Instructions:

To run dedupe, run: 'python product_dedup.py'
To evaluate the results, run: 'python eval.py'

Note: Keep all files in their original directories.

Solution:

I found that the best results were achieved when training on manufacturer, title, price, and description with recall weight equal to 0. The first thing I did was explored the manufacturer and title fields. I tried concatenating the two fields into one given that some entities omit some of these fields, but I did not see any increase in performance. I found that the best technique was to compare manufacturer fields and title fields as 'Strings.' Next, I looked a price and wrote a custom comparator to report a partial match if the minimum of the two prices is at least 75% of the maximum of the two prices. These partial match values are adjusted incrementally with percentage. Finally, I used the description field and compared the first 10 words of each description paragraph as a 'Text' field, which bolstered overall performance. Below are the results of various configurations I tried:

All of these results were measured after completing the active training for 20 iterations.

manufacturer, title, price ----------

Duplicates Found: 550

True Positives: 176
False Positives: 374
False Negatives: 1124

Precision: 0.32
Recall: 0.135384615385
F-measure: 0.19027027027

manufacturer_title, price ----------

Duplicates Found: 7850

True Positives: 563
False Positives: 7287
False Negatives: 737

Precision: 0.0717197452229
Recall: 0.433076923077

F-measure: 0.12306010929

manufacturer_title as text, price ----------

Duplicates Found: 4210

True Positives: 400
False Positives: 3810
False Negatives: 900

Precision: 0.0950118764846
Recall: 0.307692307692
F-measure: 0.145190562613

manufacturer (has missing), title, price ----------

Duplicates Found: 101

True Positives: 43
False Positives: 58
False Negatives: 1257

Precision: 0.425742574257
Recall: 0.0330769230769
F-measure: 0.0613847251963

manufacturer, title, price, description (Text - first 10 words) ----------

Duplicates Found: 297

True Positives: 156
False Positives: 141
False Negatives: 1144

Precision: 0.525252525253
Recall: 0.12
F-measure: 0.195366311835

manufacturer, title, price, description (String - first 10 words) ----------

Duplicates Found: 130

True Positives: 54
False Positives: 76
False Negatives: 1246

Precision: 0.415384615385
Recall: 0.0415384615385
F-measure: 0.0755244755245

manufacturer, title, price (flipped), description (String - first 10 words) ----------

Duplicates Found: 116

True Positives: 49
False Positives: 67
False Negatives: 1251

Precision: 0.422413793103
Recall: 0.0376923076923
F-measure: 0.069209039

manufacturer_title, price, description (String - first 10 words) ----------

Duplicates Found: 84

True Positives: 70
False Positives: 14
False Negatives: 1230

Precision: 0.833333333333
Recall: 0.0538461538462
F-measure: 0.101156069364

manufacturer_title, price, description (Text - first 10 words) ----------

Duplicates Found: 11

True Positives: 11
False Positives: 0
False Negatives: 1289

Precision: 1.0
Recall: 0.00846153846154
F-measure: 0.0167810831426

manufacturer, title, price, description (Text - first 10 words), recall_weight=1.5 ------

Duplicates Found: 178

True Positives: 79
False Positives: 99

False Negatives: 1221

Precision: 0.443820224719
Recall: 0.0607692307692
F-measure: 0.106901217862

manufacturer, title, price, description (Text - first 10 words), recall_weight=2.5 ------

Duplicates Found: 21

True Positives: 2
False Positives: 19
False Negatives: 1298

Precision: 0.0952380952381
Recall: 0.00153846153846
F-measure: 0.00302800908403

manufacturer, title, price, description (Text - first 10 words), recall_weight=2.0 ------

Duplicates Found: 2576

True Positives: 331
False Positives: 2245
False Negatives: 969

Precision: 0.12849378882
Recall: 0.254615384615
F-measure: 0.170794633643