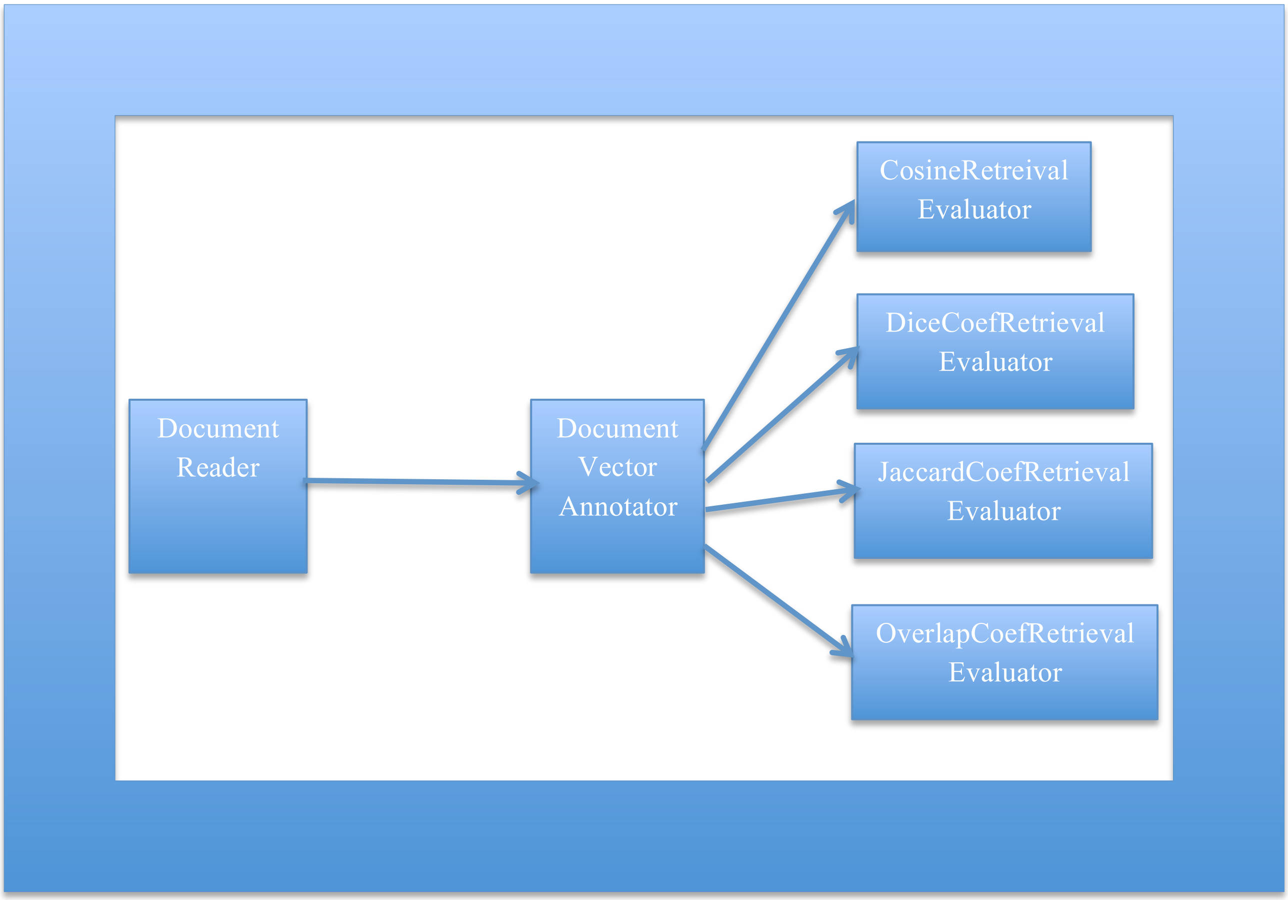
Napat Luevisadpaibul **ID:nluevisa**

Homework 4

1 Overview Design

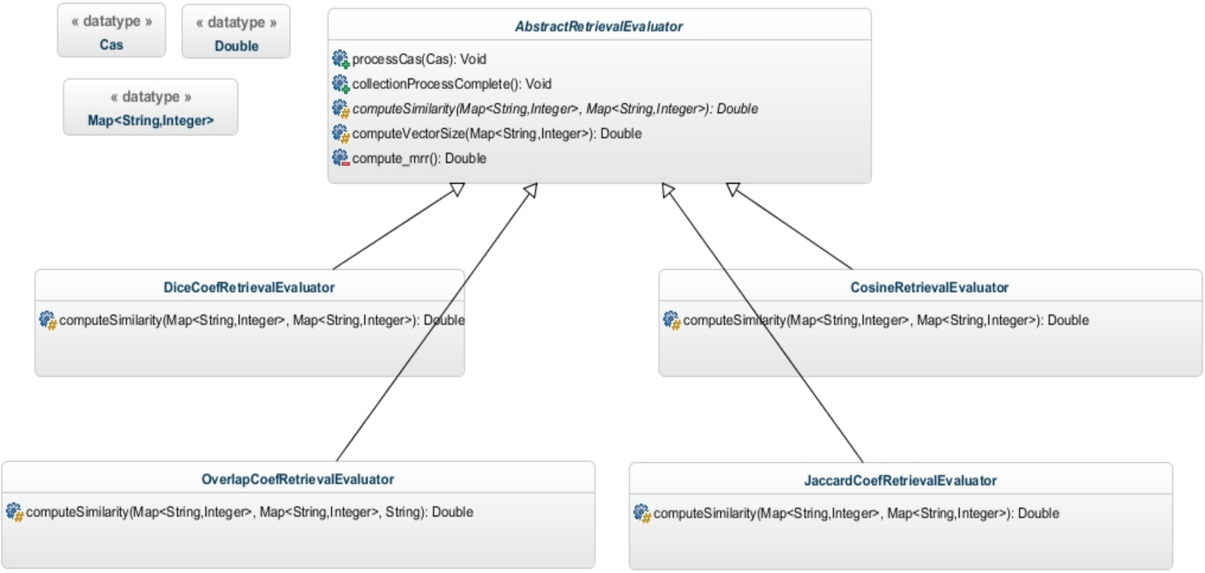
First, let's begin with the design of the aggregate analysis engine running this system. Below is a figure showing primitives analysis engines included in the final version of aggregate analysis engine.


Document Reader
Document Vector Annotator
CosineRetreival Evaluator
DiceCoefRetrieval Evaluator
JaccardCoefRetrieval Evaluator
OverlapCoefRetrieval Evaluator

DocumentReader act as a collection reader that read and parse text from input file to generate initial Document annotation. The Document annotation created by DocumentReader has its feature Text, QueryID and Relevance set appropriately. Then DocumentVectorAnnotator will create TokenList for each Document annotation by adding Token (consist of term and frequency feature) generated from Text to its TokenList. Finally, each Evaluator (act as CasConsumer) compute score for each documents with respect to their query Id, rank the result, then compute MRR metric and show it to the screen.

I apply **"Template Method" design pattern** for the Evaluator. Basically each evaluator almost does the same thing that is creating document term vectors, compute similarity score and compute MRR. The only thing that each evaluator does differently is to how to compute similarity score. Thus template method is a reasonable design pattern in this case.

