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 29, 2015

Contents

List of Figures

List of Tables

1 Introduction to Use Case and Datasets

1.1 Use Case

In this project, our purpose is to integrate a dataset about companies with another dataset about cities their headquarters are located in. However, in order to gather more information about companies, first we combine several datasets together, which are all about companies but come from different sources. Then we will integrate this result with location. As such the resulting integrated dataset may be used for additional information regarding companies and their location around the world.

First, we gather data from each data source. For most datasets we write queries for different web services(DBpedia and Freebase) to request data about companies and location. One dataset is provided as .xls file, so we transfer it into .csv file for mapping. Since a company might have abbreviation in each dataset, we do some transformation on name and country. In second phase, we identify a company in multiple datasets by their overlapping attributes. In order to reduce the comparing time, we use country as blocking key in most situations. Then, by using specific resolution strategies for each attribute, we solve the conflicting information about companies. In the end these datasets can be merged together and represented in the form of our integrated target schema.

1.2 Datasets

1.2.1 Forbes

Forbes is an American business magazine and it 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, profit, assets and market value. This dataset has 2000 entities and 11 attributes which mostly are about financial. Particularly, the value in financial attributes are showed in billion. In addition, this dataset contains official information, compared to DBPedia and Freebase which contains information less complete and less trustworthy.

1.2.2 DBpedia

To access information about companies from DBPedia we plan to use the public SPARQL endpoint (at http://dbpedia.org/sparql). This also makes it possible to filter companies by certain attributes. An example would be to only use companies for this project that have the numberOfEmployees attribute with a value higher than 100. However, in this case this also reduces the number of entities available from DBPedia from 64,255 to 11,966. Using similar parameters for instances of locations, e.g. a populationTotal of at least 10.000, we reduce the number of entities from 725.546 to 43.783. DBPedia also offers its data in the form of a data dump. However, here we may run into technical difficulties based on the size of the files (2.4GB compressed). Also, we would have to set up our own local server with a SPARQL endpoint.

1.2.3 Freebase

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. By setting the same restrictions, e.g. only selecting companies with the number_of_employees attribute, the number of entities is reduced from 283,906 to 3,182. An alternative would be to use a data dump they offer (developers.google.com/freebase/data), however here we run into similar size problems (22GB gzipped, 250GB uncompressed).

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.

```
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 Company

2.1.3 Freebase: Company

Freebase provides web service to query data and return in JSON. The total number of entities is 3,182. We make name nonoptional, since we want to use it to compare companies in each dataset. Also, number_of_employees is nonoptional, because the amount for being optional is around 230,000 and we don’t want to use it. The rest attributes are all optional. First we test queries on the query page to make sure we get the right data. Then, we build a Java project to execute the MQL Read API which provides access to the Freebase database by Metaweb query language (MQL). Additionally, during the mapping procedure, we occur some problem about multiple values in JSON array and JSON object, it couldn’t match in target schema because of the various number of values. Thus, we convert them into one string and separate with two semicolons while excuting Java project.

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.

	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

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)

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

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

As described that we have four datasets in total , we made three types of gold standards . For the convenience of matching , entities in smaller size dataset are compared to the bigger one and all gold standards are selected by stratified distribution in ascending order . The corner cases in our project are mainly about companies which are very similar but not the same and companies have different names but are 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 with 120 false and 100 true . Types of corner cases are divided into abbreviation , Incomplete name , Similar name and Same name with different countries or industries. We took some examples as following:

Abbreviation: Bank of China V.S. Industrial and Commercial Bank of China (Asia) , TRUE
Incomplete name: Chevron V.S. Chevron Corporation , TRUE
Similar name: BP V.S. TNK-BP , FALSE
different country/industry: Makita (U.S) V.S. Makita (Japan)

Freebase and DBpedia: The size of gold standard between Freebase and DBpedia is 200, including 102 false cases and 98 true cases. First we choose one company in Freebase and then search it in DBpedia. If match, then put it in true case. If not, then find one which has a similar name or equivalent values in other attributes, e.g. countries or industries. Here are some examples:
http://dbpedia.org/resource/Okinawa_Electric_Power_Company, Oki Electric Industry, FALSE
http://dbpedia.org/resource/E.ON_Russia, E.ON, FALSE
http://dbpedia.org/resource/Repsol, Repsol YPF S.A., TRUE
http://dbpedia.org/resource/Wacom_(company), Wacom, TRUE

DBPedia companies and DBPedia locations has two shared attributes: location city and location country . Because city name extracted from DBPedia Location has multiple values due to muti-districts in one city , we defined the city without specific district name as the only true value for integration . The total gold standards size for this part is 270 pairs with 190 false and 80 true.

Example of corner cases:
New York V.S. New York City , TRUE New York V.S. Syracuse, New York , FALSE http://dbpedia.org/resource/New_York_City V.S. New York City , TRUE http://dbpedia.org/resource/New_York_City V.S. Syracuse, New York , FALSE

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

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 [1] 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

[1]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 up to date compared to Freebase and DBpedia.

4.2 Gold standard

Our fused file contains five entities and each entity involves 15 attributes.For a company the evenue, numberOfEmployees, assets, marketValue and profit are always changing over time, in oder to get the latest data, we searched for different external sources. Take APPLE for example, we went to the homepage of APPLE to read the latest financial statement[2] and to get the revenue. And then went to statista [3] to acquire numberOfEmployees. And we got assets, marketValue and keyPeople went from Forbes[4]. By comparison, Forbes offers a higher quality data than Freebase and Dbpedia. Because of the same reason, we also searched for relatively fresh data on wikipedia[5] for some attributes of Location, such as population, area and elevation. In a nut shell, DBpedia offers an outdated data and updating frequency is a little low.

4.3 Conflict resolution functions

The conflict resolution functions we tried and used for each attribute are listed in table **??**. 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 oudated value in DBpedia and thus leading to a lower accuracy when testing our gold standard.

[2]http://www.apple.com/pr/library/2015/10/27Apple-Reports-Record-Fourth-Quarter-Results.html

[3]http://www.statista.com/statistics/273439/number-of-employees-of-apple-since-2005/

[4]http://www.forbes.com/companies/apple/

[5]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
numberOfEmployees	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 and DBpedia locations

When an attribute of the Forbes dataset was involved we tried a favoured 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 included *Intersection* and *Union* for Strings because of the way `industries` and `keyPeople` were stored (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.

5 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, making the problem intractable. 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 wouldd then be used as a basis for conflict resolution functions relying on provenance data.

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 29, 2015 Unterschrift