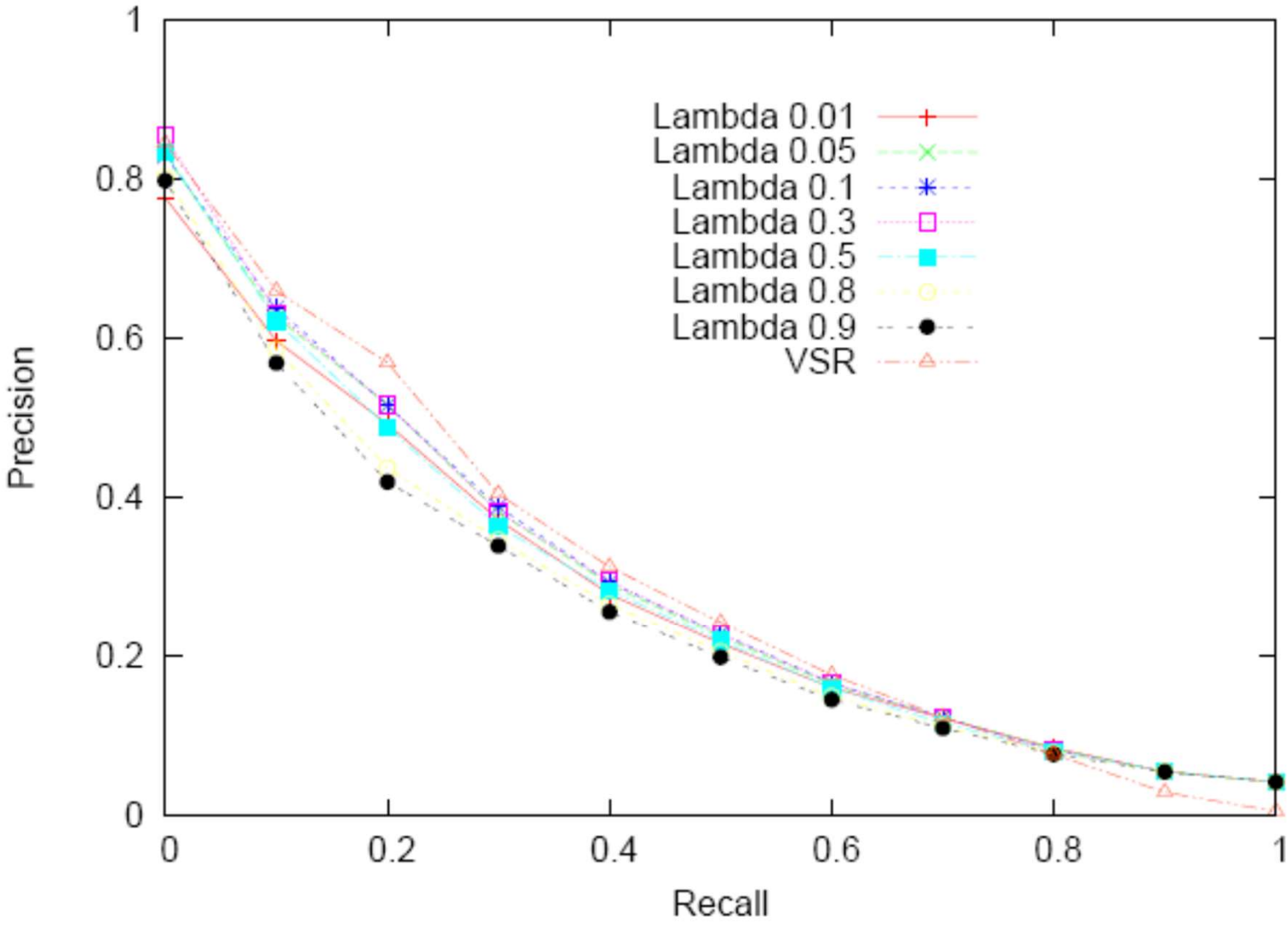
$$P(X_i \mid Y) = \lambda \hat{P}(X_i \mid Y) + (1 - \lambda)\hat{P}(X_i)$$

- $\hat{P}(X_i \mid Y)$ is the probability of drawing word $X_i$ from the urn of words in category (i.e. document) $Y$.
- $\hat{P}(X_i)$ is the probability of drawing word $X_i$ from the urn of words in the entire corpus (i.e. all document urns combined into one big urn).