Class Diagram for Evaluator can be shown below:


« datatype »
Cas
« datatype »
Double
« datatype »
Map<String,Integer>
AbstractRetrievalEvaluator
processCas(Cas): Void
collectionProcessComplete(): Void
computeSimilarity(Map<String,Integer>, Map<String,Integer>): Double
computeVectorSize(Map<String,Integer>): Double
compute_mrr(): Double
DiceCoefRetrievalEvaluator
computeSimilarity(Map<String,Integer>, Map<String,Integer>): Double
CosineRetrievalEvaluator
computeSimilarity(Map<String,Integer>, Map<String,Integer>): Double
OverlapCoefRetrievalEvaluator
computeSimilarity(Map<String,Integer>, Map<String,Integer>, String): Double
JaccardCoefRetrievalEvaluator
computeSimilarity(Map<String,Integer>, Map<String,Integer>): Double

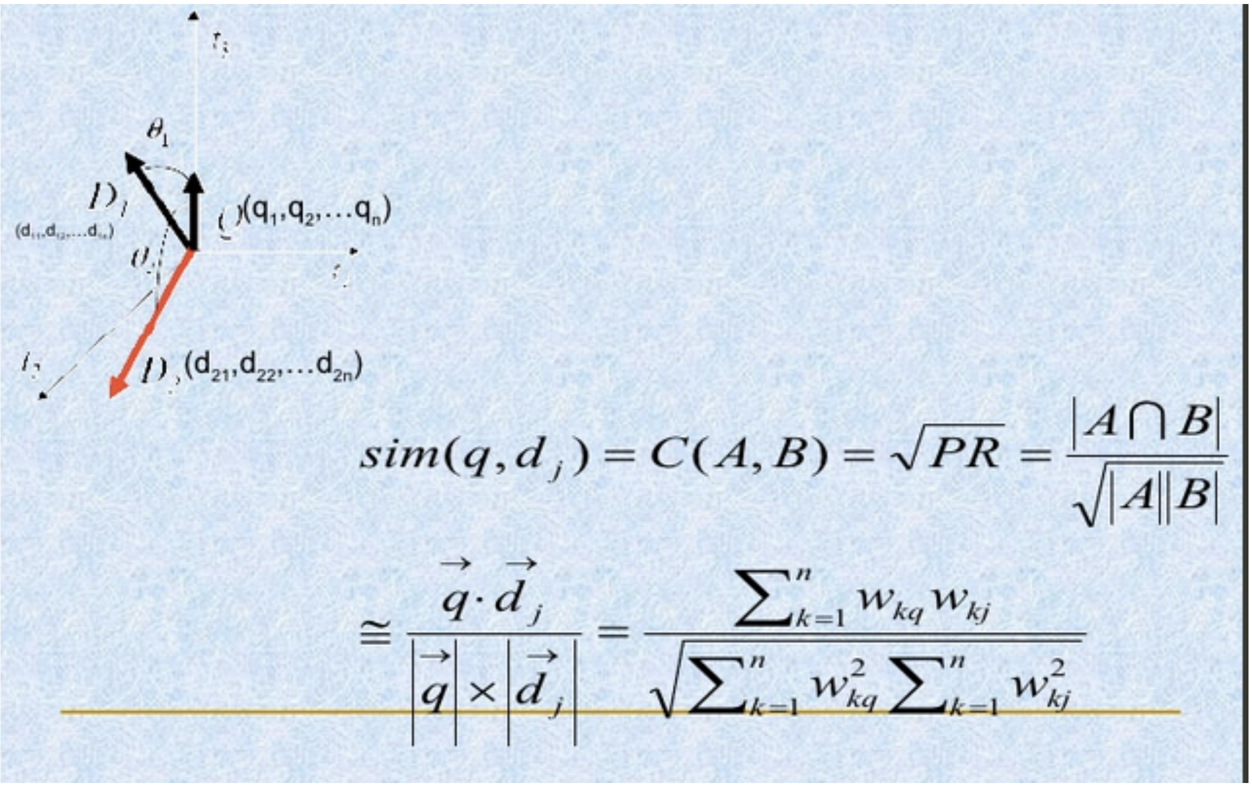
AbstractRetievalEvaluator is an abstract class that implement all methods necessary to evaluate system except **abstract computeSimilary** method. computeSimilarity can be called template method in template design pattern and the subclasses will implement this method using its own similarity function. computeVectorSize is protected because it is a common method used by all of subclass.

