Content Based Image Retrieval Through Semantic Grouping and Linear Discriminant Analysis

Robert Qualls

"…Man, he's so *superlative*"

Overview

- What is CBIR?
- The Semantic Gap
- Relevance Feedback
- Linear Discriminant Analysis (LDA)
- Two Types of LDA
- Semantic Grouping
- Experimental Parameters
- Experimental Process
- Results
- Conclusions
- Difficulties in Implementation
- Resources

What is CBIR?

- Content Based Image Retrieval is the ability of a computer to return, given a query image and a set of images, images in the set which are most similar to the query based on their semantic content (like trucks and stuff).

The Semantic Gap

- In early experimentation, low-level features like color, shape, and texture were proposed as a means to determine whether images were semantically similar.
- In practice, however, using low-level features provided unsatisfactory results.
- This is because of the gap between low-level features and high-level semantics. For example, people can more readily describe an image's semantic concepts than its specific colors or textures.

Relevance Feedback

- It is not practical to develop a system of rules that ties certain combinations of low-level features with semantic concepts. This would quickly lead to a combinatorial explosion.
- Various methods have been proposed for bridging the semantic gap. One of the most popular methods is relevance feedback which utilizes user preferences to improve retrievals. The user provides feedback by telling the system which images are similar or dissimilar to the query image.
- This project used relevance feedback to drive semantic grouping.

Linear Discriminant Analysis

- Invented in 1936 by R.A. Fisher
- LDA is a statistical technique to classify objects into mutually exclusive and exhaustive groups based on a set of measurable objects' features.
- It was used in this project to calculate distance metrics for the data set.
- Often performs just as well or better than more complex / modern classification methods like neural networks or interrogating rabid toddlers.

Two types of LDA

- Two-Class: The goal is to find a distance metric that provides good separation between two classes. In the case of image recognition, the two classes would be the positives and negatives. The downside to this approach is that the negatives may contain images from the same class mixed with images of other classes.
- Multi-Class: The goal is to find a distance metric that provides good separation between all the classes. For this project, the classes are determined via semantic grouping. The downside here is that each negative image is considered its own class until it is known which class it really belongs to.

What LDA Looks Like

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{m} (Y_j - M_i)(Y_j - M_i)'$$

$$S_b = \sum_{i=1}^{C} m_i (M_i - M)(M_i - M)'$$

where:

C	:	the number of class
m	:	the total number of images
m_i	:	the number of images in the class C_i
M	:	the average of all the images
M_i	:	the average of images in the class C_i
Y_i	:	an image

Null Space Method For LDA

- The idea is to project the between-class scatter into the null space of the within-class scatter, and then find the eigenvectors corresponding to the largest eigenvalues of the transferred between-class scatter.

