Web Data Integration

Student project report

presented by
Oliver Frendo (1510432),
Dandan Li (),
Zehui Wang () *and*
Yi-Ru Cheng ()

submitted to the
Data and Web Science Group
Prof. Dr. Christian Bizer
University of Mannheim

November 27, 2015

Contents

List of Figures

List of Tables

1 Introduction to data and use case

2 Data translation

2.1 Data collection

In order to collect suitable data we tried different data service providers such as Datahub, finally we have collected total four datasets from three different sources and in three different formats, including:

2.1.1 Forbes: Company

The Forbes offers a .xls file with a list of Top 2000 companies during the period 2000 to 2014 which were published in Forbes magazine because of great performance in terms of business achievements. This dataset describes the basic information about these top 2000 companies. For example, location shows where this company is founded, industry depicts what fields the company focus on and so on.

2.1.2 DBpedia: Company

The information of company is extracted from DBpedia, since it provides relatively complete information. To access information from DBPedia we used the public SPARQL endpoint (at http://dbpedia.org/sparql). Figure 1.1 is our query for company, actually there is total 764398 companies in DBpedia, which would be too much for us and also not easy to handle it in terms of processing time and space. In order to reduce the number of data, we limit the company types to ”company” and ”public company” and only extract the companies that provide attributes ”LocationCity” and ”LocationCountry”, these two attributes can also be related with Location Information, that’s why we consider them as necessary and others are optional. On the other hand, if all these attributes are necessary, there will be only few thousands companies extracted, because not all companies have all these nine attributes, in this case few overlapping data will be in the final integration results. In addition to this, as many attributes such as KeyPeople, locationCity have multiple values which result in the same company would appear more than one times, to avoid these duplicates we used ”group_contac”, a function in Sparql, to group many value together. There are also many values for Revenue but without date notation, so we just took the maximum value.

2.1.3 Freebase: Company

Freebase, like DBPedia, offers a web service which can be queried for data. Instead of SPARQL, however, the service is used by sending JSON requests. Like DBPedia this lets us select certain companies, or companies with certain attributes.

2.1.4 DBpedia: Location

We also extracted Location information from DBpedia with the same method as Company. Figure 1.2 is the query for location. For the same reason as Company, we limit the location types to ”city” and ”AdministrativeRegion”, which are more relevant to our company dataset. Also some attributes have many values without extra information, it’s hard to identify which one represents the current state, thus, we just took the maximum number of them among multiple values. Furthermore, the name of locations are provide in different languages, while in our project we just focus on english, so we filtered language as english.

Table 1: Basic Profile of Each Dataset

	Source	Format	Class	#Entities	#Attributes
	List of Attributes				
Forbes Global	forbes.com	xlsx	company	2000	7
	Company,country,industry,revenue,assets,marketvalue,profit				
DBpedia	dbpedia.org/sparql	csv	company	16051	9
	Company,country,industry,revenue,numberOfemployees, foundingYear,assets,keyPeople,location				
Freebase	freebase.com/query	json	company		9
	Company,country,industry,revenue,numberOfemployees, foundingYear,profit,keyPeople,location				
Dbpeida	dbpedia.org/sparql	csv	location	3270	5
	Location,country,population,area,elevation				

2.2 Integrated schema

We looked into four datasets and did the following Integrated Schema.In this table we use prefix dataset 1, 2, 3,4 respectively represent Forbes, DBpedia(company), Freebase and DBpedia(Location)

Table 2: Integrated Schema

Class Name	Attributes Name	Datasets in which attribute is found	Conflict resolution strategies
company	(company)name	dataset 1, 2, 3	LongestString, FavourSources, Voting
company,location	country	dataset 1, 2, 3, 4	Voting
company	industries	dataset 1, 2, 3	Union
company	Sales/ revenue	dataset 1, 2, 3	FavourSources
company	number Of Employees	dataset 2, 3	Average, Growth: Max
company	founding year/ date founded	dataset 2, 3	MostComplete (complete date) AND, MostComplete (sample)
company	Assets	dataset 1, 2	FavourSources
company	Market Value	dataset 1	/
company	profit	dataset 1, 3	FavourSources
company	continent	dataset 1	/
company	keyPeople/ leadership	dataset 2, 3	Union
company, location	name	dataset 2, 3, 4	Union AND FavourSources
location	population total	dataset 4	/
location	area total	dataset 4	/
location	elevation	dataset 4	/

2.3 Data transformations

Transformations were applied at two different points in this phase. The first was applied during mapping while the second was applied in the Java project. To begin with, numeric attributes with big values such as revenue or assets were often retrieved in scientific notation. Accordingly a function within MapForce was used to convert the numbers into a decimal notation. Secondly the datasets did not possess an ID attribute. Because it was going to be used later, it was generated with ”GenerateID” in MapForce.

The next transformations occured in Java. Many values, especially from the two DBpedia datasets, were loaded in the form of a URL due to our SPARQL query. As such the URL was parsed and only the actual value was kept. In addition, punctation and symbols such as ”_” were removed. Lastly we normalized country values, which was an important step for the blocking functions used later on in identity resolution. Especially values for the USA were transformed from spellings such as ”US”, ”USA”, ”United States” to ”United States of America”.

