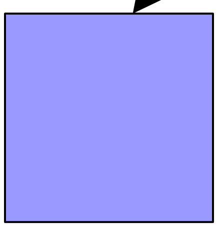
Content-Based Image Retrieval

Konstantin Shkurko
Cornell U.

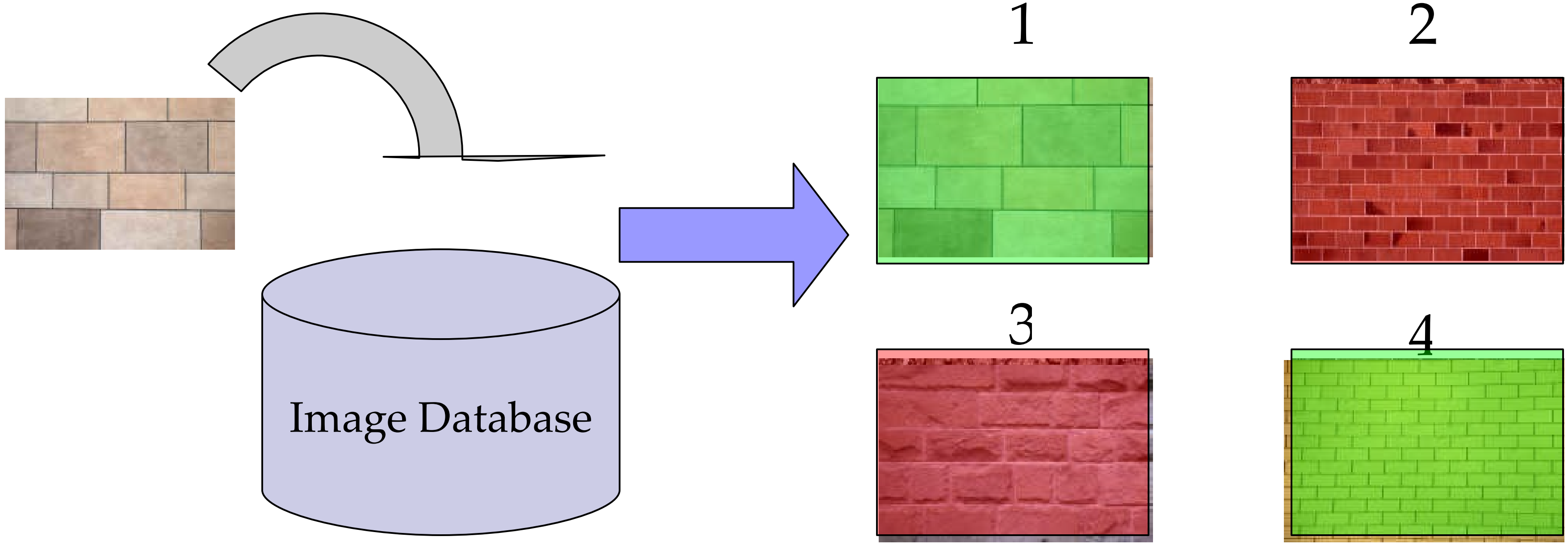
Introduction

- Previous Systems
  - started it all
  - matched colors and textures



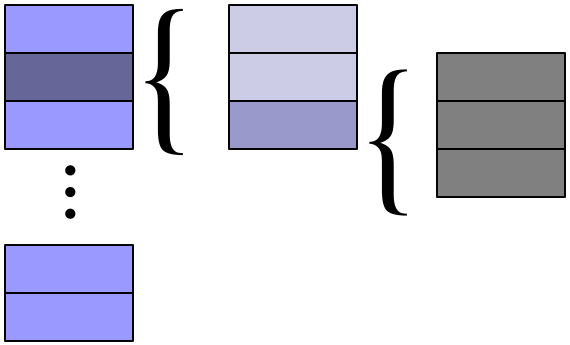
  - worked only ok
  - fail for large databases

- Relevance Feedback saves the day!
  - new approach
  - able to learn what user's looking for
  - update results for better performance



