CS 784
Advanced Topics in Database Management Systems

Department of Computer Science
University of Wisconsin - Madison
Madison, WI 53706

Fall 2015 Final Project Report

Submitted by:

Qing Li
Fan Ding

Submitted to:

Prof. AnHai Doan

Date Due: November 23, 2015

1. General Performance of Each Matcher

***** All values of precision, recall and F-1 are the mean score after the first time cross validation.

Matcher Name	Precision	Recall	F-1
Decision Tree	57%	45%	49%
Random Forest	63%	47%	52%
SVM	0%	0%	0%
Naïve Bayes	73%	**51%**	**59%**
Logistic Regression	0%	0%	0%
Linear Regression	**82%**	30%	42%

2. Selected Matcher

Naïve Bayes

Since

i. It achieves highest recall and F1 score.

ii. It also achieves the second highest precision.

iii. It might be a good candidate to trade off precision and recall.

3. Debugging Iterations

Iteration 1, using TF/IDF on Feature attribute as feature input of matchers

Matcher Name	Precision	Recall	F-1
Decision Tree	57%	45%	49%
Random Forest	63%	47%	52%
SVM	0%	0%	0%

Naïve Bayes	73%	**51%**	**59%**
Logistic Regression	0%	0%	0%
Linear Regression	**82%**	30%	42%

Target Matcher: Naïve Bayes Matcher

Problem: Both precision and recall are not satisfied.

Possible solution: Add some extra features as input.

Iteration 2, using both TF/IDF and Jaccard on Feature attribute as feature inputs of matchers

Matcher Name	**Precision**	**Recall**	**F-1**
Decision Tree	52%	50%	51%
Random Forest	71%	58%	58%
SVM	0%	0%	0%
Naïve Bayes	70%	**69%**	**64%**
Logistic Regression	0%	0%	0%
Linear Regression	**75%**	36%	55%

Target Matcher: Naïve Bayes Matcher

Problem: Recall is satisfied, however precision still is low. Through DT debugger, we find out the attribute, Feature, covers too many details, so that for some unmatched tuples, the similarity is quite.

Possible Solution: Add some rules on another attribute, such as Name or Brand.

Iteration 3, (**Trigger/Rule**)

(i) Using both TF/IDF and Jaccard on Feature attribute as feature inputs of matchers

(ii) Add a Rule on Name since Brand has already been used to block.

(iii) Target Matcher: Naïve Bayes

(iv) Using TF/IDF on the attribute, Name

Trigger	**Precision**	**Recall**	**F1**
< 0.1	88%	65%	75%
< 0.2	**100%**	**57%**	**72%**
< 0.25	100%	48%	65%

Based on precision and recall, the second trigger value is selected.

Debugging Procedures:

During each iteration, since there were only two built-in debuggers that we can use. So in the debug step, we choose DT debugger to explore the details of false positives. The labels we made were correct, however, we find that some false positives were made due to the color difference. Eg. Although the similarity between two tuples is very high, however, their colors are different and we label them as 0.

Best Matcher

Naïve Bayes

Final precision/recall/f1

Trigger	Precision	Recall	F1
< 0.2	100%	57%	72%

4. Final Results

For all matchers,

Matcher Name	Precision	Recall	F-1
Decision Tree	56.25%	52.94%	54.55%
Random Forest	72.73%	47.06%	57.14%
SVM	0.00%	0.00%	0.00%
Na ïve Bayes	**73.33%**	**64.71%**	**68.75%**
Logistic Regression	**100.00%**	5.88%	11.11%
Linear Regression	87.50%	41.18%	56.00%

Final Best Learning Method (Na ïve Bayes),

Matcher Name	Precision	Recall	F-1
Na ïve Bayes	73.33%	64.71%	68.75%

Final Best Learning Method with Rule,

Matcher Name	Precision	Recall	F-1
Na ïve Bayes	100.00%	64.71%	78.57%

5. Other Details

a. How much did it take to label the data?

Nearly three hours.

b. How much did it take to find the best matcher?

Nearly four hours.

c. How much did it take to add rules?

Nearly one hours.