11-791 HOMEWORK 4 REPORT

Siping Ji
sji
sipingji@cmu.edu
10/26/2013

Tasks

0. automatically generate the type system implementation using UIMA>>JCasGen

1. correctly extract bag of words feature vector from the input text collection

2. compute the cosine similarity between two sentences in the text collection

3. compute the Mean Reciprocal Rank (metric) for all sentences in the text collection

4. Design a better retrieval system that improves the MRR performance measure

5. Improve the efficiency of the program by doing error analysis of retrieval system

Experiment

Part 1 - Vector Space Retrieval Model implementation

Here I list some of the highlights in my system implementation:

1. Tokenization. It's straightforward to use regular expression to get the tokens of a sentence. To record the frequency of a token, I use hash map to store these tokens, and count the number of occurrences for each token.

2. StopWord removal. Stop words usually act as noise rather than as evidence in information retrieval, therefore, during tokenization, I removed all the stop words.

3. Cosine similarity computation. To efficiently compute the cosine similarity, I use Hash-map to represent each document/query vector. Therefore, the dot product and vector length is easy to implement - just iterate through keys of the hash map.

4. Mean Reciprocal Rank computation. In order to get this metric, we first need to get a rank list of the document where documents are ranked by cosine similarity value. Therefore, I implement a sentence class to represent the document, where attributes are query Id, relevance, document text and the similarity score. I also realize the comparable interface for this class, so that I can use Collections.sort to sort the list of the sentences. After we get the rank list of the document, the computation of MRR is trivial.

Part 2 - Error Analysis

The performance of the information retrieval system I implemented in the first part is as follows:

```
Similarity Measure:CosineSimilarity
qId = 1 rel = 1 0.6124 Classical music may never be the most popular
music
qId = 1 rel = 0 0.3849 Pop music has absorbed influences from most
other genres of popular music
qId = 1 rel = 0 0.2582 Everybody knows classical music when they hear
it

qId = 2 rel = 1 0.1543 Climate change and energy use are two sides of
the same coin.
qId = 2 rel = 0 0.0000 Old wine and friends improve with age
qId = 2 rel = 0 0.0000 With clothes the new are the best, with friends
the old are the best

qId = 3 rel = 0 0.4743 My best friend is the one who brings out the
best in me
qId = 3 rel = 1 0.4000 The best mirror is an old friend
qId = 3 rel = 0 0.2000 The best antiques are old friends

qId = 4 rel = 0 0.3333 Wear a smile and have friends; wear a scowl and
have wrinkles
qId = 4 rel = 1 0.1543 If you see a friend without a smile, give him
one of yours
qId = 4 rel = 0 0.1543 Behind every girls smile is a best friend who
put it there
```

```
qId = 5 rel = 0 0.1260 With clothes the new are the best, with friends
the old are the best
qId = 5 rel = 0 0.0000 Old wine and friends improve with age
qId = 5 rel = 1 0.0000 Old friends are best

 (MRR) Mean Reciprocal Rank ::0.6666666666666667
Total time taken: 0.935
```

We can see that two relevant are correctly retrieved in the first place , and two retrieved in the second place, one in the third. Therefore the MRR is 0667.

To improve the performance, we try to do error analysis here. There are two major steps in our information system. Phase 1 is tokenization, Phase 2 is matching step. We start from phase 1.

We can improve the tokenization result in a very simple way, normalize the tokens. We do it by converting all the tokens to their lowercases, i.e Pop and pop should map to the same token. The result after this refinement is showed below:

```
Similarity Measure:CosineSimilarity
qId = 1 rel = 1 0.6124 Classical music may never be the most popular
music
qId = 1 rel = 0 0.5164 Everybody knows classical music when they hear
it
qId = 1 rel = 0 0.3849 Pop music has absorbed influences from most
other genres of popular music

qId = 2 rel = 1 0.4629 Climate change and energy use are two sides of
the same coin.
qId = 2 rel = 0 0.0000 Old wine and friends improve with age
qId = 2 rel = 0 0.0000 With clothes the new are the best, with friends
the old are the best

qId = 3 rel = 0 0.5669 My best friend is the one who brings out the
best in me
qId = 3 rel = 1 0.5000 The best mirror is an old friend
qId = 3 rel = 0 0.2500 The best antiques are old friends

qId = 4 rel = 0 0.3162 Wear a smile and have friends; wear a scowl and
have wrinkles
```

```
qId = 4 rel = 1 0.1826 If you see a friend without a smile, give him
one of yours
qId = 4 rel = 0 0.1690 Behind every girls smile is a best friend who
put it there

qId = 5 rel = 1 0.2357 Old friends are best
qId = 5 rel = 0 0.1826 Old wine and friends improve with age
qId = 5 rel = 0 0.1443 With clothes the new are the best, with friends
the old are the best

 (MRR) Mean Reciprocal Rank ::0.8
Total time taken: 0.868
```

The performance is improved as expected.

