CS 839 - Spring 839 - Project Stage 3

Team - Abhinav Garg, Bidyut Hota, Chirayu Garg

Data Description

Extracted data from video game review websites is used. We selected the following sources to retrieve structured data:
Source A - Gamespot (https://www.gamespot.com/)
Source B - How Long To Beat (https://howlongtobeat.com/)
Both the data sources are video game review websites that contain general information such as title, release data, supported platforms, game developers, user ratings, etc.

We have matched tuples in between the two tables by **title** attribute.
Table A contains tuples extracted from Source A (Gamespot). - **8884 tuples**
Table B contains tuples extracted from Source B (How Long To Beat) - **8481 tuples**

Blocking Process

A multiple-stage blocking pipeline is used. Blocking stages are as follows:

1. Overlap Blocker
    a. Initially tables are compared on 'Title' attribute using a q-gram (size=3) and overlap of 4 tokens
    b. Generated candidate set is then compared again using a 1-word overlap on 'Title'
    c. The candidate tuples are then compared using a rule based blocker. Similarity measure of 'Title' attribute is used and tuples with score less than 0.4 are blocked
    d. A black-box matcher is then used to compare release dates of the game titles and if release date is more than 1 year apart then the candidate tuple is blocked

2. Attribute Equivalence blocker - On the input tables A and B attribute equivalence is used on 'Title' attribute to determine exact matches. This stages captures some of the candidate tuples dropped in word and q-gram based overlap blocker

A union of candidate tuples from step 1 and 2 listed above is used to generate total candidate tuples. Following table summarizes the count of candidate tuples at each stage of blocking process:

Blocker Process	**Sub-stage**	**Candidate Tuples**
Initial Candidates		75345204
Overlap Blocker		
	Q-gram overlap	979045
	Word overlap	530183

Rules Based Blocker		70668
Black Box Blocker (X)		13062
Attribute Equiv. Blocker (Y)		3480
Total Tuples (X U Y)	Combining candidates	**13186**

Sample G

Total **400 tuples** have been labeled which are available in file '*labeled.csv*'.

Sample G Distribution	**Size**
Positive	120
Negative	280

Set	**Size**
Development Set - I	285
Evaluation Set - J	125

5-fold Cross-validation for Classifier

Learning Method	**Precision %**	**Recall %**	**f1-Score %**
Decision Tree	96.5	96.93	96.63
Random Forest	100.0	96.25	97.93
Support Vector Machine	0.0	0.0	0.0
Linear Regression	100.0	93.67	96.45
Logistic Regression	93.15	94.43	93.68
Naive Bayes	92.43	96.93	94.43

Debugging

As random forest classifier achieved 100% precision for cross-validation on Set-I, we didn't do any further debugging on the matcher. However, as a pre-processing step before starting the blocking process we did clean the release date of tuples in table A and B. Cleaning process includes unifying the date format and setting date and months to some intelligent values where Quarter of the year was mentioned instead of exact date or partial information such as only

month or year was available. The black-box blocker utilizes these cleaned dates to do blocking which greatly improve the density of matching tuples in blocked candidate set.

Best Matcher

Random forest classifier give best cross-validation values for precision at first trial.

Test Set J Prediction

Learning Method	Precision %	Recall %	f1-Score %
Random Forest	96.43	84.38	90.0
Decision Tree	90.32	87.5	88.89
Logistic Regression	90.91	93.75	92.31
Linear Regression	93.1	84.38	88.52
Naive Bayes	88.57	96.88	92.54
Support Vector Machine	0.0	0.0	0.0

Entity Matching Pipeline Time Estimate

Entity Matching Stage	Approx. Time Taken (Hour)
Blocking Process	7
Labeling Process	2
Matcher Determination	2

Recall Discussion

In the dataset some of the entity tuples are very alike for a classifier to be able to differentiate from a match to a mismatch, for example there are titles tuple pair such as:
i) (Rogue Squadron ,Rogue Squadron 3D)
ii) (Call of Duty 1, Call of Duty 2)

The first tuple pair is a match while the second title tuple pair is not, it is highly unlikely that a learning algorithm can figure out the nuances to distinguish between two. Presently the first tuple pair is classified as a mismatch and hence recall is low. However, we do achieve a recall of 84.38 % which is not very low.

Magellan Comments

Overall of the library is very good and the documentation is detailed enough to walk a new user through the entity matching process. One possible feature that needs to be incorporated to the blocking algorithms is to have a flag that converts all string attributes to lowercase before

running the blocking algorithm. In our dataset such a feature would have greatly helped reduce potential candidate tuples lost in the blocking process.

Anaconda libraries for Magellan v0.3 for python 3.6 on windows seems to be missing the core part of execution code and as a result result in error on import of ‘py_entitymatching’ module. This was verified by downloading the tar package separately from Anaconda package website. As a result we ended up using Magellan v0.2