Features

- Color:
  - 18x3x3 HSV Color Histogram

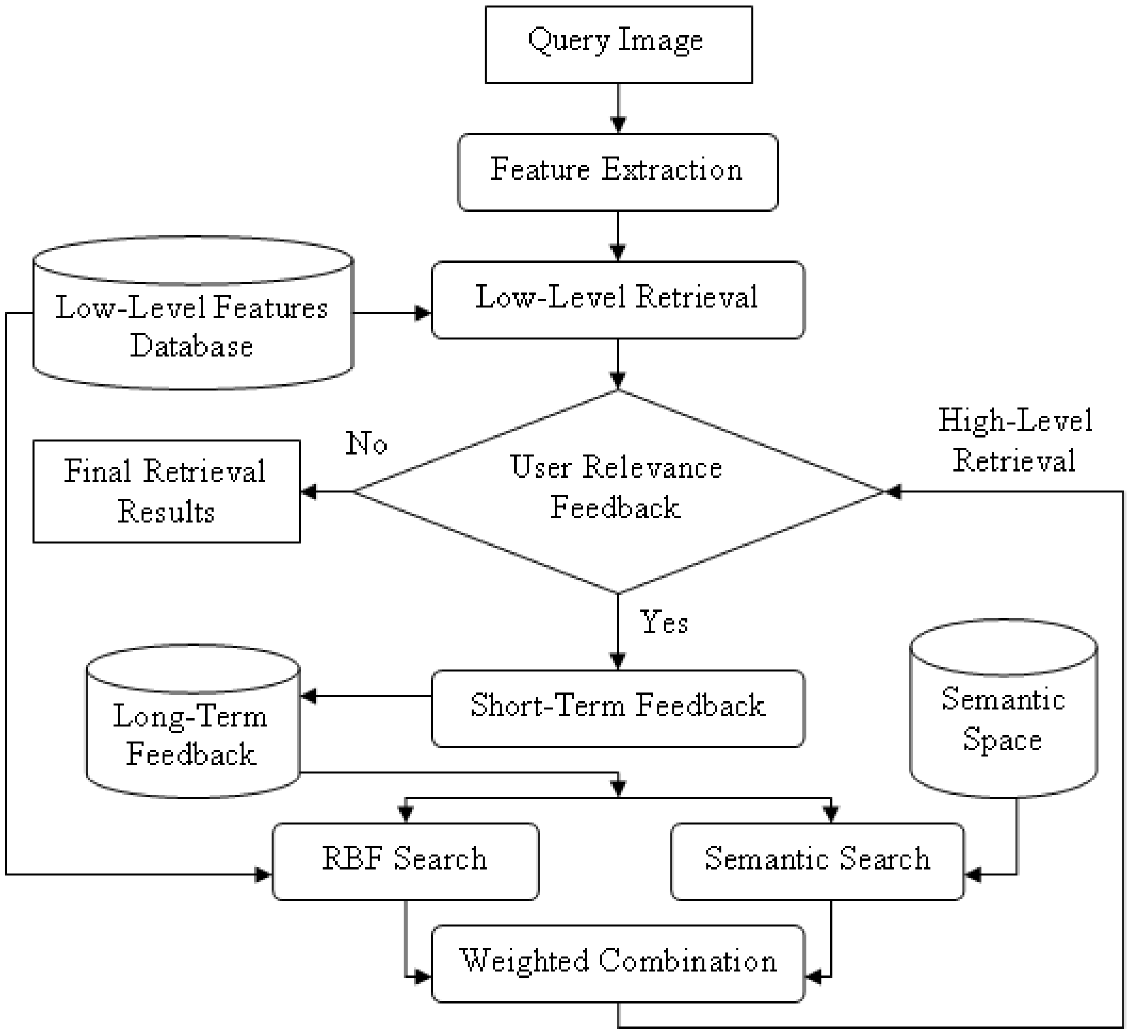


  - RGB Color Moments
    - average color
    - Standard deviation
    - “skew”

- Texture
  - 80 bin MPEG-7 Edge Histogram
    - Matt Went over it in Detail
  - no semi or global

$$\begin{array}{ccccc}
\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} &
\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} &
\begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix} &
\begin{bmatrix} 0 & \sqrt{2} \\ -\sqrt{2} & 0 \end{bmatrix} &
\begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} \\
\text{vertical} & \text{horizontal} & 45^{\circ} & 135^{\circ} & \text{non-directional}
\end{array}$$

Overall Method



Initial Retrieval

- color
  - moments = Euclidean Distance
  - HSV = Histogram Intersection
- texture
  - MPEG-7 = Histogram Intersection

