CS 839 Project Stage 3 Report

Zeping Ren, Stephen Sheen, Shiqi Yang

Data Entity

We extracted paper-based book selling information from two online book-selling websites:

- Barnes & Noble: `https://www.barnesandnoble.com`
- Better World Books: `https://www.betterworldbooks.com`

We collected book details information from both websites and obtained the two tables with the following schema:

```
A, B(title,author,isbn,publisher,edition,pages,dimension)
```

Table A contains 3645 tuples, and Table B 3632 tuples. We would like to match tuples that are referring to the same paper-based book.

Blockers

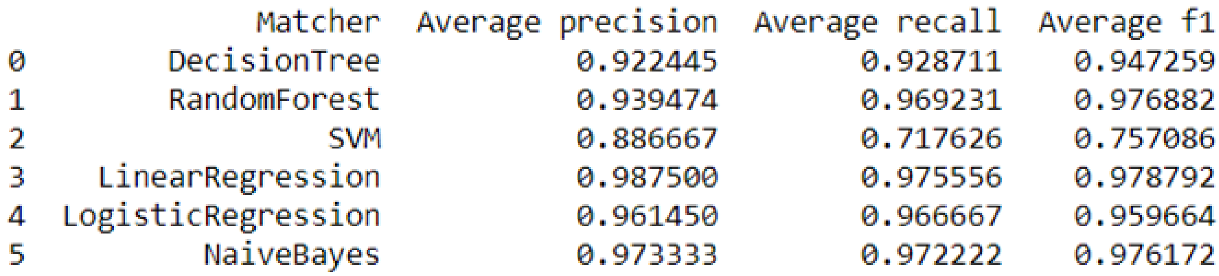
We created an overlap blocker and applied it twice to our data. The first time blocked on the attribute "title" with an overlap size of three, while the second time blocked on the attribute "author" with an overlap size of two. Stop words were removed during both times. The resulting candidate set obtained had a total of 1889 number of tuple pairs.

Sampling and Labeling

We sampled 300 tuples from the candidate set and labeled them manually to obtain the labeled set G.

Matchers

We split the labeled set G into the development set I and the testing set J, where set I contained 150 tuples and set J contained 150 tuples. We filled in "NaN" for values in set I and set J that were missing (Note: in the code, set H is set I and set L is set J, but both with missing values filled). We then performed a five-fold cross validation for the specified six learning methods (Decision Tree, Random Forest, SVM, Linear Regression, Logistic Regression, Naive Bayes) and obtained the following average precisiion, recall, and F-1:



```
              Matcher  Average precision  Average recall  Average f1
0        DecisionTree           0.922445        0.928711    0.947259
1        RandomForest           0.939474        0.969231    0.976882
2                 SVM           0.886667        0.717626    0.757086
3    LinearRegression           0.987500        0.975556    0.978792
4  LogisticRegression           0.961450        0.966667    0.959664
5          NaiveBayes           0.973333        0.972222    0.976172
```

Note, when we were performing the cross validation, we noticed that the attributes ”edition” and ”pages” contained a lot of missing values, which would not work well with these learning methods. As a result, we decided to remove both of these attributes by setting the attribute values of all tuples to ”NaN”.

After comparing the F-1 scores, the matcher using linear regression was selected as the best matcher. Since the precision, recall, and F-1 of the selected matcher were very high and stable (they were always above 0.90), we did not perform further debugging iterations. This final best matcher is linear regression had an average precision of 0.987500, recall of 0.975556, and F-1 of 0.978792.

Each of the specified six learning methods was used to train a matcher on set I, then tested on set J. The average precision, recall, and F-1 obtained from set J are shown below:

```
----------Best Matcher:  LinearRegression ----------
Precision : 98.68% (75/76)
Recall : 94.94% (75/79)
F1 : 96.77%
False positives : 1 (out of 76 positive predictions)
False negatives : 4 (out of 74 negative predictions)
---------- DecisionTree ----------
Precision : 88.64% (78/88)
Recall : 98.73% (78/79)
F1 : 93.41%
False positives : 10 (out of 88 positive predictions)
False negatives : 1 (out of 62 negative predictions)
---------- SVM ----------
Precision : 72.9% (78/107)
Recall : 98.73% (78/79)
F1 : 83.87%
False positives : 29 (out of 107 positive predictions)
False negatives : 1 (out of 43 negative predictions)
---------- RandomForest ----------
Precision : 93.98% (78/83)
Recall : 98.73% (78/79)
F1 : 96.3%
False positives : 5 (out of 83 positive predictions)
False negatives : 1 (out of 67 negative predictions)
---------- LogisticRegression ----------
Precision : 96.15% (75/78)
Recall : 94.94% (75/79)
F1 : 95.54%
False positives : 3 (out of 78 positive predictions)
False negatives : 4 (out of 72 negative predictions)
---------- NaiveBayes ----------
Precision : 93.75% (75/80)
Recall : 94.94% (75/79)
F1 : 94.34%
False positives : 5 (out of 80 positive predictions)
False negatives : 4 (out of 70 negative predictions)
```

The final best matcher Y using linear regression was trained on set I and tested on set J. The obtained accuracies were precision of 98.68%, recall of 94.94%, F-1 of 96.77%.

The approximate time estimate to do blocking was less than one second. This was most likely because the sizes of the original table A and B were not very large (about 3600 tuples each). As a result, the set obtained from the Cartesian product of the two tables, which blocking was applied, was of a manageable size. The approximate time estimate to manually label the data was about one hour. The approximate time estimate to find the best matcher was also less than one second. This was most likely because the size of the development set I was small (300 tuples), so cross validation did not take long.

Magellan comments

Good:

1. Automatic creation of features and pre-defined similarity scores were very helpful.

Bad:

1. It was hard to remove automatic-generated features that were unnecessary. There were also very few examples as to how features could be removed (most were above how features could be added).

2. The example of Entity Matching with Jupyter Notebook is using a Magellan version that is out of date. It would be helpful for future students/researchers if these are updated.