NLP Lab 2

Dixitha Kasturi
dkasturi@syr.edu

For exploring a book from the Gutenberg corpus, I chose **Leaves by Witman**

Approach :

1) tokenize text
2) convert tokens into lowercase
3) With Raw Frequency
    a) create bigram finder and scorer
    b) apply re filter for to remove non-alphanumeric tokens
    c) remove stopwords
    d) remove low frequency words
4) repeat steps a-d with pmi

Report :

Finder1 : scoring by Raw Frequency

I applied a bigram finder on the raw tokenized text. The bigrams that were generated had a lot of punctuations and non-alphabetic characters. These did not make a lot of sense. So I applied filters

```
((',', 'the'), 0.014785423690189745)
(('of', 'the'), 0.008458388348603562)
((',', 'and'), 0.008404769405030797)
((',', 'i'), 0.006762689258114892)
(('in', 'the'), 0.004054932607690297)
(('and', 'the'), 0.0037064094744673295)
((',', '('), 0.002292209837735672)
((',', 'to'), 0.0020442222237116374)
(('to', 'the'), 0.002037519855765042)
((',', 'or'), 0.002004008016032064)
(('me', ','), 0.0019771985442456820)
(('i', 'see'), 0.0019570914404058956)
(('on', 'the'), 0.0018096393455807936)
(("'d", ','), 0.0018029369776341982)
((',', 'with'), 0.0016688896187022875)
(('from', 'the'), 0.0016487825148625008)
(('you', ','), 0.0014678185803044216)
((',', 'but'), 0.0014477114764646349)
((',', ')'), 0.0014276043726248484)
(('it', 'is'), 0.0013333771221372511)
```

a) *Filter to remove non-alphabetical tokens* : To remove the punctuations and non-alphabetic characters. The results were much better than the ones in the first iteration.

```
(('of', 'the'), 0.008458388348603562)
(('in', 'the'), 0.004054932607690297)
(('and', 'the'), 0.0037064094744673295)
(('to', 'the'), 0.002037519855765042)
(('i', 'see'), 0.0019570914404058956)
(('on', 'the'), 0.0018096393455807936)
(('from', 'the'), 0.0016487825148625008)
(('it', 'is'), 0.001333771221372511)
(('with', 'the'), 0.0013002593816395334)
(('i', 'am'), 0.0012801522777997466)
(('to', 'me'), 0.0011729143906542181)
(('all', 'the'), 0.0011058907111882628)
(('by', 'the'), 0.00100535519198933)
(('out', 'of'), 0.0009986528240427343)
(('the', 'earth'), 0.0009785457202029478)
(('see', 'the'), 0.0009048196727903968)
(('do', 'not'), 0.0008512007292176325)
(('of', 'my'), 0.0008512007292176325)
(('of', 'all'), 0.0007908794176982728)
(('and', 'i'), 0.0007841770497516772)
```

b) *Filter to remove stopwords*: Generally stopwords include articles, prepositions , conjunctions etc, which don't explain the context of the text. Removing stop words resulted in different and more insight giving bigrams

```
(('young', 'men'), 0.0002077734063444615)
(('every', 'one'), 0.00016085683071829279)
(('old', 'age'), 0.00014745209482510172)
(('young', 'man'), 0.00014745209482510172)
(('old', 'man'), 0.00011394025509212404)
(('open', 'air'), 0.00011394025509212404)
(('every', 'day'), 9.383315125233744e-05)
(('every', 'thing'), 9.383315125233744e-05)
(('whole', 'earth'), 8.042841535914639e-05)
(('new', 'world'), 7.372604741255086e-05)
(('one', 'side'), 6.702367946595533e-05)
(('thousand', 'years'), 6.702367946595533e-05)
(('let', 'others'), 6.032131151935979e-05)
(('let', 'us'), 6.032131151935979e-05)
(('little', 'child'), 6.032131151935979e-05)
(('western', 'sea'), 6.032131151935979e-05)
(('comes', 'back'), 5.3618943572764255e-05)
(('old', 'men'), 5.3618943572764255e-05)
(('walt', 'whitman'), 5.3618943572764255e-05)
(('go', 'forth'), 4.691657562616872e-05)
```

c) *Filter to remove low frequency bigrams*: Upon trying, 3-5 threshold the results were not altered by much.

```
(('young', 'men'), 0.0002077734063444615)
(('every', 'one'), 0.00016085683071829279)
(('old', 'age'), 0.00014745209482510172)
(('young', 'man'), 0.00014745209482510172)
(('old', 'man'), 0.00011394025509212404)
(('open', 'air'), 0.00011394025509212404)
(('every', 'day'), 9.383315125233744e-05)
(('every', 'thing'), 9.383315125233744e-05)
(('whole', 'earth'), 8.042841535914639e-05)
(('new', 'world'), 7.372604741255086e-05)
(('one', 'side'), 6.702367946595533e-05)
(('thousand', 'years'), 6.702367946595533e-05)
(('let', 'others'), 6.032131151935979e-05)
(('let', 'us'), 6.032131151935979e-05)
(('little', 'child'), 6.032131151935979e-05)
(('western', 'sea'), 6.032131151935979e-05)
(('comes', 'back'), 5.3618943572764255e-05)
(('old', 'men'), 5.3618943572764255e-05)
(('walt', 'whitman'), 5.3618943572764255e-05)
(('go', 'forth'), 4.691657562616872e-05)
```

Finder1 : scoring by Pointwise Mutual Information:

