Stage – 3 (Entity Matching)

Manjunath Nagaraj Shettar (shettar@wisc.edu)
Samhith Venkatesh(svenkatesh5@wisc.edu)
Jayashankar Tekkedatha(tekkedatha@wisc.edu)

1. Entity Type:

The entity types we are trying to match: **Movie** across two tables.
Table 1 - Top 3050 movies of all time from IMDB.
Table 2 – Top 3000 movies of all time from Metacritic website.
Common attributes in both the tables.

1. Name
2. Release_year
3. Runtime
4. Director
5. Actors

2.Blocking

We used three types of blockers.

2.1 Overlap Blocker

We applied Overlap Blocker on the director names followed by actor names. On initial analysis we understood that word level blocking is more relevant than the q-gram blocking. Hence, we retained the tuples which had at least one overlapping director and one overlapping actor.
Number of tuples after OverLapBlocker on director names – **80675** (remaining tuples)
Number of tuples after OverLapBlocker on director names – **36029** (remaining tuples)

2.2 Attribute Blocker

Matching the movies based on attribute equivalence between movie names was considered initially. However, to make the process of blocking and matching **more challenging** this blocker was dropped.

2.3 BlackBox Blocker

BlackBoxBlocker is a great way to write custom functions. We noticed that for some tuples, release year and runtime were different. We applied the following two simple heuristics.

2.3.1 Block all the tuples whose release year differs by a factor of 2. - **7751** (remaining tuples)

2.3.2 Block all the tuples whore runtime differs by a factor of 10 minutes. – **3212** (remaining tuples)

2.3.3 We also noticed that some records in Metacritic website did not have director and actor names. We again wrote a blackbox function to retain all the tuples with overalpping words in the movie name but missing actor/director names. This significantly reduced the number of tuples and at the same time helped us retain movies with missing attributes. **862** (remaining tuples)

3. Sampling

The number of tuple pairs in the sample G that were labeled – 500.

4. Matching

The following is the precision, recall, and F-1 that we obtained on the I dataset for the first time.

	Matcher	Average Precision	Average recall	Average f1
1	**Decision Tree**	0.984981	0.979737	0.982201
2	**RF**	0.985366	0.985122	0.985183
3	**SVM**	0.848651	1.000000	0.917650
4	**LinReg**	0.980103	0.984859	0.982451
5	**LogReg**	0.985482	0.985000	0.985055
6	**NB**	0.985000	0.970366	0.977628

We decided to choose the **RandomForest** matcher as it provided the highest precision and recall.

5. **Matching Debugging**

Iteration 1: Matcher – RandomForest

Problems:

After close analysis on the feature set generated by Magellan, we noticed that the feature set generated on numerical attributes were not contributing towards increasing accuracy values i.e even after removing the numerical features the accuracy values were approximately same.

Custom blackbox_feature: We wrote a new blackbox function which calculated the absolute difference between the release year attribute of the tuples.

Matcher	RandomForest
Precision	0.985366
Recall	0.985122
F1	0.985183

6. **Final Best Matcher on I**

After performing CV for one more time we noticed Random Forest still gave us a higher precision and recall.

Matcher	Random Forest
Precision	0.989872
Recall	0.985122
F1	0.987464

7. Accuracy of all the Matchers trained on Set I and tested on Set J.

Matcher	**Precision**	**Recall**	**F1**
Random Forest	100%	99.48%	99.74%
DecisionTree	99.48%	99.48%	99.48%
SVM	83.91%	100%	91.25%
NB	99.48%	99.48%	99.48%
Linear Regression	100%	99.48%	99.74%
Logistic Regression	99.48%	98.96%	99.22%

8. Final Matcher on Test Set J

Matcher	**Random Forest**
Precision	**100%**
Recall	**99.48%**
F1	**99.74%**

9. Time estimates

Blocking	**1 day (with many breaks)**
To label the data	**10mts (2 people worked parallelly)**
To find the best matcher	**Half a day (with many breaks)**

10. Feedback

10.1 Overlap Blocker:

We noticed that many of the blockers use their own custom functions internally to do the blocking. The blockers allow us to choose the word level or qgram level blocking and minimum number of overlaps to be present. However, what option we wanted was : choosing the number of overlaps based on the total number of combinations generated. We have shown this in the example below.

Actor 1: Tom hanks, Jerry hayworth.
Actor 2: Tom hanks, Jerry Rogers.

We notice that there are two-word level overlaps. We want something like if number of overlaps are greater than 1/3rd of the combinations then don't block this tuple. Rather than just specifying one, two, three etc. overlaps, this would help us in setting the threshold deterministically based on the combinations generated.

10.2 Black Box Blocker:

We decided to write our own blocker to address the above issue. However, it took several hours to run the custom blocker. We understand that Magellan uses highly optimized source code to do the blocking through default functions. We would like to see if Magellan can help users write their own custom blockers and still get the results in short time as other built in blockers.

10.3 Automatic Features for Numerical Attributes:

As we discussed earlier in the report, we noticed that Magellan was not generating good features for the numerical attributes i.e even after removing the default numerical features generated by Magellan we did not see any change in accuracy.