INFORMATION RETRIEVAL ASSIGNMENT - 2

- ANKIT MISHRA & RAJAT PRAKASH(95)

Question 1

Preprocessing

- **Dataset loading**

A folder Humour, Hist, Media, Food is created to store all the files from zip. Latin - 1 encoding is used on text from 1133 files and loaded into python.

- **Preprocessing**

First, Filtering is done which is followed by Querying.

Using NLTK library, we have done following steps to filter and clean the textual data

○ Conversion to lowercase alphabets and tokenization
○ Replacement of contractions with actual words. We used a general dictionary filled with all the well-known contractions. Source for this dictionary : dictionary
○ Removal of stop words, removal of emoji.
○ Removal of punctuations and unwanted characters from the textual data using the library.
○ Initially, we tried stemming each token. The word belongs in the dictionary or not won't affect the search algorithms. Hence, we switched to stemming of tokens instead of lemmatization.
○ Using pickle library, we have stored all the document name, original texts, filtered and cleaned text. pickle file is generated in order to use them in future efficiently.

Index Creation

At the outset, we sorted our dictionary list by name. It helped in index creation and simplified the process. Further, a mapping of document-to-document ID is generated. It is stored for future usage. For each of the terms, we have Posting lists.

Now, we have stored both in following two files -

1. Index.pkl (Index Mapping pickle file)
2. docIds.pkl (Document IDs Mapping pickle file)

INFORMATION RETRIEVAL ASSIGNMENT - 2

- ANKIT MISHRA & RAJAT PRAKASH(95)

Methodology

1. **Jaccard Coefficient**

   Jaccard Coefficient is calculated by intersection and union after preprocessing.

2. **TF-IDF Matrix**
   There are 5 weighting schemes used -

Weighting Scheme	TF Weight
Binary	0,1
Raw count	f(t,d)
Term frequency	f(t,d)/$\sum f(t', d)$
Log normalization	log(1+f(t,d))
Double normalization	0.5+0.5*(f(t,d)/ max(f(t',d))

a) **Binary** - This is the most fundamental weighting scheme. Here, relevance depends upon the existence of a word. It is simple and with minimum calculation, it can return results on documents. It will work good on documents with similar frequency of words. Further, its disadvantage is that it is judging the relevance without even considering the frequency of a term and total size of document.It also ignores the ordering of words. It will lead to wrong results in multiple scenarios.

b) **Raw Count** - This weighting scheme overcomes the problem with binary scheme. And, now can assign weight/relevance to the terms on the basis of the document size. It will have more exhaustive calculation/iteration. But, it will be able to assign relevance to large documents with more accuracy. Its cons is that it won't be able to classify terms relevance on the basis of size of documents. It also ignores the ordering of words.

c) **Term Frequency** - Term frequency is a good weighting scheme, It took the next step and consider the frequency of each term in the document to assign weights. It treats the document as famous bag of Words model, instead of considering the order of words. It will work efficiently with no stop words and similar document size. Its cons is that if there are multiple documents with high order of difference in size, then the relevance assigned will be biassed. There is a need of IDF technique, to make it work in such cases.

d) **Log normalisation** - It is weighting scheme which works on normalization of tf-weights of all term frequencies in a document by log of the frequencies. Suppose, A has frequency of 2 and, another word B has frequency of 20. Numerically, freq(B) is 10 times freq(B). But, experiments says that the effect/relevance of B is 3-4 times. The Log Normalisation will work in such cases and provides more accurate relevance. High Time and Space Complexity.

e) **Double Normalization** - It is also termed as Maximum tf Normalization. It is using smoothing to normalise tf - weights of all terms occurring in a doc by max tf. Its role is to lower the effect of the second term. It nullifies the effect of higher frequencies of repeating words in longer documents. Meanwhile, there are two major disadvantages -
   - Any change in the list of stop word will change the ranking and term weightings.
   - Outliers (multiple occurrences of word which doesn't represent the document) will affect this weighting scheme.
   - High time complexity and space Complexity

Question 2

Preprocessing

Slicing the data to store -
- Rows with QID = 4
- relevance judgment label (for sorting and to get max DCG)
- 75th column of data (for sorting, to rank and get precision/recall)

Methodology & Output

1. First column is relevance judgment labels taken as relevance scores.
2. There are 4 relevant judgments : {0, 1, 2, 3, 4}. At first, the data is arranged to get max DCG. We used sorting in reverse order according to relevance to achieve it (Basic Analysis).
   Further, the arrangements will still remain in sorted order is calculated by taking permutations (i.e. All possible permutations in which the relevance score will remain sorted)
   Total possible arrangements are :
   `19893497375938370599826047614905329896936840170566570588205180312704`
   `857992695193482412686565431050240000000000000000000000`
3. To calculate the nDCG, we first calculated the DCG of the list of docs. Then, we calculated max DCG (using the sorting technique mentioned above). Finally, we took the ratio of DCG and max DCG .
   Hence, nDCG = DCG/ max DCG.
   a) At 50, nDCG = `0.5253808413557646`
   b) Whole Document, nDCG = `0.5979226516897831`
4. Precision-Recall plot with qid = 4

INFORMATION RETRIEVAL ASSIGNMENT - 2

- ANKIT MISHRA & RAJAT PRAKASH(95)

