Web Data Integration

Student project report (Group 5)

presented by
Oliver Frendo (1510432),
Dandan Li (1486051),
Zehui Wang (1482582) *and*
Yi-Ru Cheng (1526763)

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

December 6, 2015

Contents

List of Figures

List of Tables

1 Data translation

1.1 Use case

Our goal in this project is to integrate information about companies with information about cities in which their headquarters are located. The resulting dataset could then be analyzed from a data science point of view in order to find relationships, i.e. how does the population in a city correlate with the size or other attributes of companies. In order to gather more information about companies, we first combine several datasets together, all of which are about companies but derived from different sources. We then integrate this result with the data about locations.

1.2 Datasets

In order to collect suitable datasets we tried several data service providers such as Datahub, the German Statistisches Bundesamt or different stock exchanges. In the end we decided to collect four datasets from three different sources.

1.2.1 Forbes: Companies

Forbes is an American business magazine and is well known for its lists and rankings, including its lists of the richest Americans (the Forbes 400) and rankings of world's top companies (the Forbes Global 2000). The ranking is based on a mix of four metrics: sales, profits, assets and market value. This dataset contains information about 2000 companies for the period of 2000 to 2014 in the form of 7 attributes.

1.2.2 DBpedia: Companies

To access information in DBPedia we used the public SPARQL endpoint[1]. Figure 1 contains an excerpt of our query for companies. A problem we ran into early is the sheer number of companies (764,398) in DBpedia. In order to reduce this number we limit the resource types to `Company` and `Public_company` and only extract the companies that provide values for the attributes `locationCity` and `locationCountry`. We later use these attributes for blocking functions and to create correspondences, which is why we consider them to be neccessary, while others are optional. If we defined all attributes to be non-optional this dataset would contain only a few thousand companies, as not all companies have all values for all nine attributes. In addition to this some attributes such as `KeyPeople` have

[1] http://dbpedia.org/sparql

multiple values. As a result the company would appear more than once. In order to avoid duplicates we used the SPARQL function `group_concat` to group values together. In addition, there were other attributes such as `Revenue` with multiple values. Lacking any provenance information such as a date we decided to use the maximum value.

```
SELECT ?company
group_concat(distinct str(?keyPeople);separator=";;") as ?keyPeople ....
WHERE { {?company rdf:type dbo:Company}
UNION {?company dbr:type dbr:Public_company} .
optional{?company dbp:keyPeople ?keyPeople .}
?company dbp:locationCity ?locationCity.....}ORDER BY ?company
```

Figure 1: DBpedia query for companies

1.2.3 Freebase: Companies

Freebase provides a web service to query entities and returns them in a JSON format. The total number of entities returned by our query is 3,182. Similar to the SPARQL query above we decided to make some attributes optional while others are non-optional. Specifically, the `number_of_employees` is non-optional because we use it to compare companies in each dataset and to reduce the number of companies down from 230,000. To make sure we receive the right attributes we first tested queries on the query page. To actually retrieve the data we used Java to make calls to the MQL Read API[2]. To avoid issues during the mapping procedure in MapForce we concatenated attributes with multiple values with a delimiter.

```
[{
  "name": null,
  "name!=": "null",
  "/organization/organization/date_founded": null,
  "/business/business_operation/industry": [],
  "/business/employer/number_of_employees": [{
    "number": null
```

Figure 2: Freebase query for companies

[2]https://developers.google.com/freebase/v1/mqlread?hl=en

1.2.4 DBpedia: Locations

We extracted information about locations from DBpedia using the same method as for companies. Similarly, we limited the resource type in our SPARQL query to `City` and `AdministrativeRegion`, which are more relevant to our company dataset. We also found the same problem of many values for attributes without any extra provenance information, whch makes identifying the current state hard. As such we used the maximum number for `population`, for example. Lastly, the names of the locations are often provided in different languages in DBpedia. Because we focused on English in our project we chose the value for `name` by filtering the language labels accordingly.