The bigrams that were generated showed different results as compared to the ones that were generated by raw frequency. PMI takes into account the likelihood of the two words occurring together rather than just taking the high frequency bigrams. These top 20 bigrams without a lot of filtering made sense. So I initially skipped filtering out non-alphabetical tokens and stopwords. But when the frequency filter was applied, a lot of stopwords and non-alphabetic characters were present

```
(('walt', 'whitman'), 14.016972678093055)
(('shapes', 'arise'), 11.312428561619225)
(('thousand', 'miles'), 10.572187835420157)
(('became', 'part'), 10.44993208536916)
(('[', 'book'), 10.39707638726251)
(('thou', 'knowest'), 9.479538547454482)
(('why', 'should'), 9.438704829945907)
(('due', 'time'), 9.128003990481796)
(('white', 'hair'), 8.757042640837946)
(('open', 'air'), 8.715939807517042)
(('dear', 'son'), 8.653567947229531)
(('dear', 'friend'), 8.617042071204416)
(('new', 'ones'), 8.586984837348238)
(('me.', '}'), 8.583271334549174)
(('songs.', '}'), 8.583271334549174)
(('them.', '}'), 8.583271334549172)
(('you.', '}'), 8.583271334549172)
(('he', 'sees'), 8.552691659794359)
(('thousand', 'years'), 8.444083009672472)
(('any', 'thing'), 8.405537966010705)
```

To remove these I filtered out non-alphabetical tokens and stopwords. Though the results of these two filterings didn't show what difference it made, When frequency filtering was finally applied, it made more sense and results were different.

a) *Filter to remove non-alphabetical tokens* : To remove the punctuations and non-alphabetic characters. The results were much better than the ones in the first iteration.

```
(('abject', 'louse'), 17.186897679535367)
(('abroad', 'swift-rising'), 17.186897679535367)
(('accumulating', 'undirected'), 17.186897679535367)
(('agnus', 'dei'), 17.186897679535367)
(('ague', 'convulse'), 17.186897679535367)
(('aimedst', 'highly'), 17.186897679535367)
(('alexandrian', 'pharos'), 17.186897679535367)
(('ambushes', 'opponents'), 17.186897679535367)
(('andirons', 'straddle'), 17.186897679535367)
(('artless', 'plaints'), 17.186897679535367)
(('au', 'monde'), 17.186897679535367)
(('balsamic', 'busses'), 17.186897679535367)
(('beach-waves', 'combing'), 17.186897679535367)
(('beaver', 'pats'), 17.186897679535367)
(('behaving', 'licentious'), 17.186897679535367)
(('bitterest', 'envy.'), 17.186897679535367)
(('block', 'swags'), 17.186897679535367)
(('bokh', 'horse-herd'), 17.186897679535367)
(('brownish', 'woolen'), 17.186897679535367)
(('bulging', 'store-house'), 17.186897679535367)
```

b) *Filter to remove stopwords:* Generally stopwords include articles, prepositions , conjunctions etc, which don't explain the context of the text. Removing stop words resulted in different and more insight giving bigrams

```
(('abject', 'louse'), 17.186897679535367)
(('abroad', 'swift-rising'), 17.186897679535367)
(('accumulating', 'undirected'), 17.186897679535367)
(('agnus', 'dei'), 17.186897679535367)
(('ague', 'convulse'), 17.186897679535367)
(('aimedst', 'highly'), 17.186897679535367)
(('alexandrian', 'pharos'), 17.186897679535367)
(('ambushes', 'opponents'), 17.186897679535367)
(('andirons', 'straddle'), 17.186897679535367)
(('artless', 'plaints'), 17.186897679535367)
(('au', 'monde'), 17.186897679535367)
(('balsamic', 'busses'), 17.186897679535367)
(('beach-waves', 'combing'), 17.186897679535367)
(('beaver', 'pats'), 17.186897679535367)
(('behaving', 'licentious'), 17.186897679535367)
(('bitterest', 'envy.'), 17.186897679535367)
(('block', 'swags'), 17.186897679535367)
(('bokh', 'horse-herd'), 17.186897679535367)
(('brownish', 'woolen'), 17.186897679535367)
(('bulging', 'store-house'), 17.186897679535367)
```

c) *Filter to remove low frequency bigrams*: Upon trying, 4-6 threshold the results were not altered by much.

```
(('walt', 'whitman'), 14.016972678093055)
(('shapes', 'arise'), 11.312428561619225)
(('thousand', 'miles'), 10.572187835420157)
(('became', 'part'), 10.44993208536916)
(('thou', 'knowest'), 9.479538547454482)
(('due', 'time'), 9.128003990481796)
(('white', 'hair'), 8.757042640837946)
(('open', 'air'), 8.715939807517042)
(('dear', 'son'), 8.653567947229531)
(('dear', 'friend'), 8.617042071204416)
(('new', 'ones'), 8.586984837348238)
(('thousand', 'years'), 8.444083009672472)
(('thou', 'hast'), 8.380002873903567)
(('none', 'else'), 8.354007665370624)
(('comes', 'back'), 8.049586288834465)
(('old', 'age'), 8.019479533703628)
(('great', 'idea'), 7.990500466731861)
(('western', 'sea'), 7.948492940210288)
(('little', 'child'), 7.919590779938201)
(('many', 'times'), 7.9005049388073)
```

CONCLUSION :

For the book I analyzed, using all 3 filters (non-alphabetical, stopwords and low frequency) gave better results for both Raw frequency and PMI scoring. Observing top 20 bigrams helped in deciding whether or not to apply filters. PMI gave better results without filters when compared to raw frequency scoring. But applying filter on PMI enhanced the results. The results from PMI without stopword filtering also did make sense, but a more refined version was achieved with filtering.