Euclidean Distance

$$S_x^{Euclid} = \frac{1}{\sqrt{\sum_{i=1}^{n} (x_i - q_i)^2}}$$

Histogram Intersection

$$S_x^{Hist} = \sum_{i=1}^{n} \min(x_i^{Hist}, q_i^{Hist})$$

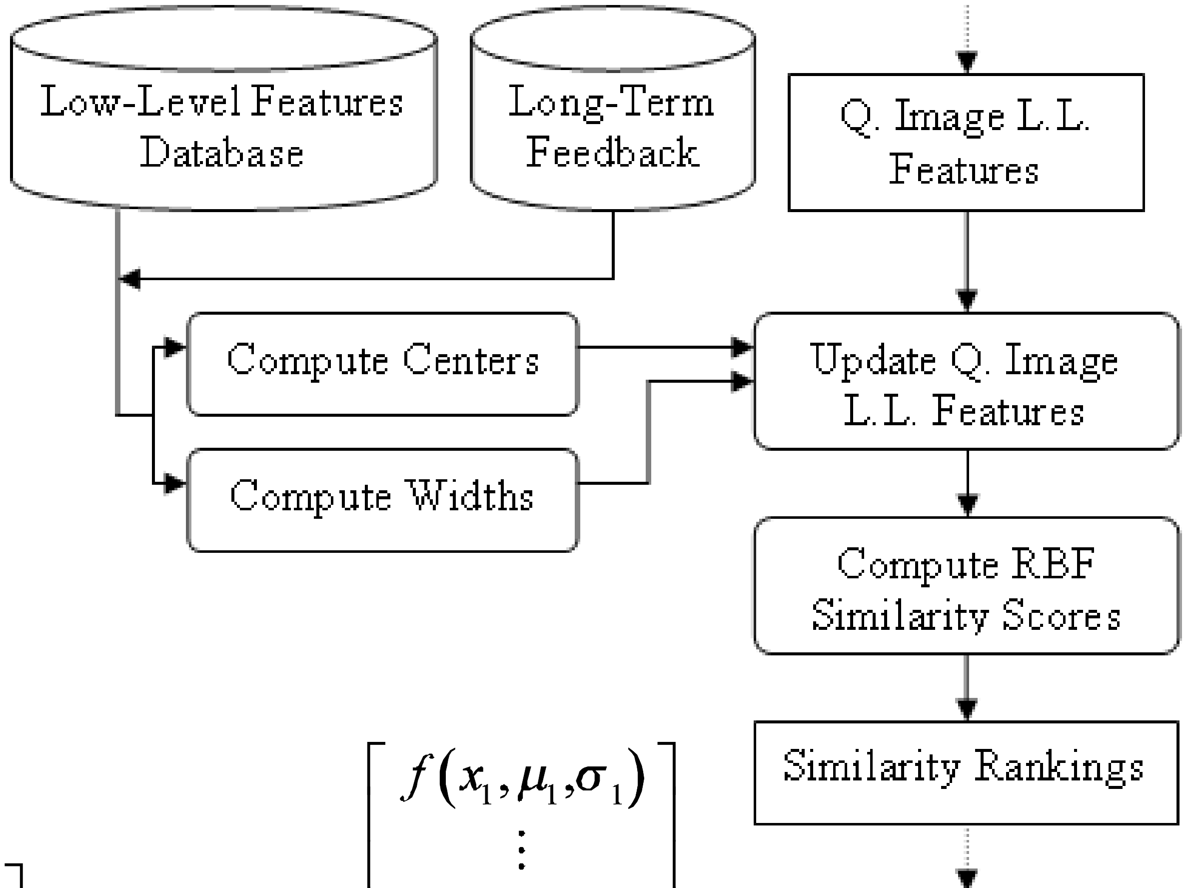
- similarity calculation

$$S_x = S_x^{Hist} + S_x^{Euclid}$$

- get similarities for all images in DB
- return top 30 results to user
- gather feedback -> Short-Term
- store feedback -> Long-Term

Radial Basis Functions

- quick learning
- formula: $$f(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$
- need 2 parameters
  - μ – centers
  - σ – widths
- RF adjusts parameters and recalculates relevance $S_x = \left\| \vec{f}\left(\vec{x},\vec{\mu},\vec{\sigma}\right) \right\|_2$



$$q^{RBF} = \begin{bmatrix} \mu_1 & \cdots & \mu_i & \cdots & \mu_n \\ \sigma_1 & \cdots & \sigma_i & \cdots & \sigma_n \end{bmatrix} \rightarrow \vec{f}\left(\vec{x}, q^{RBF}\right) = \begin{bmatrix} f\left(x_1, \mu_1, \sigma_1\right) \\ \vdots \\ f\left(x_i, \mu_i, \sigma_i\right) \\ \vdots \\ f\left(x_n, \mu_n, \sigma_n\right) \end{bmatrix}$$