Now let's think about how to refine matching algorithm in phase 2.

I first tried Jaccard Coefficient, a simpler method to just count co-occurrence of tokens as similarity. The logic to use this metric is that the documents here are rather short, some co-occurrence of keywords may well capture the gist of the document(sentence). The result is as follows:

```
Similarity Measure:JaccardCoefficient
qId = 1 rel = 1 0.3333 Classical music may never be the most popular
music
qId = 1 rel = 0 0.3333 Everybody knows classical music when they hear
it
qId = 1 rel = 0 0.1250 Pop music has absorbed influences from most
other genres of popular music

qId = 2 rel = 1 0.3000 Climate change and energy use are two sides of
the same coin.
qId = 2 rel = 0 0.0000 Old wine and friends improve with age
qId = 2 rel = 0 0.0000 With clothes the new are the best, with friends
the old are the best

qId = 3 rel = 1 0.3333 The best mirror is an old friend
qId = 3 rel = 0 0.3333 My best friend is the one who brings out the
best in me
qId = 3 rel = 0 0.1429 The best antiques are old friends
```

```
qId = 4 rel = 0 0.2500 Wear a smile and have friends; wear a scowl and
have wrinkles
qId = 4 rel = 1 0.1000 If you see a friend without a smile, give him
one of yours
qId = 4 rel = 0 0.0909 Behind every girls smile is a best friend who
put it there

qId = 5 rel = 1 0.1250 Old friends are best
qId = 5 rel = 0 0.1000 Old wine and friends improve with age
qId = 5 rel = 0 0.1000 With clothes the new are the best, with friends
the old are the best

 (MRR) Mean Reciprocal Rank ::0.9
Total time taken: 0.863
```

This time, the MRR is improved to 0.767. But how could we improve the system further in terms of the speed?

In order to do this, I tried a similarity metric similar to jaccard coefficient - dice coefficient, I believe it will achieve the same MRR, but also improve the run-time efficiency, since it does not require to compute the intersection of two sets. The result by using dice coefficient is as follows:

```
Similarity Measure:DiceCoefficient
qId = 1 rel = 1 0.5000 Classical music may never be the most popular
music
qId = 1 rel = 0 0.5000 Everybody knows classical music when they hear
it
qId = 1 rel = 0 0.2222 Pop music has absorbed influences from most
other genres of popular music

qId = 2 rel = 1 0.4615 Climate change and energy use are two sides of
the same coin.
qId = 2 rel = 0 0.0000 Old wine and friends improve with age
qId = 2 rel = 0 0.0000 With clothes the new are the best, with friends
the old are the best

qId = 3 rel = 1 0.5000 The best mirror is an old friend
qId = 3 rel = 0 0.5000 My best friend is the one who brings out the
best in me
qId = 3 rel = 0 0.2500 The best antiques are old friends
```

```
qId = 4 rel = 0 0.4000 Wear a smile and have friends; wear a scowl and
have wrinkles
qId = 4 rel = 1 0.1818 If you see a friend without a smile, give him
one of yours
qId = 4 rel = 0 0.1667 Behind every girls smile is a best friend who
put it there

qId = 5 rel = 1 0.2222 Old friends are best
qId = 5 rel = 0 0.1818 Old wine and friends improve with age
qId = 5 rel = 0 0.1818 With clothes the new are the best, with friends
the old are the best

 (MRR) Mean Reciprocal Rank ::0.9
Total time taken: 0.833
```

WE can see that by using dice coefficient, we can not only achieve good MRR, but also pay the least computational cost.

Conclusion

In this experiment, I get the experience about how to implement a simple document retrieval system under the UIMA framework. I also gain the knowledge about how to tune an information system in order to improve performance and run-time efficiency by doing error analysis in a systematic way.