CS 839 – Project

Stage 3: Entity Matching

Group Members: Pradeep Kumar, Pulkit Kapoor, Sonu Agarwal

Description of the two Web data sources

In this project stage, our team performed matching between two tables containing **Movie data** as entities.

Source1: http://www.imdb.com/ - The table from Source1 contains **3013 tuples**. This table includes the following columns –

```
MID,Title,Certificate,Genre,Rating,Running Time,Directors,Stars
Cast,Country,Release Date,Production Company,Release Year,Release Month.
```

Source2: https://www.allmovie.com/ - The table from Source2 contains **3300 tuples**. This table includes the following columns:

```
MID,Title,Certificate,Genre,Rating,Running Time,Directors,Stars
Cast,Country,Release Date,Production Company,Release Year,Release Month.
```

We converged on these sources after manually browsing through several websites which provide movie information. We looked for a pair of websites which provided a large number of common attributes and all these attributes were present in some fixed format. We then decided the set of attributes which we wanted to extract from the two sources.

Blocking

Our blocking consists of the following two rules:

1. We first applied blocking rule based on the movie name. We used Jaccard measure with 3-gram tokens between the movie names across two table, with the very low threshold of 0.3. This reduced the number of tuples to 5210.
2. We then applied the second rule on the tuple set provided by the 1st rule. We used overlapping measure of 2 with 2-gram tokenization on director's name. This reduced the number of tuples to 2028.

   As a result, we reduced the final size of our candidate set from **9942900** (=3013 x 3300) to **2028.**

It took us approximately 10 hours to find the best combination of blocking rules along with debugging to get an adequate set of candidate tuples without losing positive examples.

Sampling and Labelling

We sampled randomly **500 tuple pairs** (Sample G) from the set of 2028 potential candidates and labelled them.

Number of positive labels: 150
Number of negative labels: 350

We spent about 2 hours on labelling data.

Training and Selecting the best classifiers

We used 6 learning methods for training on **set I** using 5-fold cross validation. The methods include: (1) Decision Tree, (2) Random Forest, (3) SVM, (4) Naive Bayes, (5) Logistic Regression, and (6) Linear Regression. Our classifiers use 27 features, and below is the **first-attempt accuracy performance of our classifiers on the training set I**:

Machine Learning Algorithm	Precision	Recall	F-1
Decision Tree	0.98	0.96	0.97
Random Forest	0.99	0.96	0.97
SVM	0.96	0.87	0.91
Naïve Bayes	0.98	0.96	0.97
Logistic Regression	0.96	0.96	0.96
Linear Regression	0.95	0.97	0.96

We selected **Random Forest** learning based matcher after cross validation.

The **final best matcher (Y)** selected is the following –

Machine Learning Algorithm	Precision	Recall	F-1
Random Forest	.99	.96	.97

Testing on Set J

Machine Learning Algorithm	Precision	Recall	F-1
Decision Tree	.91	.95	.93
Random Forest	.98	.95	.96
SVM	.91	.89	.90
Naïve Bayes	.89	.95	.92
Logistic Regression	.92	1.00	.96
Linear Regression	.92	1.00	.96

After selecting the final best matcher Y and training it on I, the final precision/recall/F-1 on set J is the following –

Machine Learning Algorithm	Precision	Recall	F-1
Random Forest	.98	.95	.96

We devoted around 4 hours to find the best matcher. Most of this time went to find the best set of features (based on different set of attributes) which could give us the best matching accuracy.

Feedback on Magellan:

- In matcher part, there are support of multiple machine learning models. Most of these models have some hyper-parameters associated with and in many practical problems, tuning these parameters plays a big role in getting the desired accuracy. So, we feel that there should be extensive support for changing all the related parameters and corresponding help documents.
- Getting the following warning while executing "*em.select_matcher*":

```
"/Users/sonuagarwal/anaconda2/lib/python2.7/site-packages/scipy/linalg/bas
ic.py:1226: RuntimeWarning: internal gelsd driver lwork query error, requi
red iwork dimension not returned. This is likely the result of LAPACK bug
0038, fixed in LAPACK 3.2.2 (released July 21, 2010). Falling back to 'gel
ss' driver.
  warnings.warn(mesg, RuntimeWarning)"
```