Semantic Space

- Matt talked about building it

Semantic Basis Images

Database Images

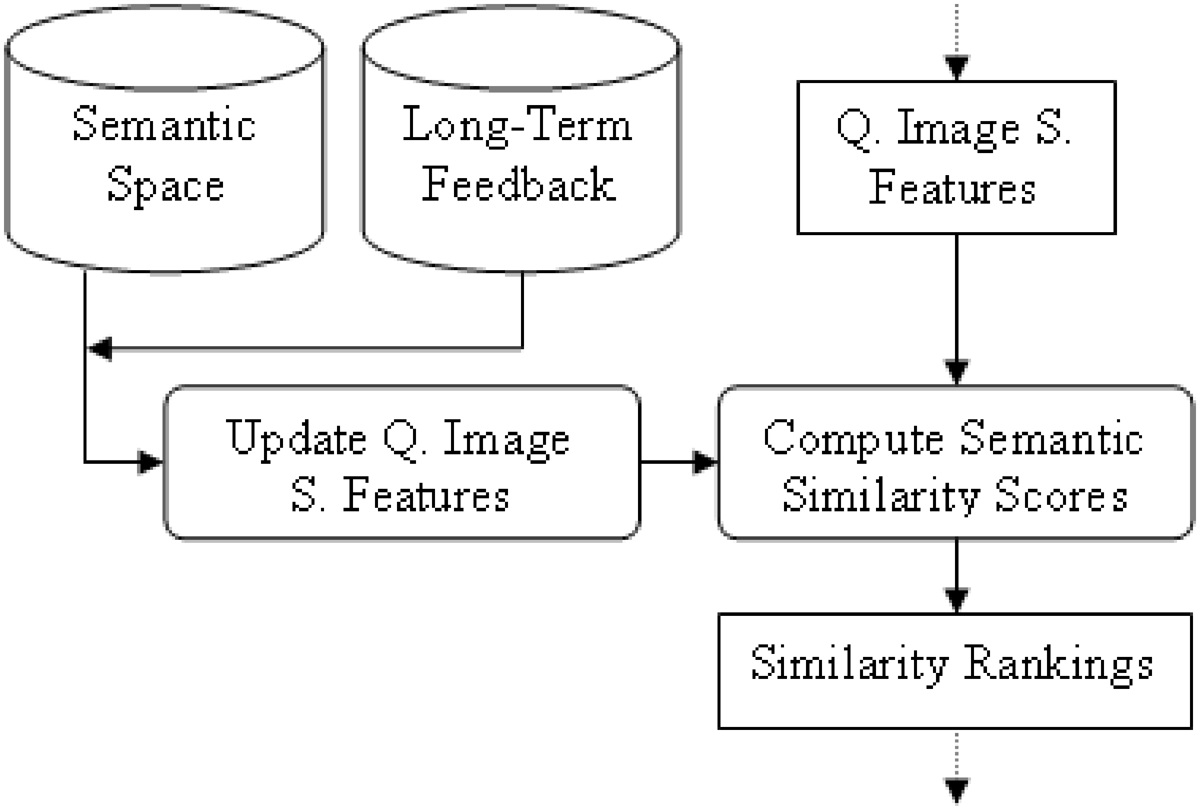
1	0	0	0	-1
-1	0	0	1	0
0	1	-1	-1	0

Semantic Feature Row

relevant

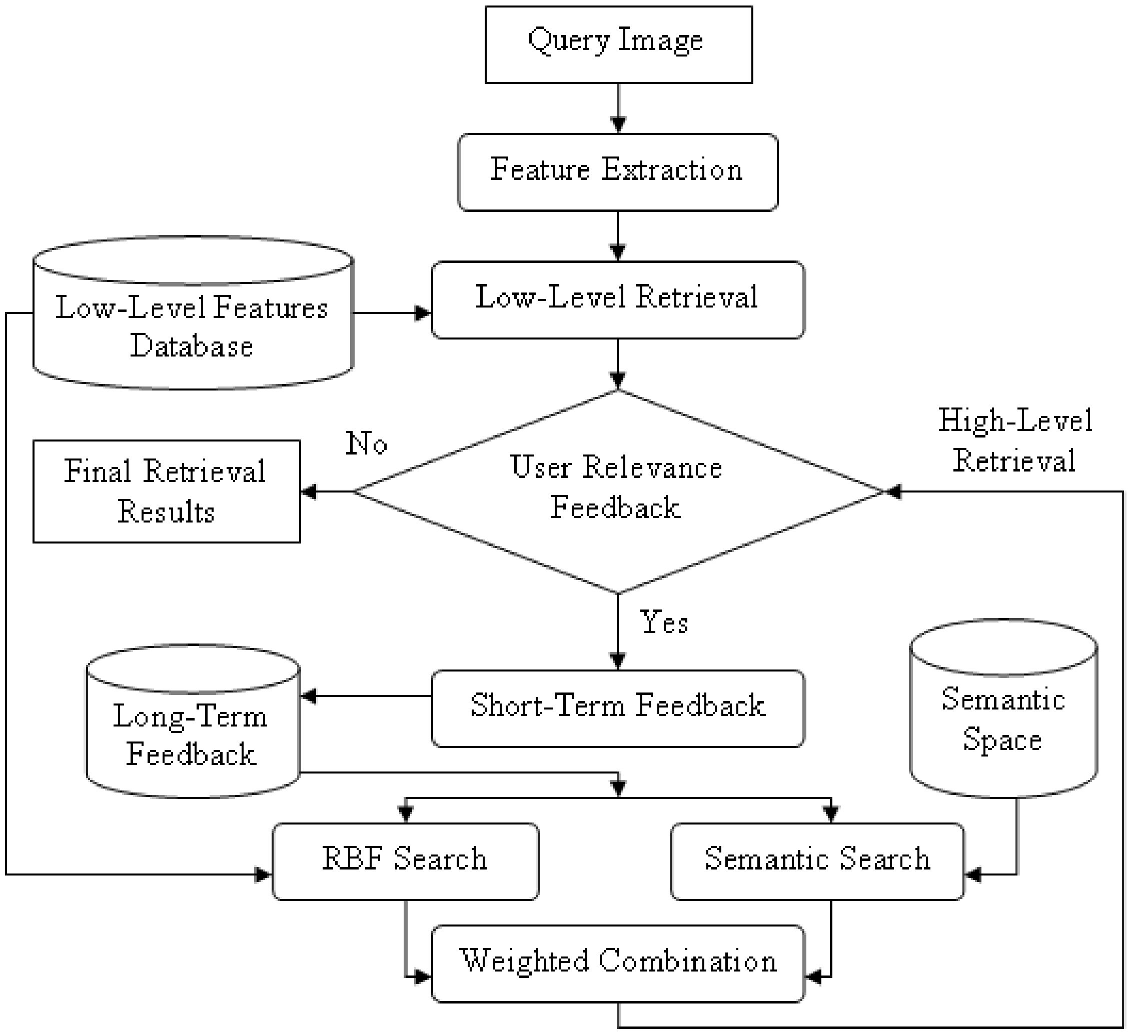