1.3 Data collection

The results of our data collection are shown in table 1, which describes the basic profiles of each dataset.

	Source	Format	Class	#Entities	#Attributes
	List of attributes				
Forbes	forbes.com	xlsx	company	2000	7
	name, countries, industries, revenue, assets, marketValue, profit				
DBpedia	dbpedia.org/sparql	csv	company	16051	9
	name, countries, industries, revenue, numberOfEmployees, dateFounded, profit, keyPeople, locations				
Freebase	freebase.com/query	json	company	3182	9
	name, countries, industries, revenue, numberOfEmployees, dateFounded, profit, keyPeople, locations				
DBpedia	dbpedia.org/sparql	csv	location	3270	5
	locationName, country, population, area, elevation				

Table 1: Basic profile of each dataset

1.4 Integrated schema

We analyzed the four datasets and created the following integrated schema. In the table we use the prefixes dataset 1, 2, 3 and 4 to represent Forbes, DBpedia (companies), Freebase and DBpedia (locations) respectively.

Class Name	Attributes Name	Datasets in which the attribute is found
company	(company) name	dataset 1, 2, 3
company, location	country	dataset 1, 2, 3, 4
company	industries	dataset 1, 2, 3
company	revenue	dataset 1, 2, 3
company	numberOfEmployees	dataset 2, 3
company	dateFounded	dataset 2, 3
company	assets	dataset 1, 2
company	marketValue	dataset 1
company	profit	dataset 1, 3
company	continent	dataset 1
company	keyPeople	dataset 2, 3
company, location	(location) name	dataset 2, 3, 4
location	population	dataset 4
location	area	dataset 4
location	elevation	dataset 4

Table 2: Integrated schema. Datasets 1, 2, 3 and 4 refer to Forbes, DBpedia (companies), Freebase and DBpedia (locations)

1.5 Data transformations

Transformations were applied at two different points in this phase. The first was applied during mapping while the second was applied within Java. To begin with, numeric attributes with large 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.

Java was used to apply the next transformations. 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 a uniform ”United States of America”.

2 Identity resolution

2.1 Gold standards

As described above we have four datasets in total. We created three gold standards divided into a training set and a test set. Each test set's size is 10% of the training set's size mostly including corner cases. The method of finding the same entities is to compare an entity from a dataset with a smaller size to one with a greater size and all gold standards are selected by stratified distribution in ascending order. The corner cases in our project are mainly entities which are very similar but not the same and entities that have different names but describe the same entity.

Forbes and Freebase are compared by the shared attributes: company name, country, industries and revenue. The gold standard has 220 pairs in total, 120 of which are false and 100 of which are true. We labeled the types of our corner cases as Abbreviation, Incomplete name, Similar name and Same name with different countries or industries. Some examples of this are:

- Chevron vs Chevron Corporation $\longrightarrow true$
- BP vs TNK-BP $\longrightarrow false$
- Makita (U.S) vs Makita (Japan) $\longrightarrow false$

Freebase and DBpedia: This gold standard has 200 pairs, including 110 false cases and 90 true cases. To create it we chose one company in Freebase and then searched for it in DBpedia. If they matched, we added it as true, otherwise we looked for one which has a similar name or equivalent values in other attributes, for example countries or industries and added it as false. Some examples of corner cases are:

- Okinawa_Electric_Company vs Oki Electric Industry $\longrightarrow false$
- E.ON_Russia vs E.ON $\longrightarrow false$
- Repsol vs Repsol YPF S.A. $\longrightarrow true$
- Wacom_(company) vs Wacom $\longrightarrow true$

DBpedia companies and DBpedia locations has two shared attributes: locationCity and locationCountry. Because city names extracted from DBpedia locations have multiple values due to multiple districts in one city, we defined the city without a specific district name as the value for integration. This gold standard has 270 pairs, 190 of which false and 80 of which are true. Two examples of corner cases are:

- New York vs New York City $\longrightarrow true$
- New_York_City V.S. Syracuse, New York $\longrightarrow false$

2.2 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.

