Text Retrieval Lab

Part 1: TF-IDF Text Retrieval

Task 1: Stop Word Removal

Go into lab1-2021 folder

Running `stop stoplist50 docOrig\AbassiM.txt` in the command prompt displays the output

run `stopScript.bat return` in the command prompt places all *files.stp* into DocStop:

We should now have 112 additional *.stp* files

Q1: What is the percentage reduction in the number of words in a document as a consequence of stop-word removal? Specifically, what is the reduction in the case of the file AgricoleW.txt?

A1: The percentage reduction in no. of words using stop-word removal would usually be around 20-30% of words.

From the file *AgricoleW*: $405 - 303 = 102$ words removed, that is $\frac{102}{405} \times 100 = 25.2\%$

```
Enter source file path: docOrig/AgricoleW.txt

Total characters = 2551
Total words      = 405
Total lines      = 36
```

```
Enter source file path: docStop/AgricoleW.stp

Total characters = 2070
Total words      = 303
Total lines      = 51
```

Task 2: Stemming

We can create *file.stm* from *file.stp*, printing results in command prompt:

`porter-stemmer docStop\AbassiM.stp`

We can use `stemScript.bat return` to store all *stm* files into DocStem.

We should now have 112 additional *.stm* files

Q2: Find the file AgricoleW.stm. What are the results of applying the porter-stemmer to the words communications, sophisticated and transmissions?

A2:

Communications → commun

sophisticated → sophist

transmissions → transmiss

Task 3: Create Document Index Files

We have *index.c* in the folder that we can use to index a list of files under the name: *textFileList*, *stopFileList* and *stemFileList*:

Code:

```
index textFileList > textIndex
index stopFileList > stopIndex
index stemFileList > stemIndex
```

We should now have 3 additional files with the names above

Q3.1: What are the document lengths of the documents: *docOrig\DongP.txt*, *docStop\DongP.stp* and *docStem\DongP.stm*? Why are they different?

A3.1: From the three Index Files:

DongP.txt: 42.396210 (from *TextIndex*)

DongP.stp: 42.392876 (from *StopIndex*)

DongP.stm: 40.547958 (from *StemIndex*)

It can be decimal values because it based on TF-IDF values instead of no. of documents which are integers.

They are different because the documents lengths are dependent on the term weights and because some terms are combined in the *.stm* files, the overall weight has decreased, resulting in a smaller document length.

Q3.2: Why is the difference between the document lengths of *docStem\DongP.stm* and *docOrig\DongP.txt* greater than the difference between the document lengths of *docStop\DongP.stp* and *docOrig\DongP.txt*?

A3.2: The difference between the *.stm* and *.txt* is greater than *.stp* and *.txt* because more words/tokens/terms were lost between the *.txt* and *.stp*.

Q4: The IDF of the term *adjacent* is 0.009. Why is it so close to zero?

A4: Because, out of the 112 documents, *adjacent* appears in 111 documents, so it has less weight, and is less useful:

```
141 word=adjacent wordCount=118 docCount=111 IDF=0.008969
```

Q5: Find the word *algorithm* in the three index files. Explain why the entries for this word are different in the three files.

A5:

```
185 word=algorithm wordCount=14 docCount=7 IDF=2.772589
   1 docName=docOrig\EftekhariS.txt count=2 weight=5.545177
   2 docName=docOrig\LokCY.txt count=1 weight=2.772589
   3 docName=docOrig\NgTA.txt count=1 weight=2.772589
   4 docName=docOrig\PangG.txt count=2 weight=5.545177
   5 docName=docOrig\RajaI.txt count=5 weight=13.862944
   6 docName=docOrig\WangMY.txt count=2 weight=5.545177
   7 docName=docOrig\ZhangJ.txt count=1 weight=2.772589
```

```
184 word=algorithm wordCount=14 docCount=7 IDF=2.772589
   1 docName=docStop\EftekhariS.stp count=2 weight=5.545177
   2 docName=docStop\LokCY.stp count=1 weight=2.772589
   3 docName=docStop\NgTA.stp count=1 weight=2.772589
   4 docName=docStop\PangG.stp count=2 weight=5.545177
   5 docName=docStop\RajaI.stp count=5 weight=13.862944
   6 docName=docStop\WangMY.stp count=2 weight=5.545177
   7 docName=docStop\ZhangJ.stp count=1 weight=2.772589
```

```
152 word=algorithm wordCount=36 docCount=15 IDF=2.010449
   1 docName=docStem\AliR.stm count=2 weight=4.020897
   2 docName=docStem\BenHasineA.stm count=4 weight=8.041795
   3 docName=docStem\BradyE.stm count=2 weight=4.020897
   4 docName=docStem\BronksA.stm count=2 weight=4.020897
   5 docName=docStem\ChanWK.stm count=1 weight=2.010449
   6 docName=docStem\EftekhariS.stm count=5 weight=10.052243
   7 docName=docStem\LokCY.stm count=4 weight=8.041795
   8 docName=docStem\MohdNasir.stm count=2 weight=4.020897
   9 docName=docStem\NgTA.stm count=1 weight=2.010449
  10 docName=docStem\PangG.stm count=2 weight=4.020897
  11 docName=docStem\PargeterA.stm count=1 weight=2.010449
  12 docName=docStem\RajaI.stm count=6 weight=12.062693
  13 docName=docStem\SodenJ.stm count=1 weight=2.010449
  14 docName=docStem\WangMY.stm count=2 weight=4.020897
  15 docName=docStem\ZhangJ.stm count=1 weight=2.010449
```

TextIndex includes stop words that aren't remove as terms, *StopIndex* is similar to *TextIndex* without the stop words while *StemIndex* might combine different forms of words into 1, decreasing the index count and amount of entries.