non-relevant

never picked

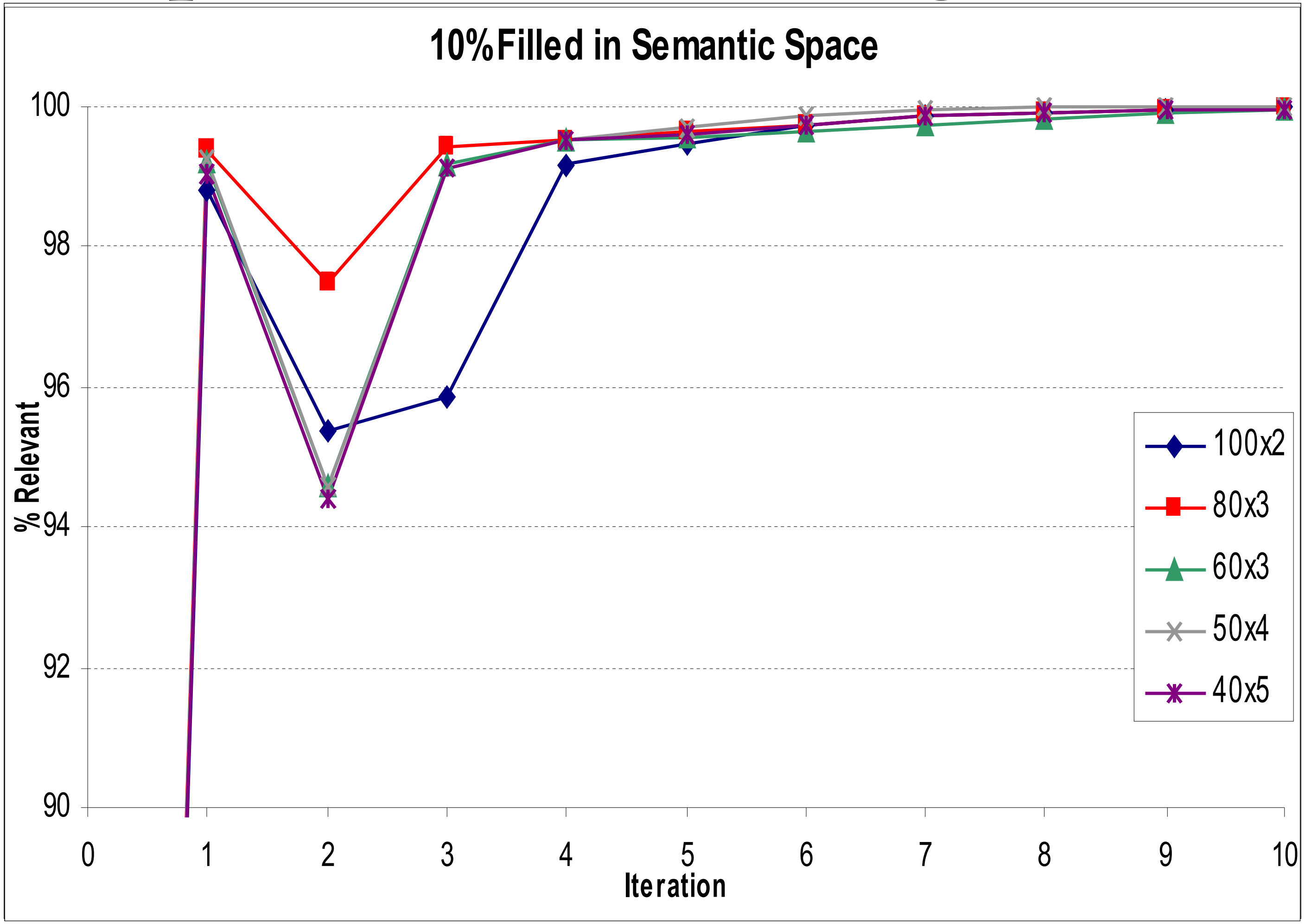


$$q^{Sem} = \left(x_1^R \vee \ldots \vee x_i^R \vee \ldots \vee x_r^R\right) \wedge \overline{\left(x_1^N \vee \ldots \vee x_j^N \vee \ldots \vee x_n^N\right)}$$

