Cole Lamers
CS 738
PA4 Report

- Unlabeled
  - articles100
    - I noticed that changing the Jaccard threshold did not matter for these. The results returned were all the same with similar scores. Now this can mean that the dataset was too small to make broader comparisons, or that there really are only a few documents that seems to be similar. For all threshold scores .3 thru .9, they all found the items in the articles_100.truth file
  - arcticles1000
    - Here .5, .7, and .9 had found the same files in the articles_1000.truth file. It was only .3 that actually ended up finding a lot more similarities between files. Now part of this could possibly be because I also took the effort to filter out articles, suffixes, and auxiliary verbs. My similarity scores may have been skewed just slightly more because out of the total number of words being removed that fell into this category would've influenced the scores returned.
  - articles2500
    - With this set, the thresholds .7 and .9 matched the articles in the articles_2500.truth file exactly. However with a .5 threshold, there was one extra one that matched, files t4326 and t4328. I think this was partially influenced because of my decision to parse the articles, auxiliaries, and suffixes. There are a few subtleties that stand out but for the most part it is different, aside from some very similar sentences with minor differences. It seems like some news headline that was injected/utilized widespread and was slightly altered between the articles.
    - When I reran the application ignoring the removal of special words, the .5 threshold no longer contained the extra comparison.
  - articles10000
    - For this one, the .7 and .9 thresholds matched the articles_10000.truth file's data. Again, the .5 threshold and reviewing the files, they were t4326 and t4328 again. The same result also occurred when I ran it without checking the special words/chars and it was no longer in the .5 threshold files. With the .3 threshold though, it appears that there are numerous files throughout with similarities that it's impossible to manually parse through them. From what I've reviewed, these appear to have a couple things in common but for the most part are totally different.
- Labeled
  - NOTE: Images are of process run without pruning of word types.
  - setA

file1	file2	jaccard	set
g4pC_taska.txt	orig_taska.txt	0.9	a
orig_taska.txt	g0pE_taska.txt	0.85	a
g4pC_taska.txt	g0pE_taska.txt	0.75	a
g4pC_taska.txt	g2pC_taska.txt	0.5	a
orig_taska.txt	g2pC_taska.txt	0.5	a
g2pC_taska.txt	g0pE_taska.txt	0.5	a
g0pD_taska.txt	g2pC_taska.txt	0.45	a
g0pD_taska.txt	g4pC_taska.txt	0.3	a
g0pD_taska.txt	orig_taska.txt	0.3	a
g0pD_taska.txt	g0pE_taska.txt	0.3	a

- Here are the results I got from comparing setA files. Many seemed to be copies of one another as opposed to copies directly from the source file. Overall the data came back fairly accurate according to the JaccardSimilarity.csv file.

- setB

file1	file2	jaccard	set
g0pA_taskb.txt	orig_taskb.txt	0.4	b
g4pE_taskb.txt	orig_taskb.txt	0.3	b
g1pD_taskb.txt	g4pD_taskb.txt	0.3	b
orig_taskb.txt	g2pE_taskb.txt	0.3	b

  - What was interesting with this data set was running both parsing the word types and without, the last one actually was only included in the run without the additional parsing. Meaning, the additional pruning removed a false positive. From all the other items I was not expecting to see that result and it is interesting that the strictness of words checked are not always correlated with returned false positives. Additionally, the scores changed higher and lower for some.
- setC

file1	file2	jaccard	set
orig_taskc.txt	g0pB_taskc.txt	0.65	c
orig_taskc.txt	g4pB_taskc.txt	0.45	c
g4pE_taskc.txt	g0pB_taskc.txt	0.4	c
orig_taskc.txt	g0pD_taskc.txt	0.35	c
g3pA_taskc.txt	orig_taskc.txt	0.3	c
g3pA_taskc.txt	g0pB_taskc.txt	0.3	c
g4pB_taskc.txt	g2pA_taskc.txt	0.3	c