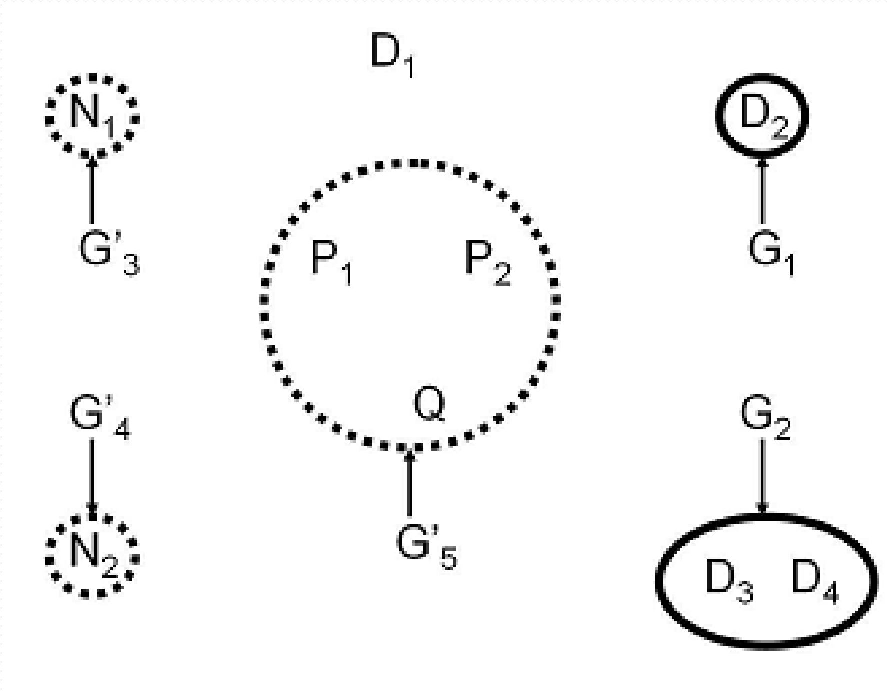
1. Compute the eigenvector factorization: $S_w = \tilde{H}\Lambda\tilde{H}'$.
2. Discard the eigenvectors corresponding to the positive eigenvalues in Λ. The remaining eigenvectors span the null-space of S_w. The matrix of the remaining eigenvectors is denoted by H. It satisfies: $\|H'S_wH\| = 0$.
3. Project the between-class scatter into the null space of the within-class scatter: $\tilde{S}_b = H'S_bH$
4. Compute the eigenvector factorization: $\tilde{S}_b = \tilde{U}\Sigma\tilde{U}'$
5. Select the k eigenvectors in $\tilde{U}$ corresponding to the largest eigenvalues in Σ. Their matrix is denoted by U.
6. Set $W = HU$.

Semantic Grouping

- LDA maximizes the ratio of between-class variance to within-class variance. Those classes have to come from somewhere – in this case, it’s semantic grouping.
- Semantic grouping is a method of grouping images together based on relevancy or irrelevancy to a query image.
- Ultimately, the goal is to have each group contain all of the images of a semantic category (or get as close as possible)
- The grouping algorithm can be interpreted as four cases.

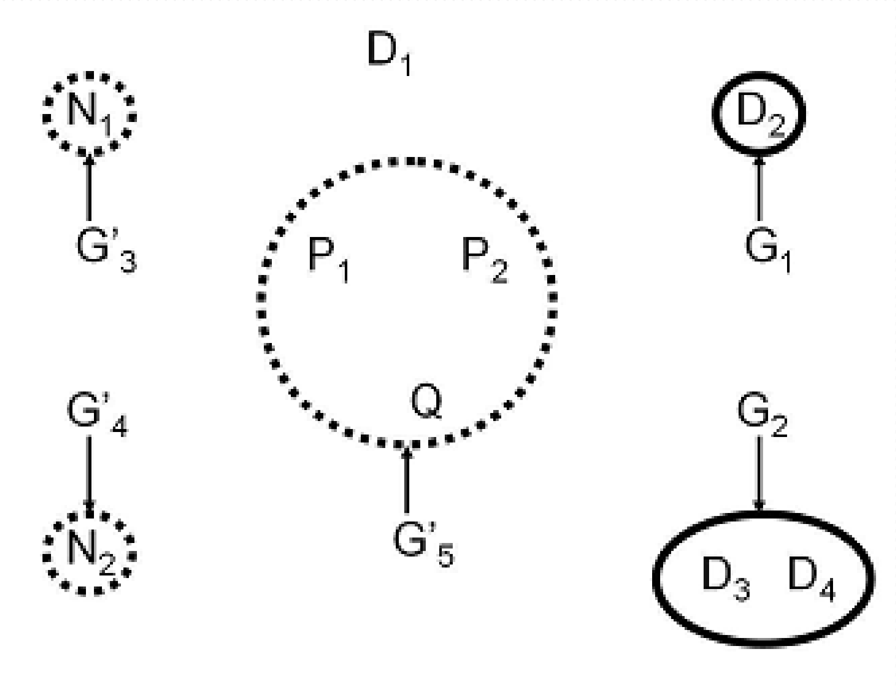
Semantic Grouping Case 1

If all the negative images in the current iteration *have not been assigned* to any group, *put each negative image in an* independent group in the history.