Each concrete evaluator class is represent a primitive analysis engine and independent to each other. That’s why we can compare the result of multiple document ranking strategies in a single execution of the pipeline by add these evaluator’s descriptors in an aggregate analysis engine.

2 Similarity Function

This section will provide the formula to calculate similarity between documents in this system.

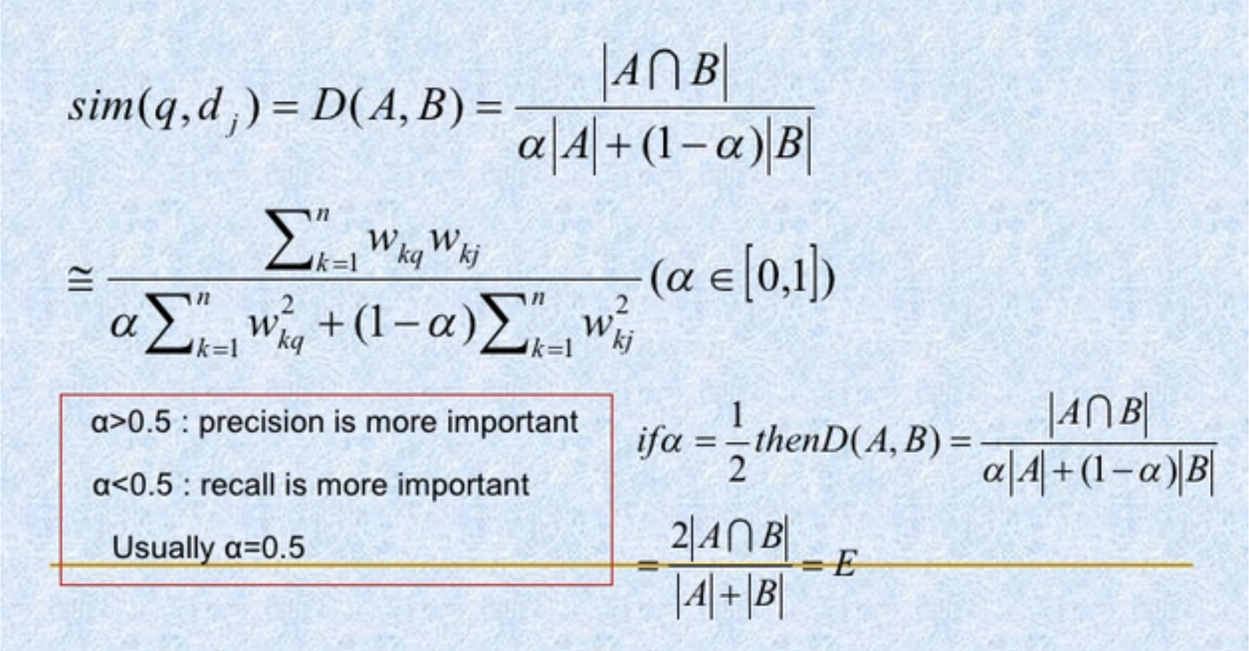
2.1 Cosine Similarity



$$sim(q,d_j) = C(A,B) = \sqrt{PR} = \frac{|A \cap B|}{\sqrt{|A||B|}}$$

$$\cong \frac{\vec{q}\cdot\vec{d}_j}{|\vec{q}|\times|\vec{d}_j|} = \frac{\sum_{k=1}^{n} w_{kq}w_{kj}}{\sqrt{\sum_{k=1}^{n} w_{kq}^2 \sum_{k=1}^{n} w_{kj}^2}}$$