2.3 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 neighborhood 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

2.4 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		

3 Data fusion

3.1 Input data

Table 5 contains the attribute densities and consistencies for each dataset: We used each of the four described in previous sections. One notable attribute is `countries`, which has a density of 1 in all datasets except for Freebase. This is due to the nature of Freebase: We query for company locations, which very often returns a city for which the country is not defined (see [3] for an example). The density of `locations` for Freebase shows a density of 1, supporting this. It also explains why a cross product blocking function returns more matches than partitioning by country when comparing Freebase to another dataset (see table 4). Overall we raised the density especially of `industries`, `countries`, `locations`, `dateFounded` and `keyPeople` considering the high number of companies in the final fused dataset. On the other hand due to the relatively low number of correspondences that include the Forbes dataset the densities for `marketValue` and `continent` are accordingly low.

Attribute	**Forbes (n=2000)**	**Freebase (n=3182)**	**DBpedia (n=16051)**	**Consist-encies**	**Fused (n=6470)**
name	1,00	1,00	1,00	0,97	1,00
countries	1,00	0,40	1,00	1,00	1,00
industries	0,98	0,54	0,61	0,93	0,65
revenue	1,00	0,16	0,15	1,00	0,21
numberOf-Employees	0,00	1,00	0,32	1,00	0,38
dateFounded	0,00	0,81	0,79	0,99	0,82
assets	1,00	0,00	0,06	1,00	0,12
marketValue	1,00	0,00	0,00	1,00	0,06
profit	0,98	0,13	0,00	1,00	0,07
continent	1,00	0,00	0,00	1,00	0,06
keyPeople	0,00	0,31	0,55	0,97	0,59
locations	0,00	1,00	1,00	0,90	1,00

Table 5: Attribute densities and consistencies per dataset

For this project we decided to use the source as provenance data, as querying additional metadata (such as the author or the most recent date modified) for both Freebase and DBpedia would have made the data collection considerably more

[3] http://www.freebase.com/m/0c0bbxc

time consuming. We gave Forbes a higher data quality score than the others because we consider it to be more reliable and probably more up to date compared to Freebase and DBpedia.

3.2 Gold standard

Our fused file contains six entities, each of which has 15 attributes. For a company the revenue, numberOfEmployees, assets, marketValue and profit are always changing over time so in oder to get the most current and reliable values we used external sources. To retrieve the financial attributes for the company "Apple", for example, we used their website to read the latest financial statement[4]. We used a different website "statista" [5] to acquire the numberOfEmployees. Lastly we collected values for assets, marketValue and keyPeople from Forbes[6]. We consider Forbes to offer higher quality and more reliable data than Freebase and DBpedia. For the same reason, we searched for fresh data on Wikipedia[7] for some attributes of a location, for example the area and the elevation.

3.3 Conflict resolution functions

The conflict resolution functions we tried and used for each attribute are listed in table 6. To begin with, the maximum accuracy we achieved with any approach was 0.94 due to some attributes not existing in any of the four datasets or because of very outdated values. The first notable difference between functions occurs for the attribute `industries`. The result of using an intersection between multiple datasets left very few values after fusing, because the industries were often very different. As such we decided to use a union and remove duplicates, thus leading to a more descriptive fused entity. Interestingly, the difference between using an intersection and a union is not as high for `keyPeople` as it is for `industries`. Next we tried both *Max* and *Average* for `numberOfEmployees`. Logically *Max* should be better, since most companies should be growing, in the same sense that a profit should be positive. Looking deeper into this, the company IBM had a lower number of employees in reality than what was recorded in DBpedia, indicating an outdated value in DBpedia and thus leading to a lower accuracy when testing our gold standard.