  - The results are very similar to the CSV file, and a similar affect happened like the previous one. When pruning of words occurred, the scores actually lowered or heightened slightly between these scores.
- setD

file1	file2	jaccard	set
orig_taskd.txt	g3pA_taskd.txt	1	d
g4pC_taskd.txt	orig_taskd.txt	0.9	d
g4pC_taskd.txt	g3pA_taskd.txt	0.9	d
orig_taskd.txt	g2pB_taskd.txt	0.65	d
g4pC_taskd.txt	g2pB_taskd.txt	0.6	d
g2pB_taskd.txt	g3pA_taskd.txt	0.6	d
g0pC_taskd.txt	g4pC_taskd.txt	0.45	d
g0pC_taskd.txt	orig_taskd.txt	0.45	d
g0pC_taskd.txt	g2pB_taskd.txt	0.45	d
g0pC_taskd.txt	g3pA_taskd.txt	0.45	d
g4pB_taskd.txt	g4pC_taskd.txt	0.35	d
g4pB_taskd.txt	orig_taskd.txt	0.35	d
g4pB_taskd.txt	g3pA_taskd.txt	0.35	d
g1pA_taskd.txt	orig_taskd.txt	0.35	d
g1pA_taskd.txt	g2pB_taskd.txt	0.35	d
g1pA_taskd.txt	g3pA_taskd.txt	0.35	d
g0pC_taskd.txt	g1pA_taskd.txt	0.3	d
g1pA_taskd.txt	g4pC_taskd.txt	0.3	d
g2pE_taskd.txt	g4pD_taskd.txt	0.3	d

- Of setD, it actually managed to find many more files with higher similarity scores when compared against one another. The interesting part I found was parsing for the special words showed the scores actually dropped across the board. What is interesting about that is because more similar words would've been removed, and when that should've made it easier to find duplicates, it actually harmed the approach for finding an exact match (like these results did). So the pruning in effect wouldn't have been able to detect an exact match that it should have like in the JaccardSimilarity.csv.

- setE

file1	file2	jaccard	set
orig_taske.txt	g4pB_taske.txt	0.7	e
g4pD_taske.txt	g4pB_taske.txt	0.55	e
orig_taske.txt	g4pD_taske.txt	0.5	e
g1pB_taske.txt	g4pC_taske.txt	0.45	e
g4pC_taske.txt	g0pC_taske.txt	0.45	e
g2pB_taske.txt	g4pC_taske.txt	0.4	e
g0pE_taske.txt	g2pB_taske.txt	0.35	e
g0pE_taske.txt	g1pB_taske.txt	0.35	e
g0pE_taske.txt	g3pB_taske.txt	0.35	e
g2pB_taske.txt	g1pB_taske.txt	0.35	e
g2pB_taske.txt	orig_taske.txt	0.35	e
g1pB_taske.txt	g0pC_taske.txt	0.35	e
orig_taske.txt	g4pC_taske.txt	0.35	e
g0pE_taske.txt	g0pC_taske.txt	0.3	e
g2pB_taske.txt	g0pC_taske.txt	0.3	e

  - For setE, it was a similar scenario. Although there was one anomaly that stood out to me and that was g4pB_taske.txt and orig_taske.txt. Here there is a high metric of .7. When you remove the special words again, this score actually shot up to .95 similarity. Meanwhile, almost all other scores actually dropped by about .05 to .20 points. In a way it is interesting how certain words like that in the English language can have so much of an affect on the ability to detect a duplicate or not. But in general as we can see of this data, there are many very similar ones, but one being a very clear duplicate with just a different kind of formatting within the files.

- Conclusion
  - Overall it was very interesting to compare the approaches of comparing the results. Removing auxiliaries and the like has enough of an influence on the detection of duplicates that it actually does seem valid to perform. Not to mention it's false positive rate appears to be more on the side of extremes where it is definitely right, or only slightly wrong. With just making comparisons to the words as is, it is more consistent, but seems to skew toward an average assumption over an accurate assumption, thus leading to high scores and more false positives.