2.2 Dice Coefficient (I use alpha = 0.5)



$$sim(q,d_j) = D(A,B) = \frac{|A \cap B|}{\alpha|A| + (1-\alpha)|B|}$$

$$\cong \frac{\sum_{k=1}^{n} w_{kq}w_{kj}}{\alpha\sum_{k=1}^{n} w_{kq}^2 + (1-\alpha)\sum_{k=1}^{n} w_{kj}^2} (\alpha \in [0,1])$$

α>0.5 : precision is more important

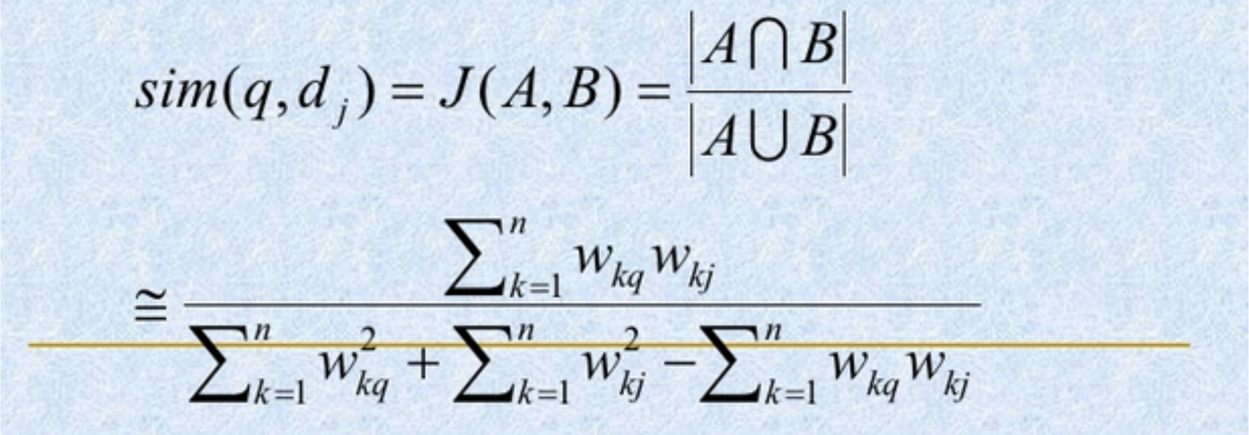
α<0.5 : recall is more important

Usually α=0.5

$$if\alpha = \frac{1}{2} thenD(A,B) = \frac{|A \cap B|}{\alpha|A| + (1-\alpha)|B|}$$

$$= \frac{2|A \cap B|}{|A| + |B|} = E$$

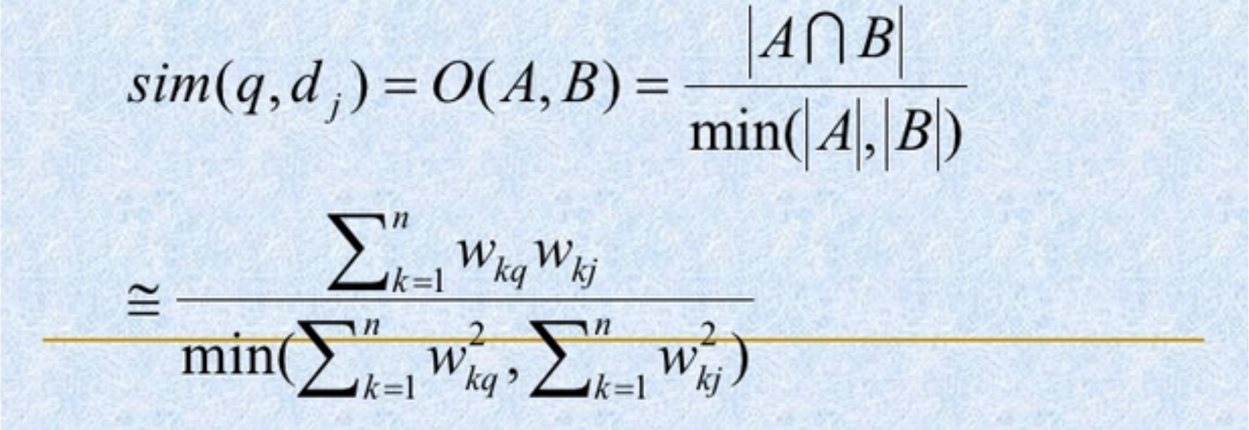
2.3 Jaccard Coefficient



$$sim(q,d_j) = J(A,B) = \frac{\left|A \cap B\right|}{\left|A \cup B\right|}$$

$$\cong \frac{\sum_{k=1}^{n} w_{kq} w_{kj}}{\sum_{k=1}^{n} w_{kq}^2 + \sum_{k=1}^{n} w_{kj}^2 - \sum_{k=1}^{n} w_{kq} w_{kj}}$$

