**1.) trec results**

**I wasn’t able to get trec working on my computer. Every time I tried to use it, it would throw an error. When talking about PRF I used the program I wrote for part 2 to write my observations.**

**2.) Observations about PRF**

When running my program with PRF, using words from the relevant document. What I noticed was that the first document that was returned previously is rated much higher since now the query is using more words in that document. So I think the top document used for query expansion should be removed from the returned results in an actual application. The other thing I noticed is that how python sorts the results impacts this a lot. There might be a lot of words rated as 1.00 but if I only take the top 3 out of 9 words all rated as 1.00 you might not pick the most relevant words for query expansion. So what I noticed is it impacted our small set of documents very little, and increased the rating of the document we did query expansion on. The parameter x I think is very interesting. If you take more terms for query expansion you may dilute the search you actually want. For example if I have a 1 word query, and I do query expansion for 15 words, I may not be getting relevant results due to the fact that those 15 words may not be relevant to the search of my 1 word. On the other hand if you have x as 1 or 2, and only do query expansion by a small number it may have very little to no effect on the results of the original query. So there is definitely some balance and testing that needs to be done to select a good x for the collection you are running the query expansion on.

**3.) Output**

Output from Pseudo-relevance feedback (PRF)

Query: battery

(5, 1.0)

(10, 0)

(9, 0)

(8, 0)

(7, 0)

(6, 0)

(4, 0)

(3, 0)

(2, 0)

(1, 0)

New Query: battery teen tough he

(5, 4.301029995663981)

(10, 0)

(9, 0)

(8, 0)

(7, 0)

(6, 0)

(4, 0)

(3, 0)

(2, 0)

(1, 0)

Query: screen

(3, 0.44465780490343443)

(6, 0.39164905395343774)

(8, 0.3010299956639812)

(2, 0.3010299956639812)

(1, 0.3010299956639812)

(10, 0)

(9, 0)

(7, 0)

(5, 0)

(4, 0)

New Query: screen port properli charg

(3, 4.523839050951059)

(6, 0.39164905395343774)

(8, 0.3010299956639812)

(2, 0.3010299956639812)

(1, 0.3010299956639812)

(10, 0)

(9, 0)

(7, 0)

(5, 0)

(4, 0)

Query: speed

(10, 0)

(9, 0)

(8, 0)

(7, 0)

(6, 0)

(5, 0)

(4, 0)

(3, 0)

(2, 0)

(1, 0)

Query speed not found during search.

Creating query of synonyms . . .

New Query: speed fast quick acceleration

(6, 0.6989700043360189)

(2, 0.6989700043360189)

(10, 0)

(9, 0)

(8, 0)

(7, 0)

(5, 0)

(4, 0)

(3, 0)

(1, 0)

Output from Kappa Statistic

name: Yuen-qrels.txt value: 0.700 Fair Agreement

name: patil-qrels.txt value: 1.000 Good Agreement

name: aggarwal-qrels.txt value: 1.000 Good Agreement

name: jha-qrels.txt value: 0.850 Good Agreement

name: enigala-qrels.txt value: 1.000 Good Agreement

name: Tse-qrels.txt value: 0.780 Fair Agreement

name: maski-qrels.txt value: 0.489 Weak Agreement

name: waterman-qrels.txt value: 0.850 Good Agreement

name: chau-qrels.txt value: 0.659 Weak Agreement

name: Khvalchik-qrels.txt value: 0.769 Fair Agreement

name: hoff-qrels.txt value: 0.850 Good Agreement

name: Li-qrels.txt value: 0.850 Good Agreement

name: Chen-qrels.txt value: 0.780 Fair Agreement

name: thomas-qrels.txt value: 0.923 Good Agreement

name: zhang-qrels.txt value: 0.850 Good Agreement

name: Wishoff-qrels.txt value: 1.000 Good Agreement

name: trubov-qrels.txt value: 0.700 Fair Agreement

name: pithyaach-qrels.txt value: 0.788 Fair Agreement