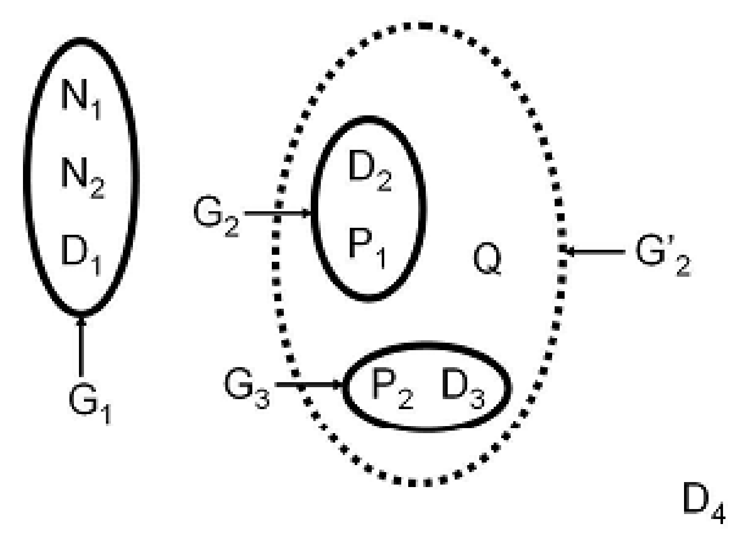
Semantic Grouping Case 2

If all the positive images *have not been as*signed to any group, *put them all in a new group in the history.*



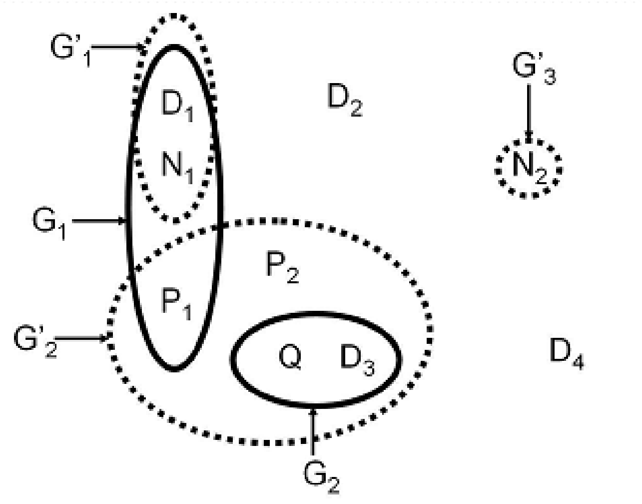
Semantic Grouping Case 3

- If some of the positive images *are assigned* to some of the semantic groups, and *no negative images* are assigned to those groups, create a new group containing all positives and all the images in their groups.



Semantic Grouping Case 4

If some of the positive images *are assigned to* some of the groups*, and some negative images **are** assigned to these groups, put all the positives in a new* group*, and modify the negative-containing groups without the positive images*