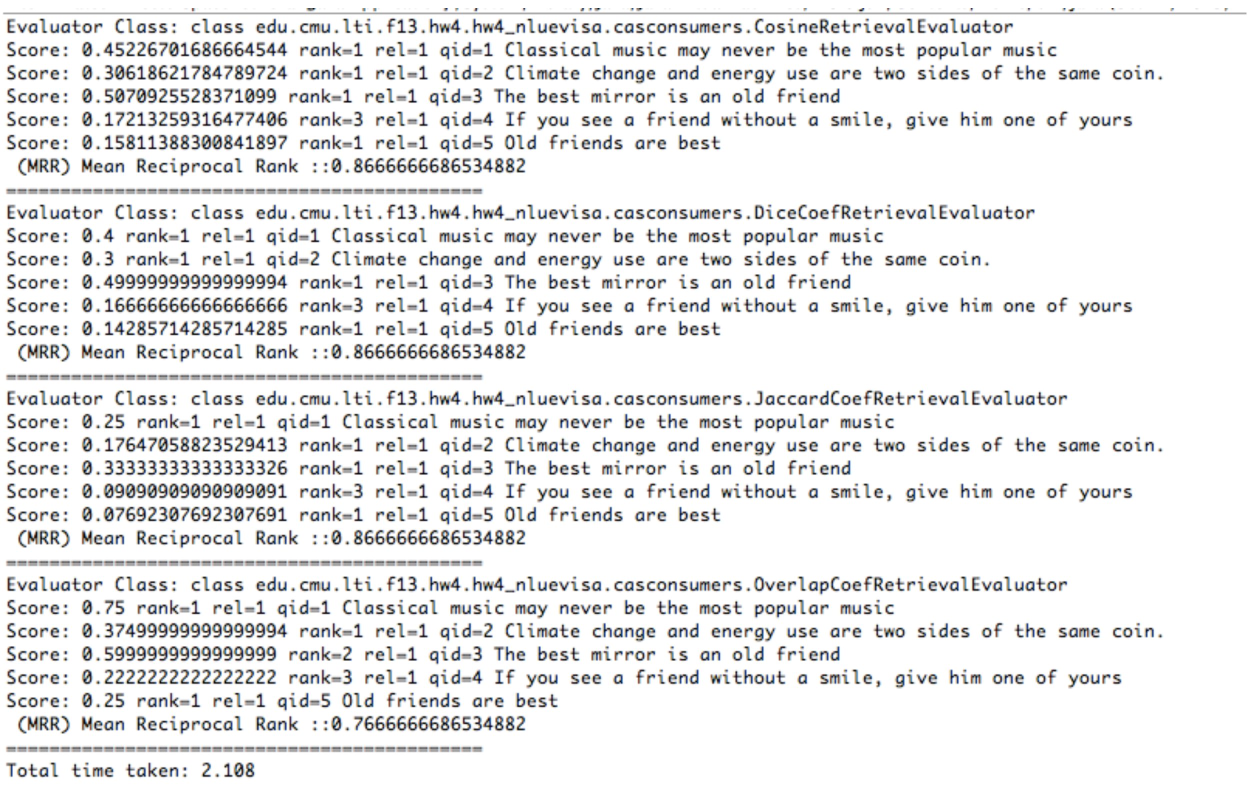
2.4 Overlap Coefficient



$$sim(q,d_j) = O(A,B) = \frac{\left|A \cap B\right|}{\min(\left|A\right|, \left|B\right|)}$$

$$\cong \frac{\sum_{k=1}^{n} w_{kq} w_{kj}}{\min(\sum_{k=1}^{n} w_{kq}^2, \sum_{k=1}^{n} w_{kj}^2)}$$

3 Program Result before Doing Error Analysis



```
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.CosineRetrievalEvaluator
Score: 0.4522670168664544 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.30618621784789724 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5070925528371099 rank=1 rel=1 qid=3 The best mirror is an old friend
Score: 0.17213259316477406 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.15811388300841897 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.8666666686534882
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.DiceCoefRetrievalEvaluator
Score: 0.4 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.3 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.49999999999999994 rank=1 rel=1 qid=3 The best mirror is an old friend
Score: 0.16666666666666666 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.14285714285714285 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.8666666686534882
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.JaccardCoefRetrievalEvaluator
Score: 0.25 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.17647058823529413 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.33333333333333326 rank=1 rel=1 qid=3 The best mirror is an old friend
Score: 0.09090909090909091 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.07692307692307691 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.8666666686534882
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.OverlapCoefRetrievalEvaluator
Score: 0.75 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.37499999999999994 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5999999999999999 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.2222222222222222 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.25 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.7666666686534882
============================================
Total time taken: 2.108
```

The result shows that Cosine Similarity, Dice Coefficient and Jaccard Coefficient produce the same MRR metric (0.8666), while Overlap Coefficient give the least score (0.7666). The MRR for the top three methods may be the same but the score from each method are clearly different so we should see different in MRR if we have enough samples.

