Stage III Report

Chang Guo (cguo42@wisc.edu)
Yuncong Hao(hyuncong@wisc.edu)
Qun Zou(qzou5@wisc.edu)

- **Describe the type of entity you want to match, briefly describe the two tables (e.g., where did you obtain these tables), list the number of tuples per table.**

  We want to match LA's restaurants from yelp website with LA's restaurants from TripAdvisor website.
  They both have attributes as shown below:

restaurant name	postal code	phone number	star	price	number of reviews

- **Describe the blocker that you use and list the number of tuple pairs in the candidate set obtained after the blocking step.**

  We used phone number as the attribute to do the attribute equivalent blocking with allowance of the missing value on phone number. And then we used overlap blocking on the name of the restaurants with the overlap_size to be one in word level.

- **List the number of tuple pairs in the sample G that you have labeled.**

  We labeled 250 tuple pairs.

- **For each of the six learning methods provided in Magellan (Decision Tree, Random Forest, SVM, Naive Bayes, Logistic Regression, Linear Regression), report the precision, recall, and F-1 that you obtain when you perform cross validation for the first time for these methods on I.**

	Precision	Recall	F1
SVM	0.946151677	0.954675324	0.94765226
Decision Tree	0.927923655	0.914442224	0.91774784
Random Forest	0.924723105	0.963766233	0.94199607
Naive Bayes	0.937983179	0.941341991	0.93683124
Logistic Regression	0.904877720	0.936897546	0.91731691
Linear Regression	0.800000310	0.81371892	0.80192567

- **Report which learning based matcher you selected after that cross validation.**
  We selected Support Vector Machine as our matcher.

- **Report all debugging iterations and cross validation iterations that you performed. For each debugging iteration, report (a) what is the matcher that you are trying to debug, and its precision/recall/F-1, (b) what kind of problems you found, and what you did to fix them, (c) the final precision/recall/F-1 that you reached. For each cross validation iteration, report (a) what matchers were you trying to evaluate using the cross validation, and (b) precision/recall/F-1 of those.**
  a) We tried to debug SVM.
  b) For iteration 2, we found there was still dirty data remained in our golden data table, (for example, the uppercase and lowercase characters were taken as different input; same restaurant name with special character " ' ", like Denny's from the two websites were considered to be different) so that we redid the data cleaning and blocking step to regenerate a new golden data table with 400 pairs.
  c) For Iteration 3, we found there's no other improvements we can do to increase the accuracy. We've found the the result which have different prediction value with real value are the pairs with similar restaurant names (can be detected by human beings) but also missing values in more than one other columns(including postcode, address, phone number).
  Iteration 2:

	Precision	Recall	F1
SVM	0.97022435897435	0.96999999999999997	0.9700150375
Decision Tree	0.93025641025641026	0.92000000000000004	0.93003508771929833
Random Forest	0.97999999999999998	0.94999999999999996	0.93975757575757568
Naive Bayes	0.97022435897435899	0.96999999999999997	0.97001503759398489
Logistic Regression	0.97999999999999998	0.94999999999999996	0.94004801920768299
Linear Regression	0.827826538	0.83960620181470158	0.826665490

Iteration 3:

	Precision	Recall	F1
SVM	0.988333333	0.9825	0.98532258
Decision Tree	0.965256410	0.9753571429	0.96928554
Random Forest	0.980377867	0.9775	0.97863027
Naive Bayes	0.993333333	0.9825	0.98775847
Logistic Regression	0.965256410	0.9825	0.97298924
Linear Regression	0.830157432	0.8410297823	0.83572819

- **Report the final best matcher that you selected, and its precision/recall/F-1.**
  We decided to use Naives Bayes.
- **For each of the six learning methods, train the matcher based on that method on I, then report its precision/recall/F-1 on J.**

	Precision	Recall	F1
SVM	0.988333333	0.9825	0.98532258
Decision Tree	0.965256410	0.9753571429	0.96928554
Random Forest	0.980377867	0.9775	0.97863027

Naive Bayes	0.993333333	0.9825	0.98775847
Logistic Regression	0.965256410	0.9825	0.97298924
Linear Regression	0.827826538	0.83960620181470158	0.826665490

- **For the final best matcher Y selected, train it on I, then report its precision/recall/F-1 on J.**

	Precision	Recall	F1
Naive Bayes	0.993333333	0.9825	0.98775847

- **Report approximate time estimates: (a) to do the blocking, (b) to label the data, (c) to find the best matcher.**

  Block: 1.5hr
  Label : 3hr
  Find : 3.5hr

- **Provide a discussion on why you didn't reach higher recall, and what you can do in the future to obtain higher recall.**

  Firstly, our result of precision, recall and F1 values are high enough to meet the matcher requirement. Also, after our iteration endeavor, we have not observed the values increase anymore. We concluded that there could still be some tuples have similar names but with missing values on the entities information (including phone number, postcode, address). For these tuples, they can be detected by human beings however it's really hard to set specific rules for our matcher regarding on these issues. So we concluded that we cannot reach higher recall. To obtain higher recall, we need more information to fill in entities with missing value and dirty value information.

- **BONUS POINTS: provide comments on what is good with Magellan and what is bad, that is, as users, what else would you like to see in Magellan. Are there any features/capabilities that you would really like to see being added? Any bugs? Depending on how detailed and helpful these comments are, you can get bonus point from 1-10 (which will help with the final grade, not just with the project).**

  Good: The user guide and documentation is clear and easy to follow.
  Features to add:

When we are doing the debugging, we'd love to extract the last prediction column as the result to see the difference with prediction value and true value. We'd love to have a function that allow us to extract the last column directly and also tell us which value we predict wrong.