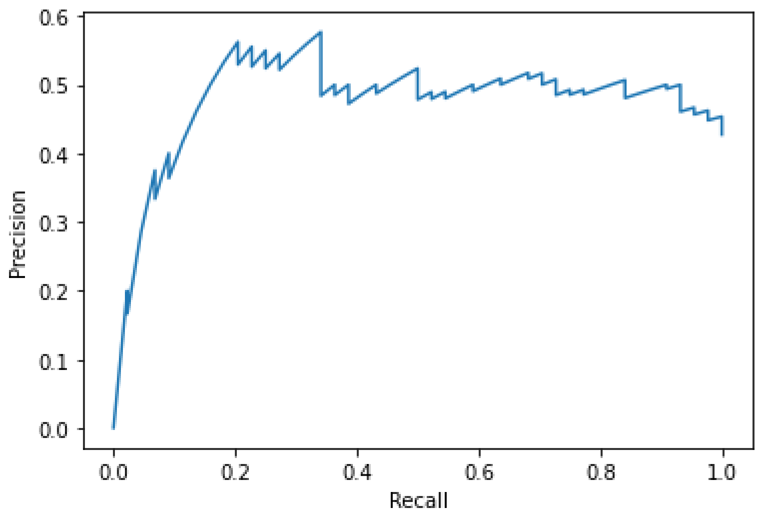

Fig 2.4 : Recall vs Precision (ranked according to WholeDocument attribute)

Question 3

Preprocessing

Using NLTK library, we have done following steps to filter and clean the textual data

- Conversion to lowercase alphabets and tokenization
- Replacement of contractions with actual words. We used a general dictionary filled with all the well-known contractions. Source for this dictionary : dictionary
- Removal of stop words, removal of emoji.
- Removal of punctuations and unwanted characters from the textual data using the library.
- Initially, we tried stemming each token. The word belongs in the dictionary or not won't affect the search algorithms. Hence, we switched to stemming of tokens instead of lemmatization.
- Using pickle library, we have stored all the document name, original texts, filtered and cleaned text. pickle file is generated in order to use them in future efficiently.

Methodology

INFORMATION RETRIEVAL ASSIGNMENT - 2

- ANKIT MISHRA & RAJAT PRAKASH(95)

After preprocessing, TF-ICF and Selecting Top-K features is done.
Nested_posting_list is created from the data.

Following steps are followed -

1. Term Frequency calculation (occurrences of a term in all documents for a particular class)
2. Class Frequency calculation (Count of classes in which that term occurs)
3. Inverse-Class Frequency calculation [log(N / CF), N : no. of classes]
4. Dataset splitting insequential order, for instance, choosing the first 800 documents in the train set and last 200 in the test set for the train: test ratio of 80:20.
5. Using TF-ICF scoring technique for efficient feature selection. Select the top k features for each class. Further, the effective vocabulary shall be the union of the top k features of each class.
6. For each class, Naive Bayes Model is trained on the training data.
7. Model testing on testing data and report the confusion matrix and overall accuracy.
8. Above steps are performed on 50:50, 70:30, and 80:20 training and testing split ratios.

Result for various splits -

For Split 50:50 ------------------------------
Top K features for each class are ----------------
['talk.politics.misc', 'ca.polit', 'cramer', 'soc.men', 'cramer.com']
['sci.m', 'geb.pitt.edu', 'rec.food.cook', 'n3jxp', 'chastiti']
['sci.spac', 'sci.astro', 'henry.toronto.edu', 'prb.digex.com', 'spacecraft']
['comp.graph', 'comp.graphics.anim', 'vga', 'polygon', 'tiff']
['rec.sport.hockey', 'nhl', 'hockey', 'playoff', 'bruin']

The accuracy of the Model is- 99.56
Confusion Matrix is-
[503. 0. 1. 0. 0.]
[2. 488. 0. 0. 0.]
[1. 1. 479. 0. 0.]
[2. 3. 1. 521. 0.]

INFORMATION RETRIEVAL ASSIGNMENT - 2

- ANKIT MISHRA & RAJAT PRAKASH(95)

[0.   0.   0.   0. 498.]

For Split 70:30 ------------------------------
Top K features for each class are ----------------
['talk.politics.misc', 'ca.polit', 'talk.religion.misc', 'clayton', 'cramer']
['sci.m', 'geb.pitt.edu', 'n3jxp', 'chastiti', 'geb.dsl.pitt.edu']
['sci.spac', 'sci.astro', 'spacecraft', 'henry.toronto.edu', 'orbit']
['comp.graph', 'comp.graphics.anim', 'vga', 'tiff', 'polygon']
['rec.sport.hockey', 'nhl', 'hockey', 'playoff', 'bruin']

The accuracy of the Model is- 99.46666666666667
Confusion Matrix is-
[314.   0.   0.   0.   0.]
[1. 302.   0.   3.   0.]
[1.   0. 299.   0.   0.]
[2.   0.   1. 269.   0.]
[0.   0.   0.   0. 308.]

For Split 80:20 ------------------------------
Top K features for each class are ----------------
['talk.politics.misc', 'ca.polit', 'cramer', 'talk.religion.misc', 'cramer.com']
['sci.m', 'geb.pitt.edu', 'rec.food.cook', 'n3jxp', 'chastiti']
['sci.spac', 'henry.toronto.edu', 'sci.astro', 'nsmca.alaska.edu', 'orbit']
['comp.graph', 'comp.graphics.anim', 'vga', 'polygon', 'pov']
['rec.sport.hockey', 'nhl', 'hockey', 'playoff', 'bruin']

The accuracy of the Model is- 99.8
Confusion Matrix is-
[192.   0.   1.   0.   0.]
[0. 200.   1.   0.   0.]
[0.   0. 193.   0.   0.]
[0.   0.   0. 198.   0.]
[0.   0.   0.   0. 215.]