4 Error Analysis

From the previous experiment, we can clearly see that the only query that our system did not do well is query four, so we have to identify what cause the error in this query.

I print Document Term Vector for all document associated with query ID 4 and got result like this:

```
Query:
{distance=1, new=1, is=1, shortest=1, friends=1, a=1, the=1, smile=1,
between=1}

Incorrect Document:
{wrinkles=1, have=2, a=2, scowl=1, friends;=1, smile=1, wear=2, and=2}

Correct Document:
```

```
{of=1, see=1, give=1, if=1, smile,=1, a=2, one=1, you=1, him=1,
without=1, friend=1, yours=1}

Incorrect Document:
{girls=1, is=1, put=1, it=1, a=1, there=1, behind=1, best=1, every=1,
smile=1, friend=1, who=1}
```

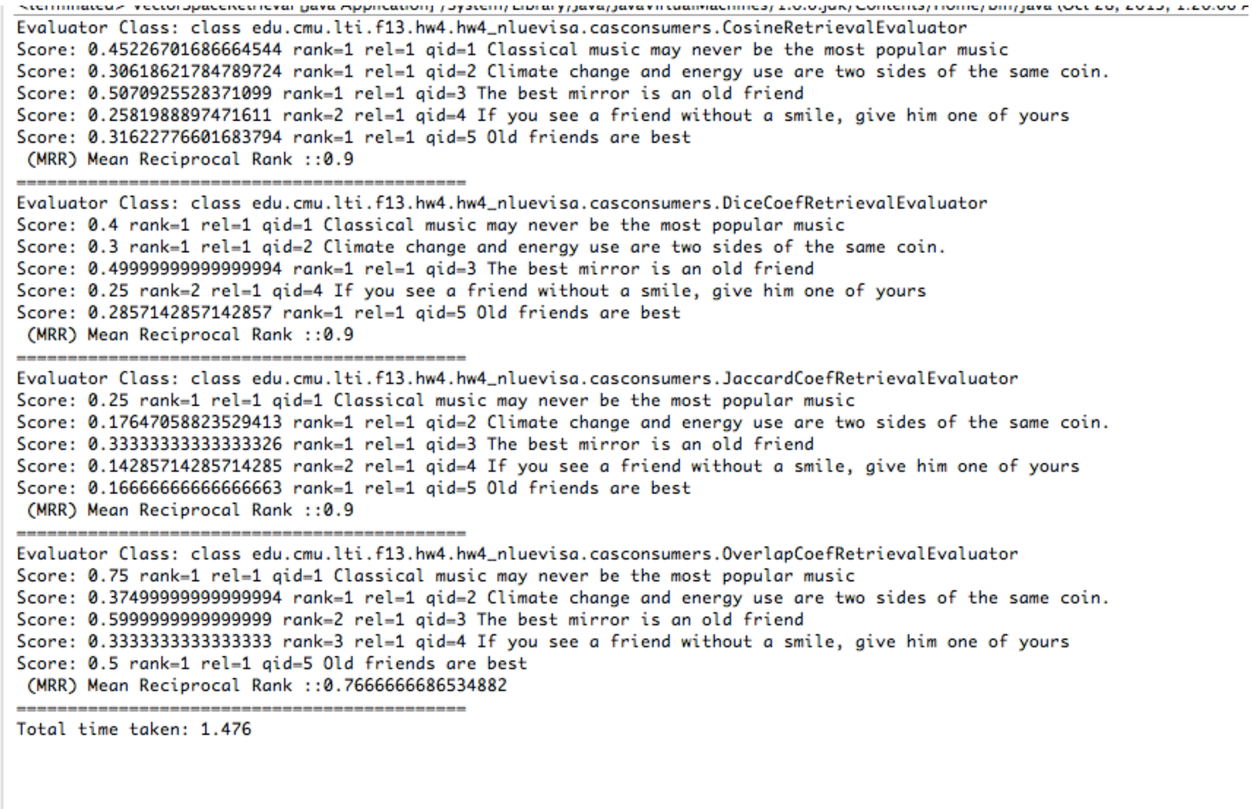
From this answer I notice two errors:

1. The system treat terms "friends" and "friend" as different term but we should treat them as the same term by stemming the terms before add it to TokenList.
2. Document vector contain many common word like 'is', 'a' ,'the' which should not be the term that contain in our vector.

I improve the system by using **Porter Stemming algorithm** to stem the word before add to TokenList and experiment on discarding the stop word too. All of this step can be done in DocumentVectorAnnotator class.