Task 4: Retrieval

The query file contains the text: *communication and networks*

For retrieval, we use stopping and stemming to the query first:

`stop stoplist50 query > query.stp`

`porter-stemmer query.stp > query.stm`

We should have 2 additional files: *query.stp* and *query.stm*

We can now do retrieval with the *retrieve.exe* file:

`retrieve textIndex query > RetrieveOrg`

`retrieve stopIndex query.stp > RetrieveStop`

`retrieve stemIndex query.stm > RetrieveStem`

We now have 3 additionaly retrieval files with different best document results.

Q6: Compare the results of these two searches with the result for the original raw text files. What do you conclude?

A6:

```
Best document is docOrig\TomlinsonM.txt (0.152037)
Best document is docStop\TomlinsonM.stp (0.152309)
Best document is docStem\YiuMLM.stm (0.261187)
```

It is concluded that after stemming, other terms in other documents were stemmed into the stemmed version of 'communication and networks', therefore, another document different from the stop and original version was selected as the best document for the query. As for the stop result, it has similar values because only useless and 'noise words' were removed.

Task 5: 2 Additional Queries for Task 4

Creating own queries:

q1: computers and laptops

q2: is digital technology the best?

Stopping and stemming both files:

`stop stoplist50 q1 > q1.stp` + `porter-stemmer q1.stp > q1.stm`

`stop stoplist50 q2 > q2.stp` + `porter-stemmer q2.stp > q2.stm`

We should now have 4 additional files created.

`retrieve textIndex q1 > RetrieveOrg_q1` + `retrieve stopIndex q1.stp > RetrieveStop_q1` + `retrieve stemIndex q1.stm > RetrieveStem_q1`

`retrieve textIndex q2 > RetrieveOrg_q2` + `retrieve stopIndex q2.stp > RetrieveStop_q2` + `retrieve stemIndex q2.stm > RetrieveStem_q2`

And another 6 additional files. If we don't want to output any files, no need to direct it with `>` .

Result/ Output:

q1:

```
Best document is docOrig\RobertsSM.txt (0.077057)
Best document is docStop\RobertsSM.stp (0.077285)
Best document is docStem\RossiterJ.stm (0.156027)
```

q2:

```
Best document is docOrig\NyagoA.txt (0.230118)
Best document is docStop\NyagoA.stp (0.176783)
Best document is docStem\NyagoA.stm (0.158520)
```

q2 has larger differences between original and stop files because the q2 has more stop words than q1

Part 2: Latent Semantic Analysis

Task 1: Create Word-Document Matrix

executable doc2vec.exe creates matrix W. We will apply this program to the stemmed documents with the command:

`doc2vec stemFileList > WDM`

creating a document vector for each document in *docStem* folder and stacks them to create matrix file *WDM*. We should now have a new *WDM* file in lab1-2021.

Task 2: Apply SVD to Word-Document Matrix

in MATLAB, the commands:

`W = load('WDM');` (reads the data in *WDM* into the MATLAB matrix W)

`[U,S,V]=svd(W);` (runs SVD on W, decomposing it as $W = USV^T$.)

Q1: Are the matrices U and V as you would expect? Explain.

A1:

U and V satisfies $UU^T = I = U^TU, VV^T = I = V^TV$

Yes the S elements are verified to be correct:

112x2743 double

	1	2	3	4	5	6
1	274.5965	0	0	0	0	0
2	0	53.7182	0	0	0	0
3	0	0	47.1338	0	0	0
4	0	0	0	44.0433	0	0
5	0	0	0	0	37.6685	0
6	0	0	0	0	0	33.4974
7	0	0	0	0	0	0
8	0	0	0	0	0	0

and satisfies $s_1 \geq s_2 \geq ... \geq s_N$

Q2: What are the values of the first 3 diagonal entries in S?

A2:

1. 274.596487768020
2. 53.7182065440782
3. 47.1338164059137

These are singular vectors, or 'latent semantic classes', that corresponds to the columns of V. Getting the 1st, 2nd and 3rd column of V:

```
sV1 = V(:,1);
sV2 = V(:,2);
sV3 = V(:,3);
```

The most important words that determine the interpretation of vector 1 are the biggest values (positive or negative). We also want to know the position/index of the biggest values so we know which word it corresponds to:

```
[m1,am1] = mink(sV1,3)
```

```
m1 =
   -0.6151
   -0.2492
   -0.2451

am1 =
   1902
   1723
   2325
```

`[m2,am2] = mink(sV2,3)`

```
m2 =
   -0.1535
   -0.1191
   -0.1090
am2 =
   608
   2043
   576
```

`[m3,am3] = mink(sV3,3)`

```
m3 =
   -0.1708
   -0.0911
   -0.0895
am3 =
   2268
   608
   2181
```

We used *min* instead of max here because, during our session, Matlab seems to think that *max* is *min* and vice versa.

Q3: Find the three most significant words for each of the singular vectors sv1, sv2 and sv3. What is your interpretation of the corresponding semantic classes?

A3:

These are the summaries of an approximate/abstract concept:

sV1:

1: project

2: outcome

3: student

sV2:

1: data

2: resonate

3: couple

sV3:

1: speech

2: data

3: should

Interpretation:

sV1: project specifications/ intro

sV2: results and analysis

sV3: conclusion