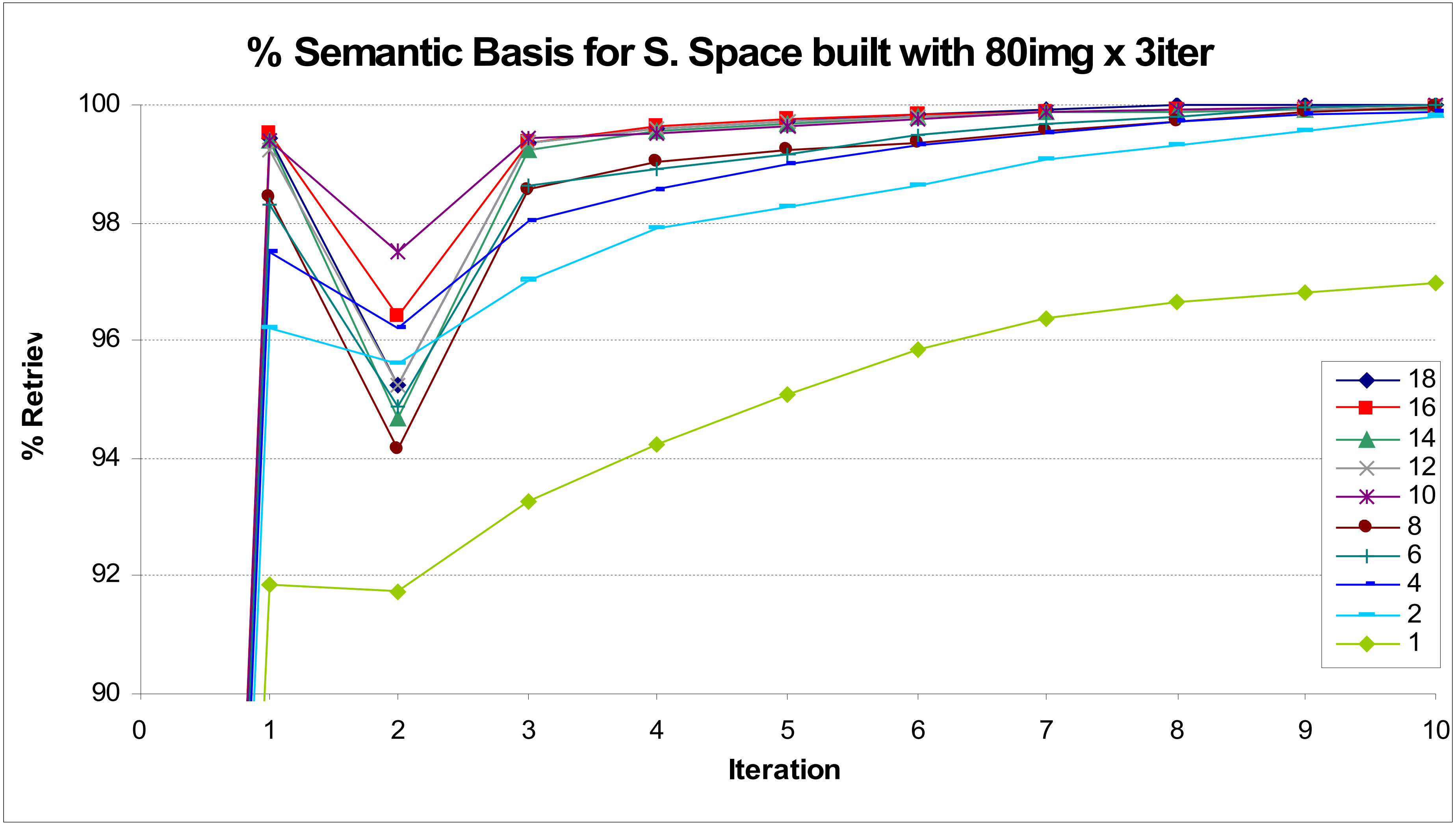
Overall Method



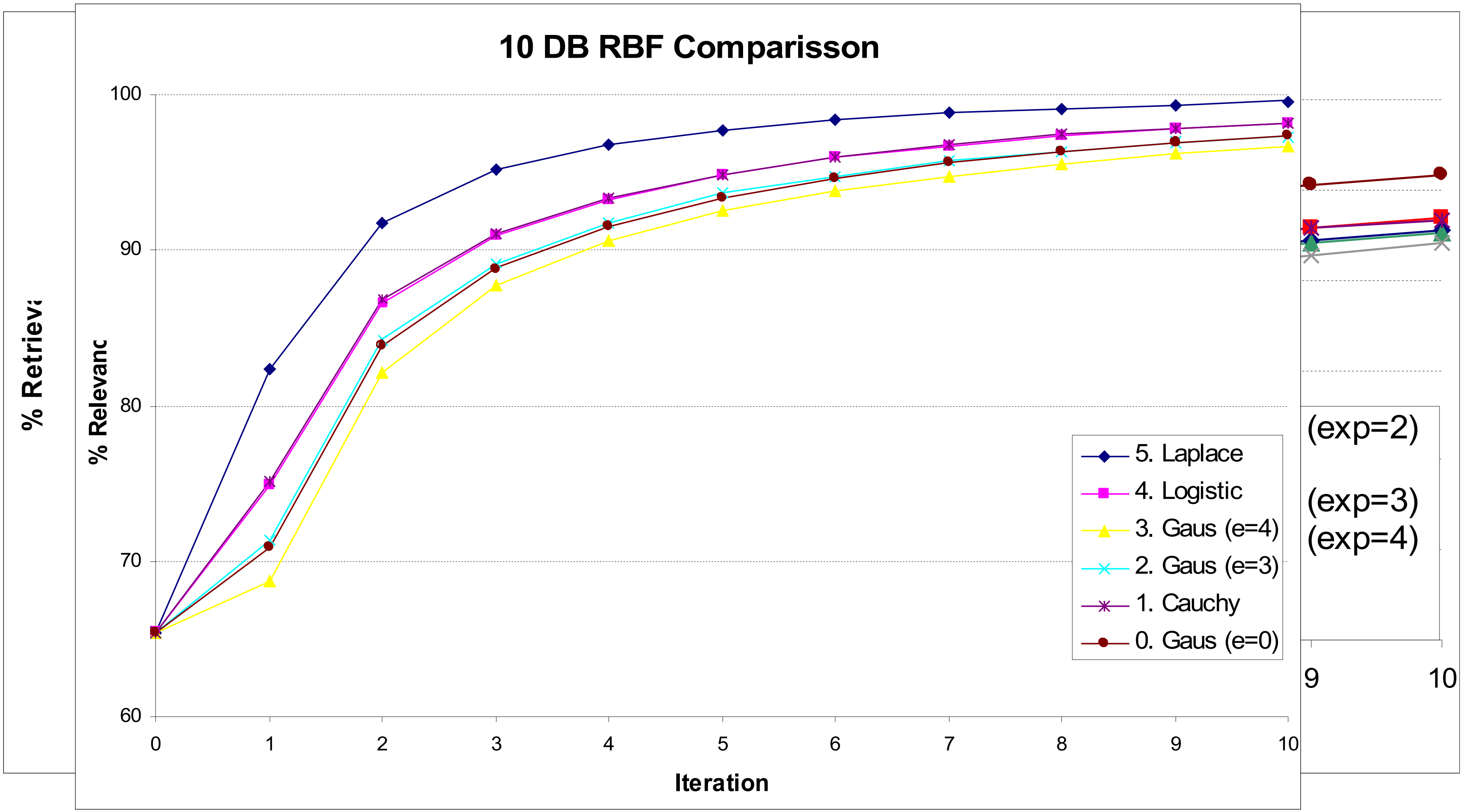
Results
Experiment 1: Filling SS



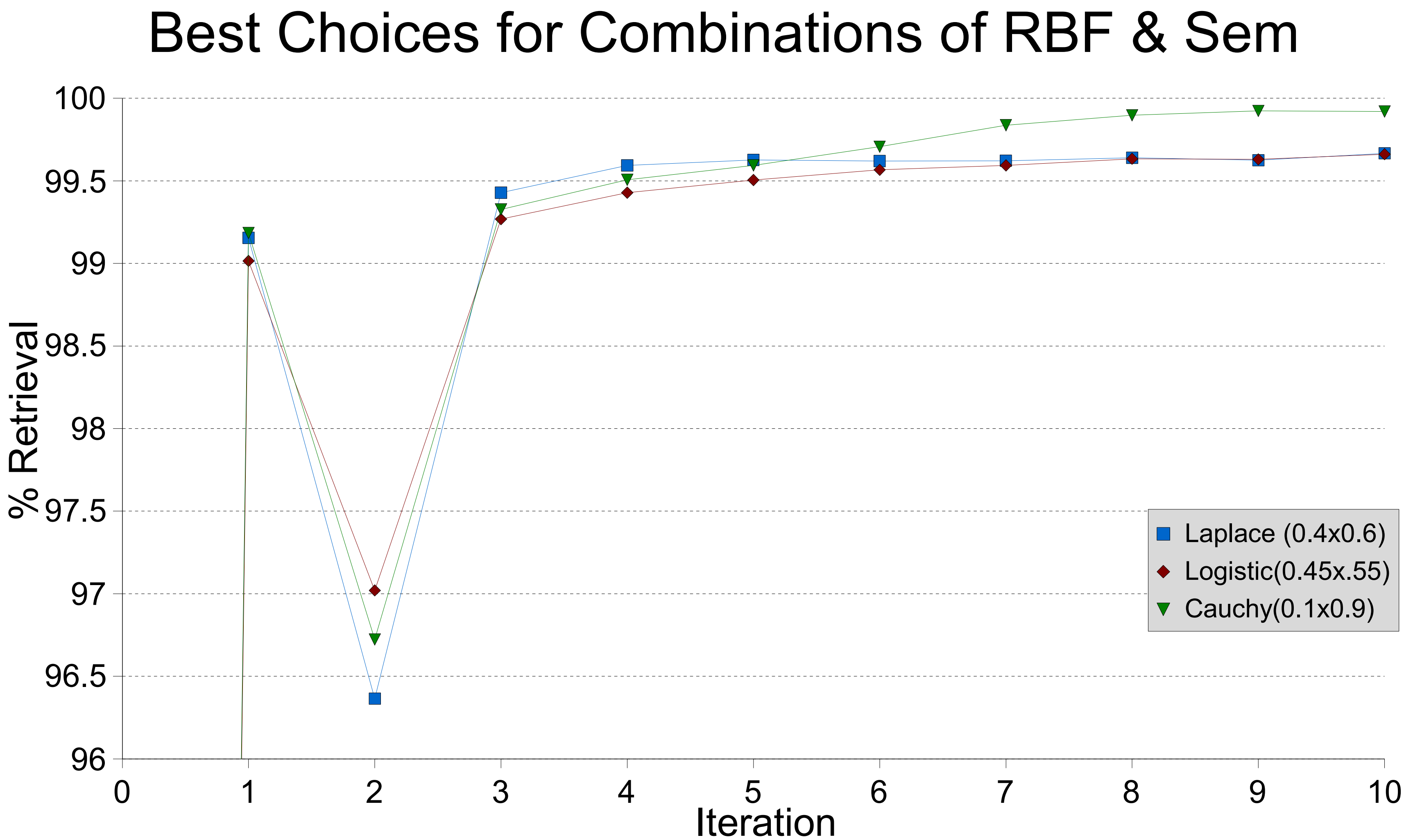