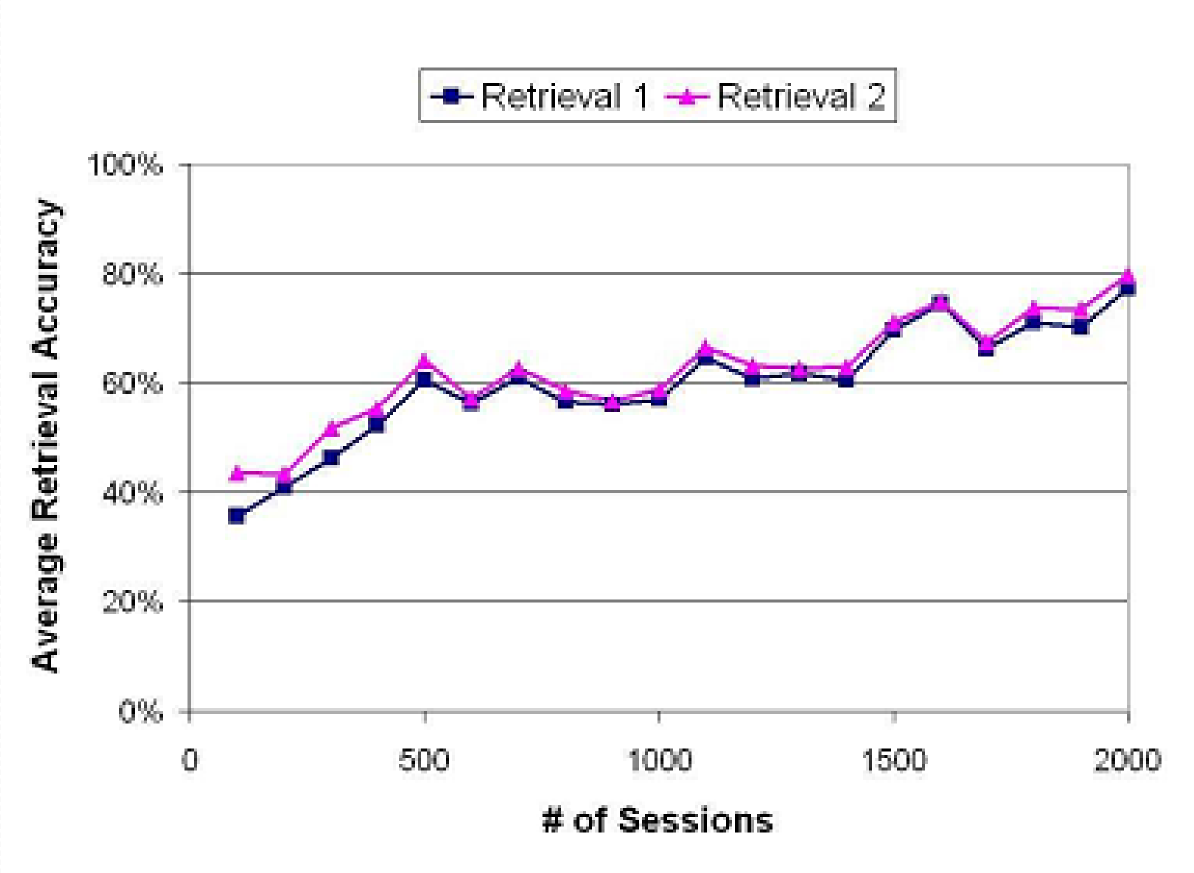
Experimental Parameters

- Dataset: 2000 images, 100 images per category, feature vector length of 100.
- Image candidates per retrieval: 30
- Sessions performed: 2000
- Retrievals per session: 2