5 Program Result after Doing Error Analysis

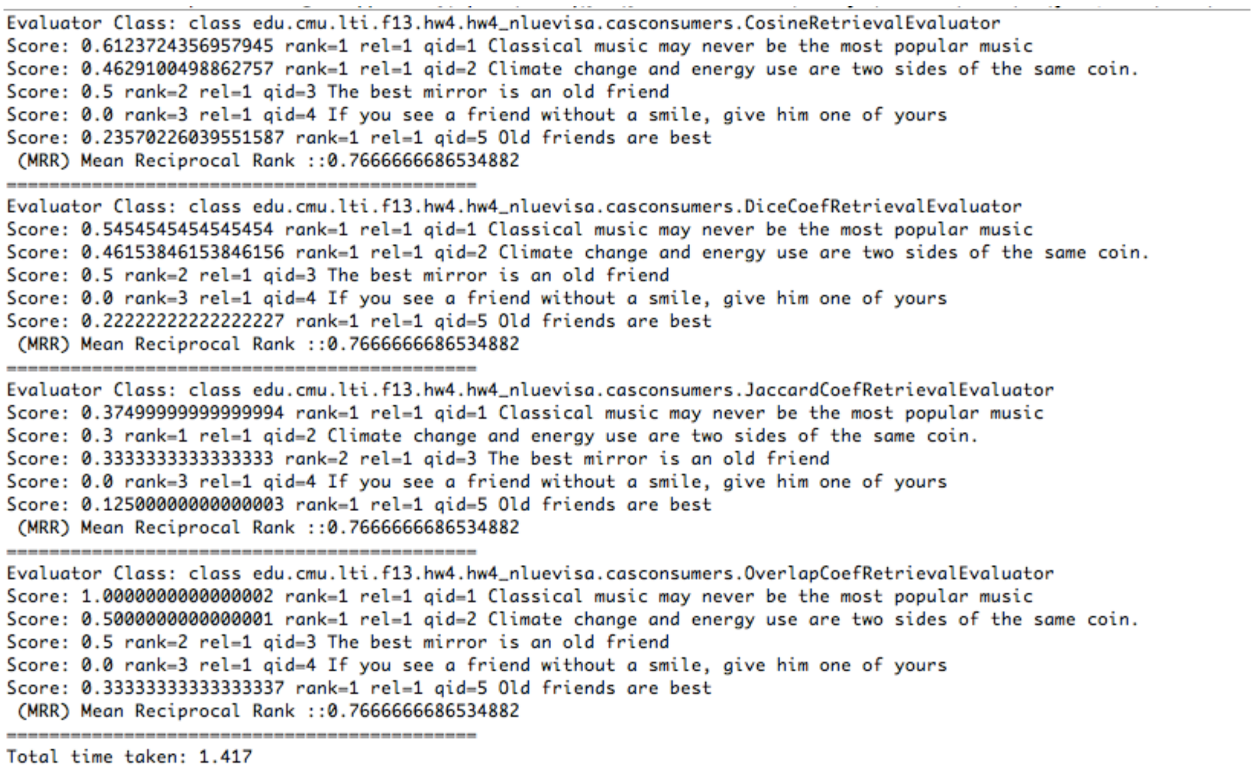
5.1 Result from apply Porter Stemmer (not discard stop word)



```
<terminated> VectorSpaceRetrieval [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Oct 28, 2013, 1:20:08 P
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.CosineRetrievalEvaluator
Score: 0.4522670168664544 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.30618621784789724 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5070925528371099 rank=1 rel=1 qid=3 The best mirror is an old friend
Score: 0.2581988897471611 rank=2 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.31622776601683794 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.9
=========================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.DiceCoefRetrievalEvaluator
Score: 0.4 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.3 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.49999999999999994 rank=1 rel=1 qid=3 The best mirror is an old friend
Score: 0.25 rank=2 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.2857142857142857 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.9
=========================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.JaccardCoefRetrievalEvaluator
Score: 0.25 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.17647058823529413 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.33333333333333326 rank=1 rel=1 qid=3 The best mirror is an old friend
Score: 0.14285714285714285 rank=2 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.16666666666666663 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.9
=========================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.OverlapCoefRetrievalEvaluator
Score: 0.75 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.37499999999999994 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5999999999999999 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.3333333333333333 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.5 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.7666666686534882
=========================================
Total time taken: 1.476
```

We can see that the result does improve. MRR for using Cosine, Dice and Jackard improve from 0.8666 to 0.9 while overlap got the same result.