Experiment 2: Size of Basis



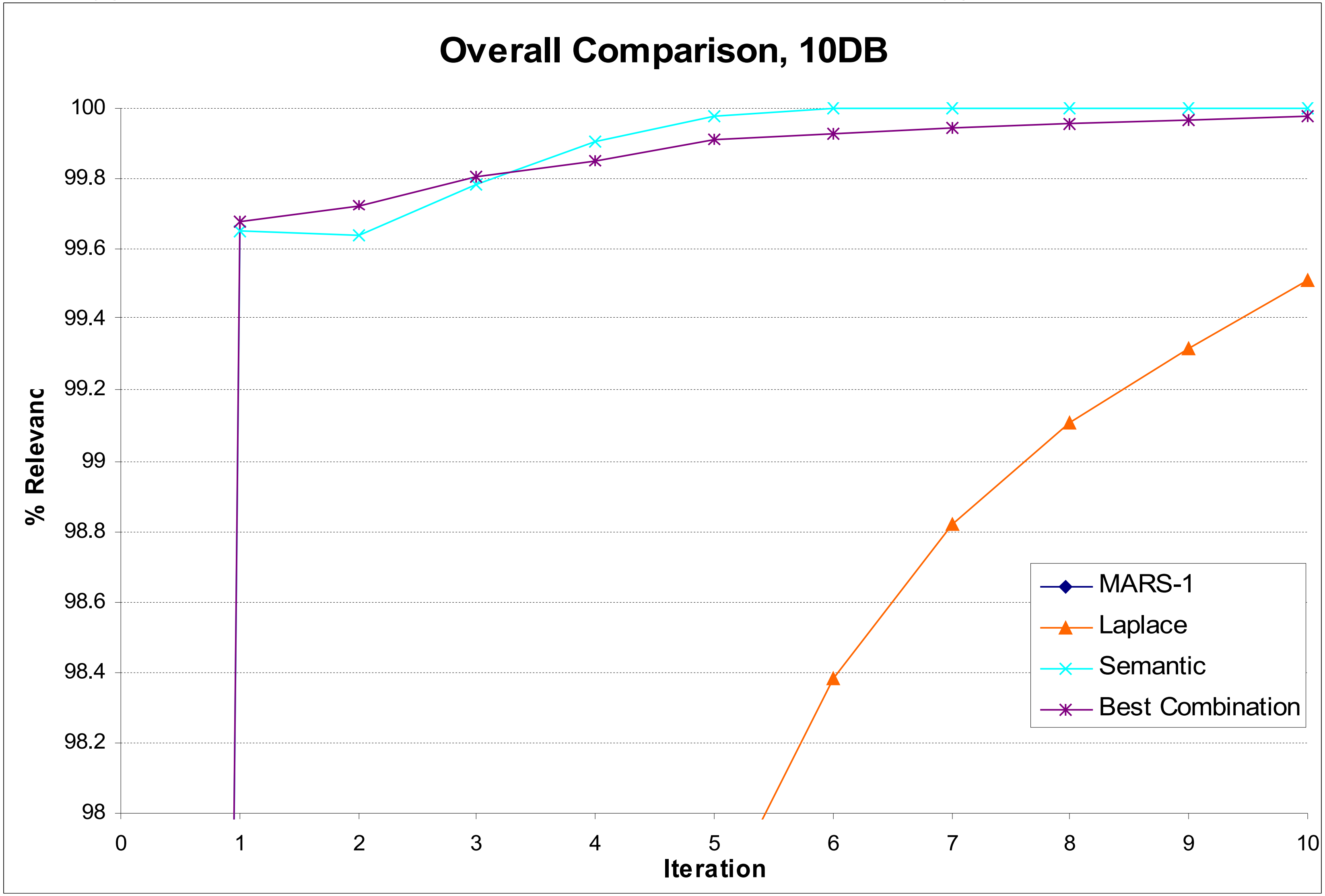
Experiment 3: RBF Selection



Experiment 4: Combinations



Experiment 4: Final Comparison



Conclusions

- Semantic Space is great!
- RBF does help
- need more tests

Future Work

- Implement cross-semantic categorizing
- Automatic combining of RBF & Sem
- Find better RBFs

Questions?