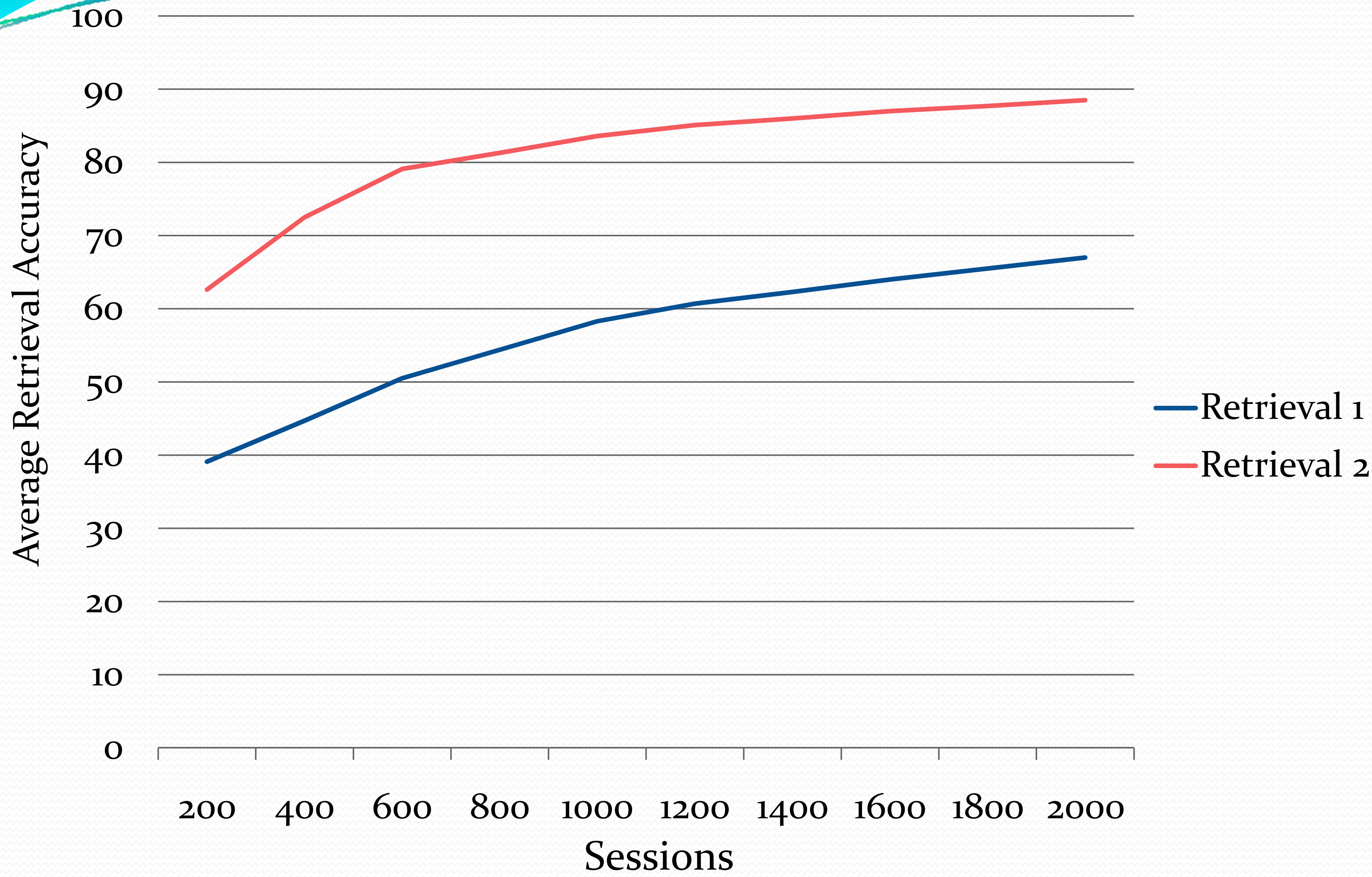
Experimental Process

1. A query image is randomly selected from the database
2. If it is the first retrieval of the session, the original distance metric is used. If it's the second or later retrieval, the distance metric computed in the previous retrieval is used.
3. Using the distance metric, the 30 smallest distance images are returned to the user.
4. The user tells the system which images are positive or negative.
5. The system updates the semantic group history
6. Using the updated history, the system computes a new distance metric via LDA.