5.2 Result from discarding stop word (not apply Porter Stemmer)



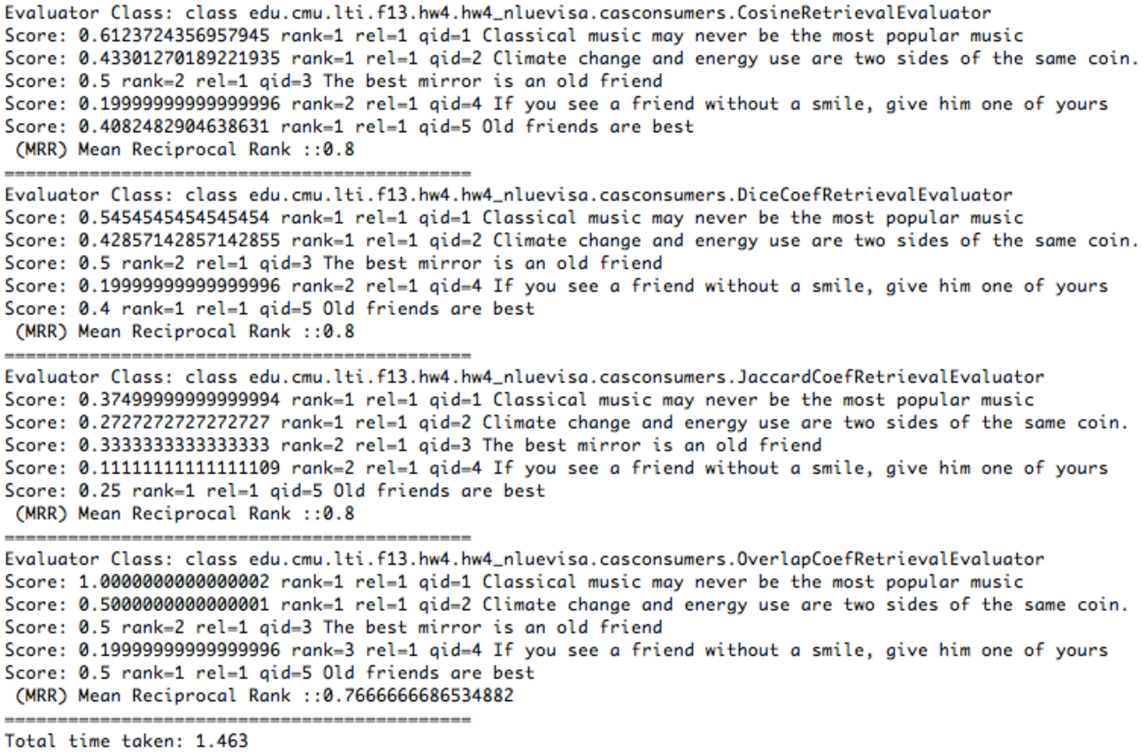
```
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.CosineRetrievalEvaluator
Score: 0.6123724356957945 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.4629100498862757 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.0 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.23570226039551587 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.7666666686534882
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.DiceCoefRetrievalEvaluator
Score: 0.5454545454545454 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.46153846153846156 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.0 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.22222222222222227 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.7666666686534882
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.JaccardCoefRetrievalEvaluator
Score: 0.37499999999999994 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.3 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.3333333333333333 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.0 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.12500000000000003 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.7666666686534882
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.OverlapCoefRetrievalEvaluator
Score: 1.0000000000000002 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.5000000000000001 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.0 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.33333333333333337 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.7666666686534882
============================================
Total time taken: 1.417
```

The result actually has gotten worse because now correct document from query 3 drops to rank 2. That is because the incorrect document that had been penalized before discard stopping words now get higher score that correct document

```
qid=3 rel=99    One's best friend is oneself
Doc1: qid=3     rel=1  The best mirror is an old friend
Doc2: qid=3     rel=0  My best friend is the one who brings out the
best in me
Doc3: qid=3     rel=0  The best antiques are old friends
```

Doc2 now get higher score than doc1 because it has term "best" occur twice and "friend" occur once while Doc1 has both "best" and "friend" occur onces. Before we discarded stop word it had been penalize by its length but after discard stop word, the size of Doc1 and Doc2 are not much difference and frequency of term "best" dominate the score.

5.3 Result from apply both Stemming and Discard Stop word



```
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.CosineRetrievalEvaluator
Score: 0.6123724356957945 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.43301270189221935 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.19999999999999996 rank=2 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.4082482904638631 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.8
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.DiceCoefRetrievalEvaluator
Score: 0.5454545454545454 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.42857142857142855 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.19999999999999996 rank=2 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.4 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.8
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.JaccardCoefRetrievalEvaluator
Score: 0.37499999999999994 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.2727272727272727 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.3333333333333333 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.11111111111111109 rank=2 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.25 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.8
============================================
Evaluator Class: class edu.cmu.lti.f13.hw4.hw4_nluevisa.casconsumers.OverlapCoefRetrievalEvaluator
Score: 1.0000000000000002 rank=1 rel=1 qid=1 Classical music may never be the most popular music
Score: 0.5000000000000001 rank=1 rel=1 qid=2 Climate change and energy use are two sides of the same coin.
Score: 0.5 rank=2 rel=1 qid=3 The best mirror is an old friend
Score: 0.19999999999999996 rank=3 rel=1 qid=4 If you see a friend without a smile, give him one of yours
Score: 0.5 rank=1 rel=1 qid=5 Old friends are best
 (MRR) Mean Reciprocal Rank ::0.7666666686534882
============================================
Total time taken: 1.463
```

6 Conclusion

My system adopt “Template Method” design pattern to reduce code redundancy and be able to compare the result of four similarity functions in the same pipeline. After doing error analysis, I experiment two preprocessing methods that is 1. Using Porter Stemmer to stem term 2. Discard stop word. The experiment result shows that using Porter Stemmer can improve MRR metric, but discard stop word get worse result in this case. I believe that in larger sample data set, discarding stop word will likely improve the result. In my current system, I comment out the code that discard stop word to achieve MRR = 0.9. I can turn it on again by uncomment it or use configuration to handle on/off discarding stop word option.