[4]http://www.apple.com/pr/library/2015/10/27Apple-Reports-Record-Fourth-Quarter-Results.html
[5]http://www.statista.com/statistics/273439/number-of-employees-of-apple-since-2005/
[6]http://www.forbes.com/companies/apple/
[7]https://en.wikipedia.org

Attributes Name	Datasets in which the attribute is found	Conflict resolution function	Accuracy
name	dataset 1, 2, 3	FavourSources	0,94
		Voting	0,94
		LongestString	0,94
country	dataset 1, 2, 3, 4	Voting	0,94
industries	dataset 1, 2, 3	Intersection	0,88
		Union	0,94
revenue	dataset 1, 2, 3	FavourSources	0,94
numberOf-Employees	dataset 2, 3	Average	0,94
		Max	0,93
dateFounded	dataset 2, 3	MostComplete (date)	0,94
		MostComplete (sample)	0,94
		Combination	0,94
assets	dataset 1, 2	FavourSources	0,93
		Max	0,94
marketValue	*dataset 1*	*SingleSource*	*0,94*
profit	dataset 1, 3	FavourSources	0,94
		Max	0,94
continent	*dataset 1*	*SingleSource*	*0,94*
keyPeople	dataset 2, 3	Intersection	0,93
		Union	0,94
locationName	dataset 2, 3, 4	Intersection+ MostComplete	0,93
		Union+ MostComplete	0,94
population	*dataset 4*	*SingleSource*	*0,94*
area	*dataset 4*	*SingleSource*	*0,94*
elevation	*dataset 4*	*SingleSource*	*0,94*

Table 6: Conflict resolution functions. Datasets 1, 2, 3 and 4 correspond to Forbes, DBpedia, Freebase companies and DBpedia locations

When an attribute of the Forbes dataset was involved we tried a favored sources approach to confirm our assumption that the Forbes dataset is more reliable. However, considering the results for `assets`, for example, indicates that this is not always the case. After trying different methods for `dateFounded` we decided to use a combination by choosing the most complete date or the most complete record as a fallback. Using an intersection or union for `locations` seemed to show similar results compared to `keyPeople`. Lastly there are no obvious differences in accuracy in the functions used for `name`. This is due to the fact that very often differences in the name are due to the company type (such as "Inc."), which are filtered out. As such we decided to use *LongestString* in order to keep entities as descriptive as possible.

Conflict resolution functions we implemented include *Intersection* and *Union* for Strings because of the way we stored `industries` and `keyPeople` (delimited by ";;"), *Max*, *MostComplete (date)*, *MostComplete (record)*, a combination of the two, *Intersection* and *Union* combined with *MostComplete (record)*, and *SingleSource*, which is applied when only one source possesses a given attribute.

4 Conclusion

Looking back the datasets we collected from DBpedia and Freebase had sparse attributes. In addition to this the difficulty of collecting the right balance of data from Freebase and DBpedia made data collection a lengthy task. On the other hand once we had created the queries for the two datasets the flexibility of this method allowed us to select and extract exactly the attributes we wanted. As a result we had the opportunity to try many different identity and conflict resolution functions after collecting and mapping the data. During the phase of identity resolution we generated a relatively low number of correspondences of Freebase with Forbes, because many companies in Forbes are not available in our dataset. This is explained by many companies in Freebase not having a `NumberOfEmployees` attribute, for example. Changing this attribute to optional or removing it returns over 200.000 companies from Freebase, which would make the problem of identity resolution and fusion considerably harder. Lastly for the process of fusing our datasets we tried many different conflict resolution functions and implemented a few ourselves. As a result the fused dataset shows improved densities especially among shared attributes. Looking forward, a promising approach to improve accuracy would be to select metadata per attribute from DBpedia and Freebase such as the most recent date modified. This would then be used as a basis for conflict resolution functions relying on provenance data.