The Paper's Results

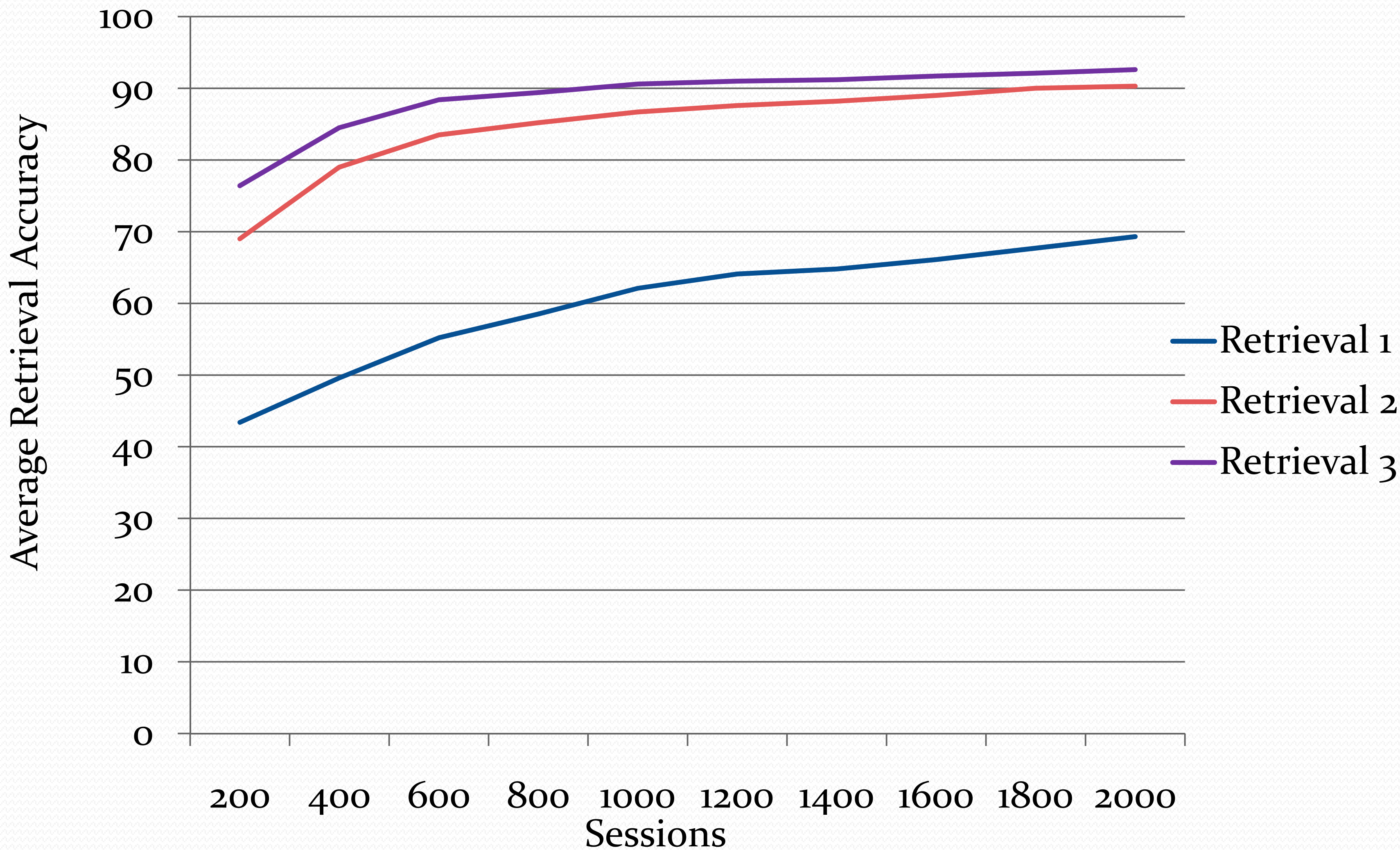


My Results

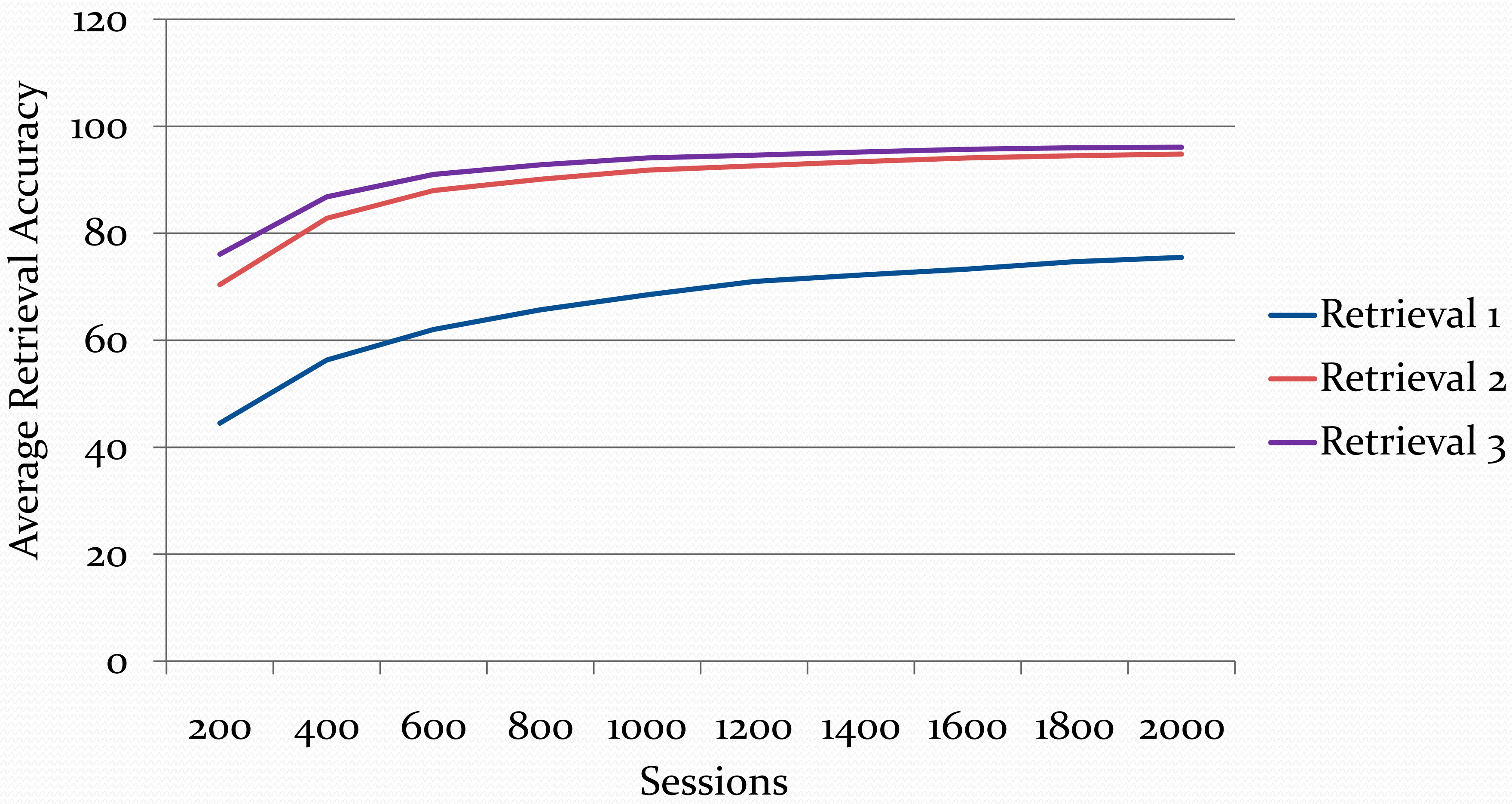


Average Retrieval Accuracy
100
90
80
70
60
50
40
30
20
10
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Sessions
Retrieval 1
Retrieval 2

My Results Using 3 Retrievals



My Results Using 3 Retrievals and 60 Candidates per Retrieval



Implementation Problems

- Matlab.
- I used a matrix that contained the image identifiers for each semantic group. As the number of groups rose past 3,000, and as the number of images per group got closer to 100, "not enough memory" errors started to become more common. Ultimately, code was written that allowed the system to reuse old but empty groups.
- My implementation required maintaining several related matrices and arrays for the semantic grouping algorithm. This made it very tedious to write. In fact, it took over a month to get it working.
- In order for semantic grouping to work effectively, image indices needed to be randomized since the images were already categorized.

Resources

- T. Yoshizawa and H. Schweitzer. Long-term learning of semantic grouping from relevance-feedback. In MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval, pages 165–172, New York, NY, USA, 2004. ACM Press.
- http://people.revoledu.com/kardi/tutorial/LDA/LDA.html
- http://www.dtreg.com/lda.htm
- http://people.revoledu.com/kardi/tutorial/LDA/