3 Identity resolution

3.1 Gold standards

3.1.1 Forbes ↔ Freebase

3.1.2 Freebase ↔ DBpedia companies

3.1.3 DBpedia companies ↔ DBpedia locations

3.2 Matching rules

This section explains the matching rules we tried in order to generate correspondences accurately. We matched the following datasets with each other:

- Forbes vs Freebase
- Freebase vs DBpedia
- DBpedia companies vs DBpedia locations

Attribute	**MatchingRule**	**P**	**R**	**F1**
Forbes vs Freebase				
name	Equals	1,0000	0,7500	0,8571
	Levenshtein	0,8571	1,0000	0,9231
countries	Equals	0,8571	1,0000	0,9231
	Jaccard	0,8571	1,0000	0,9231
	Highest Jaccard	0,8571	1,0000	0,9231
industries	Jaccard	0,9091	0,8333	0,8696
	Combination of Jaccard and Levenshtein	0,8571	1,0000	0,9231
revenue/ profit	PercentageSimilarity (max_percentage=0.5)	0,8571	1,0000	0,9231
Freebase vs DBpedia				
revenue/ numberOfEmployees	PercentageSimilarity (max_percentage=0.5)	0,9167	0,9167	0,9167
dateFounded	YearSimilarity (maxDifference=20)	0,9167	0,9167	0,9167
keyPeople	Jaccard	0,9167	0,9167	0,9167
	Combination of Jaccard and Levenshtein	0,9167	0,9167	0,9167
locations	Jaccard	0,9167	0,9167	0,9167
	Highest Jaccard	0,9167	0,9167	0,9167
DBpedia companies vs DBpedia locations				
countries	Highest Jaccard	0,9706	0,9429	0,9565
locations	Jaccard	0,9630	0,7429	0,8387
	Highest Jaccard	0,9706	0,9429	0,9429

Table 3: Matching rule accuracies

In particular the rules for `name`, `industries` and `locations` show different results. For `name` we chose to use Levenshtein because of misspellings, or because of the company type (e.g. ”Inc.” or ”PLC”). However, this also introduces some problematic cases such as ”West Japan Railway” and ”East Japan Railway”, which are different companies but possess very similar attribute values and also generate a very high Levenshtein similarity. For `industries` we tried Jaccard first. This however is not an accurate measure of similarity because of slight differences like ”Transport” and ”Transportation”. As such we chose to use a combination of Jaccard and Levenshtein which led to better results:

$$sim_{Jaccard+Levenshtein} = \frac{\sum_{x,y} max(sim_{Levenshtein}(x,y))}{|x| + |y| - \sum_{x,y} max(sim_{Levenshtein}(x,y))}$$

To give an example of two companies with two industries each: ”Computer, Transportation” and ”Computers, Transport” would generate a similarity of 0 with Jaccard but 0.75 with our approach. We used the same approach for comparing `keyPeople`, where misspellings of names are more important. `locations` and `countries` were compared using ***Highest Jaccard***: This means we compared each location of an entity with each location of another entity using Jaccard and then picked the highest value. To give an example: Comparing a company with two locations ”New York” and ”London” with another company with only one location ”New York City” would give bad results using *Equals* or *Levenshtein*, which is why we chose to use the highest Jaccard value. Very often there were entities with multiple countries or locations but only single intersections. Due to the sparsity and potential unreliability of Freebase and DBpedia we wanted the similarity to reflect this. Lastly we compared numeric attributes such as `Revenue` using the PercentageSimilarity: However numeric data from Freebase and DBpedia is too sparse, unreliable or outdated. Learning a matching rule via linear regression confirms this by assigning weights of 0 to both these attributes.

3.3 Blocking functions

Table 4 shows the blocking functions we tried and used in our project. For the comparison of the Forbes and Freebase datasets a partitioning by `countries` shows good results, which is consistent with the high density of the attribute in both datasets. We also tried a sorted neighbourhood approach on the same attribute which seemed to be less effective. Using a cross product approach for comparing Freebase with DBpedia was impossible due to the large size of DBpedia. As such we tried partitioning by `countries`, `dateFounded` (where the blocking key is $year/20$) and a combination of the two. The combination reflects our own implementation of a partitioning blocker, where we generate a match to be evaluated if the one of the two blocking keys are the same. This shows the best results because both attributes are relatively, but not completely, dense in both datasets, which is why the reduction ratio is lower then when using only one of the two. When comparing companies with locations from the DBpedia datasets `countries` is the only possible blocking key.

Dataset Comparison	**Blocking function**	**Time**	**Match**	**Ratio**	**P**	**R**	**F1**
Forbes vs Freebase	CrossProduct	00:32	509	1,00	0,86	1,00	0,92
	SortedNeigh. (Country)	00:05	319	6,80	0,87	0,58	0,70
	Partitioning (Country)	00:02	425	20,19	0,86	1,00	0,92
Freebase vs DBpedia	Partitioning (Country)	00:44	576	15,92	0,90	0,75	0,82
	Partitioning (DateFounded)	00:39	496	9,43	0,89	0,67	0,76
	Partitioning (Combination)	01:22	671	6,13	0,92	0,92	0,92
Companies vs Locations	Partitioning (Country)	00:41	7.921	4,11	0,97	0,94	0,96

Table 4: Blocking functions

3.4 Learning matching rules

We were able to improve the results of our identity resolution by learning the weights for a linear matching rule from a linear regression in RapidMiner over our handwritten rules. To give an example, the learned weights for the datasets from Freebase and DBpedia are as follows. Interestingly, both `keyPeople` and `locations` seem to be important, while the weights for both numeric attributes revenue and `numberOfEmployees` is assigned a weight of 0, indicating the attributes are not very useful for an accurate comparison. Lastly the `name` attribute has the highest weight, as expected.

name	0.689	revenue	0.000
countries	0.088	numberOfEmployees	0.000
industries	0.025	keyPeople	0.377
dateFounded	0.170	locations	0.218
intercept	−0.135		

4 Data fusion

4.1 Input data

4.2 Gold standard

4.3 Conflict resolution functions

4.4 Accuracy

References

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Master-/Bachelorarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Er- klärung rechtliche Folgen haben wird.

Mannheim, November 27, 2015 Unterschrift