# Amount of Smoothing

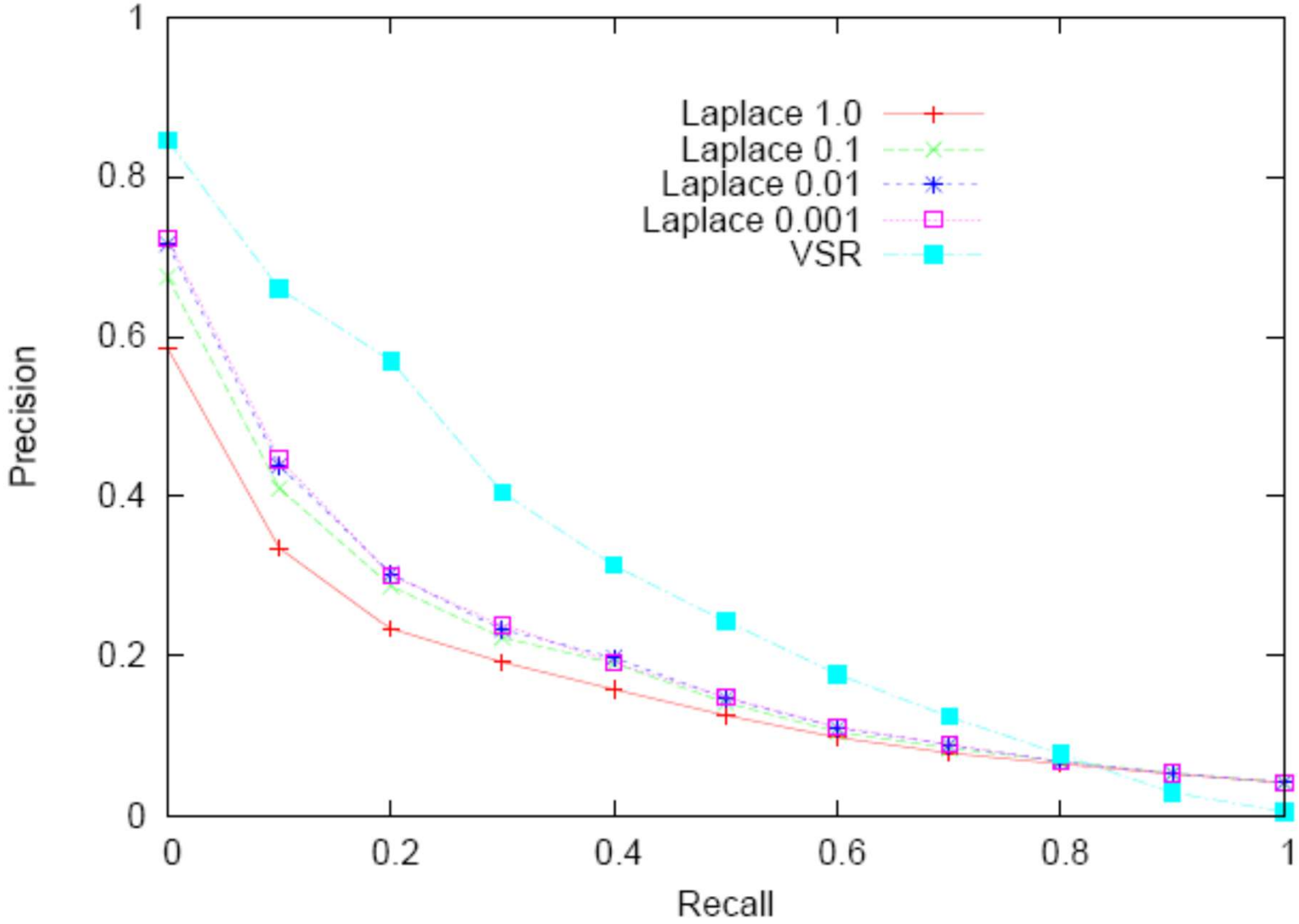
- Value of $\lambda$ controls the amount of smoothing.
- The lower $\lambda$ is, the more smoothing there is since the unconditioned term is weighted higher $(1 - \lambda)$.
- Setting $\lambda$ properly is important for good performance.
- Set $\lambda$ manually or automatically based on maximizing performance on a development set of queries.
- Lower $\lambda$ tends to work better for long queries, high $\lambda$ for short queries.

# Experimental Results on CF Corpus
# Effect of λ Parameter

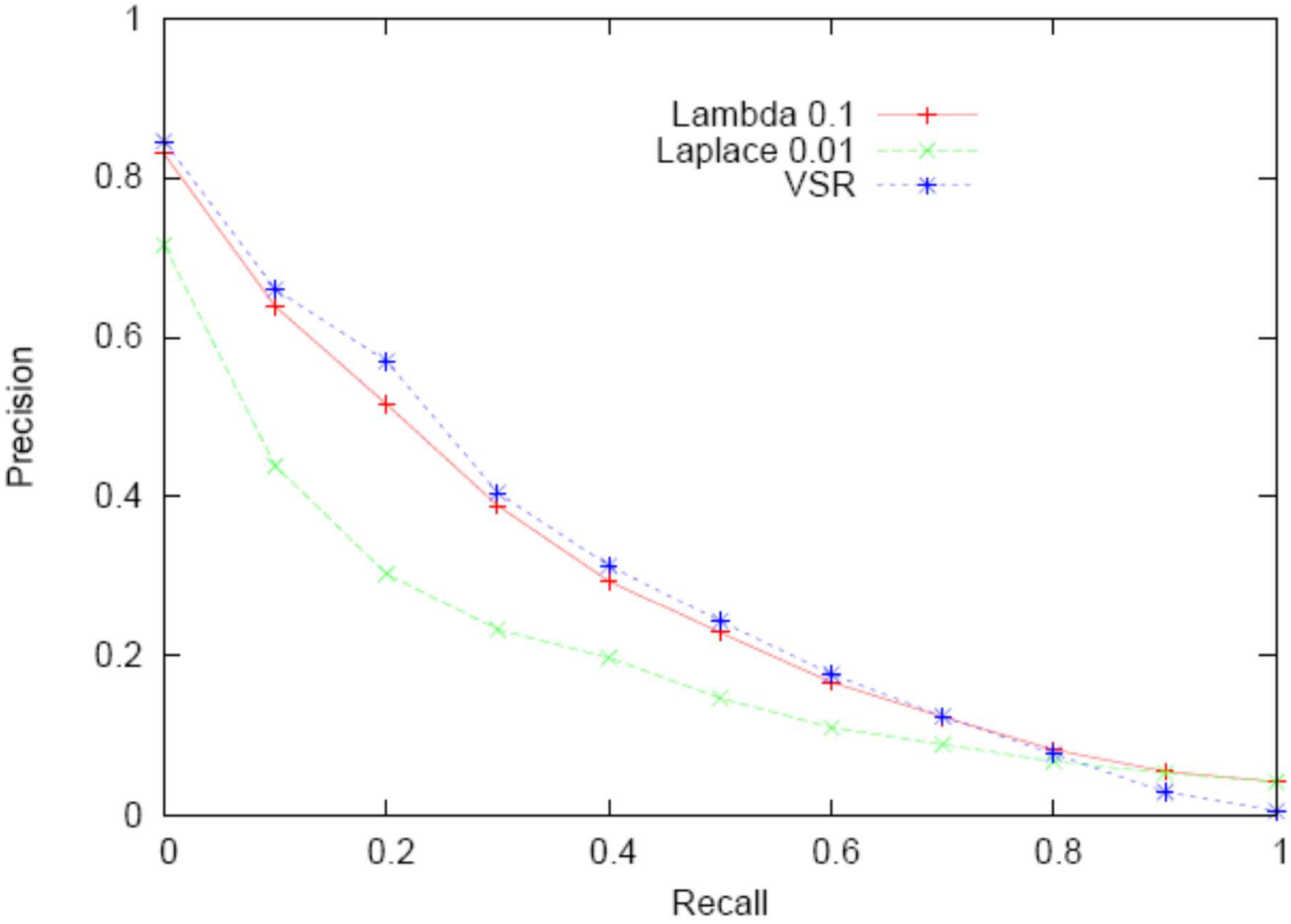



Given long queries, large amount of smoothing (λ=0.1) seems to work best

# Experimental Results on CF Corpus
# Effect of Laplace Smoothing Parameter



# Experimental Results on CF Corpus
# Comparison of Smothing Methods and VSR




Laplace smoothing does much worse.

Linear interp does about the same as vector-space

# Performance of Language Model Approach

- Larger scale TREC experiments demonstrate that the LM approach with proper smoothing works slightly better than a well-tuned vector-space approach.
- Need to make LM approach efficient by exploiting inverted index.
  - Don't bother to the compute probability of documents that do not contain *any* of the query words.