# How Geocoding Tools Can Help Cleaning Data

Christine Rimmert

*christine.rimmert@uni-bielefeld.de*
IWT, Universität Bielefeld, Universitätsstr. 25, Bielefeld, 33615 (Germany)

**Introduction**
Many studies in bibliometrics concern localities of research. The bibliometric databases Web of Science (WoS) and Scopus contain information about this (addresses of authors). Here we will have a closer look at the city part of the address field (in WoS and Scopus). Sometimes it is sufficient to know the cities where research is taking place and even if it is necessary to know the single institutions the city-information can help a lot in assigning address-information to institutions.

There are many difficulties concerning the city-information in the databases (errors, lack of standardization of city names, missing information etc.). Geocoding tools (like Googlemaps, Yahoo Placefinder, Open Street Map) are used a lot in bibliometric research, mostly to create maps of research, but geocoding tools can also be used in another context: they can help to correct, complete or standardize city-information. Here the address-data of WoS and Scopus (year 2008, assigned to Germany in the databases) is used as a sample dataset (4,816 distinct city names, 10,065 distinct postal code–city–combinations after the splitting-step, see below). Yahoo Placefinder (YP) is used as geocoding tool.

**Occurring difficulties**
As mentioned above there are many problems to solve before using the city-information.

In contrast to WoS the city-information in Scopus is not separated into city name and postal code and sometimes there is even no city-information at all. A splitting of the existing city-information (into city name and postal code) and an addition of the missing information should be done.

In the case of using WoS and Scopus data it is useful to transform the Scopus data into the 'WoS-form' (e.g. transform national special characters into the base form). When the preparation described above is done for Scopus data there are numerous problems left, e.g.:

- Wrong country assignments: There is city information assigned to Germany in the databases but belonging to cities in other countries. They are not wanted in studies about German research localities so they have to be identified and excluded.
- Typing errors: There are typing errors in the city-information (due to errors in the original document or in the processing of the databases owners).
- Varieties of names: City names are given with additional information, abbreviations, in different languages etc.
- Other information in the city-field: Sometimes there is no city-information at all or other information is mixed up into the city field (street names, additional cities, states, names of institutions, nonsense...).
- Only postal code given: Sometimes there is only a postal code available, the city name is missing.

**Possible solutions with YP**
In the following it will be discussed how the YP performs on solving the problems mentioned above.

*Splitting*
The postal code could mostly be separated from the city name with help of some obvious regular expressions. 69 distinct city- information strings were not separable by the chosen regular expressions. YP-results for these are shown in Table 1 (GER: city in Germany, Other: cities in other countries).
.

**Table 1.** Results: Splitting.

| City extracted | | | No city extracted |
|---|---|---|---|
| Correct | | Not correct | |
| GER | Other | | |
| 63.77% | 7.25% | 15.94% | 13.04% |

Some items could already be identified as foreign ones in this step due to their special form of postal code.

*Missing data (10,090 items)*
In case of missing city-information, an attempt could be made to extract city information from other fields (e.g. organization1-3,address_part – these fields usually contain names of institutions, departments respectively street names).

**Table 2.** Results: YP as city-extractor (Extr. correct= correct city/extracted city).

| | Org 1 | Org 2 | Org 3 | addr_p |
|---|---|---|---|---|
| Dist. entries | 4,105 | 4,989 | 1,447 | 148 |
| Extr. city | 63.12% | 8.26% | 12.58% | 82.43% |
| Extr. correct | 88.61% | 46.6% | 53.85% | 61.48% |

These results lead to the following procedure: If there is a YP-result for organization1 take it (start with organization1 because 'Extr. correct' is maximal here), if not go to address_part, if there is no YP-result for address_part go on with organization3, if there is no YP-result for organization3 go on with organization2. Results of this combined procedure are shown in Table 3.

**Table 3.** Results: combined procedure.

| City extracted | | No city extracted |
|---|---|---|
| Correct | Not correct | |
| 79.62% | 6.72% | 13.66% |

*Wrong country assignment*
YP should either have no hit or have a hit with another country assignment than Germany, so that these items can be identified. Some of the items belonging to other countries betrayed themselves already in the splitting step because of the special forms of their postal codes (366 total, 122 distinct). Results are shown in Table 4.

**Table 4.** Already detected foreign items.

| City extr. (Germany) | City extr. (other country) | No city extracted |
|---|---|---|
| 4.92% | 89.89% | 5.19% |

In the rest of the sample there are 146 'foreign items'. Results for these are shown in Table 5.

**Table 5**. Not yet detected foreign items.

| City extr. (Germany) | City extr. (other country) | No city extracted |
|---|---|---|
| 13.70% | 15.07% | 71.23% |

So for the most wrong country assignments you can get a hint from YP (no city, foreign country).

*Typing errors/varieties*
A special type of errors are typing errors and a special case of these are errors with a Levenshtein-Distance (LD) one to the original (correct) string. The city information was ordered by the frequency of occurrence and—beginning from the most frequent entries—the ones with LD one to the entry were selected among the less frequent ones. The first was assumed as standard, the less frequent ones as variants of this. This procedure (LDP) delivered 1,171 pairs of variants and standards, 745 of them seemed to be correct (manual control). So LDP performs correct with a rate of ~63.62%.

To increase this the share v_i/s_i can be considered (where v_i is the frequency of the variant and s_i the frequency of the standard of a given pair p_i).

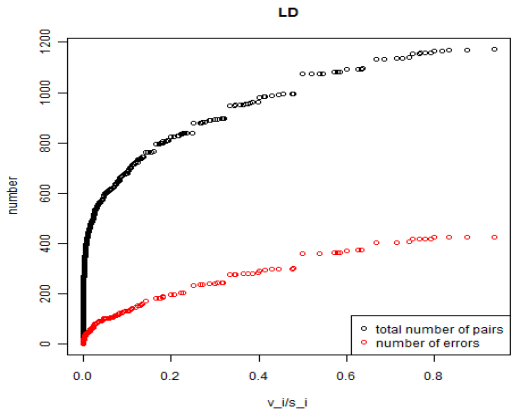


**Figure 1.** Improving LDP by using v_i/s_i

Including only pairs with v_i/s_i < a for appropriate chosen real number a (with 0<a<1, e.g. a=0.2) it is possible to reduce the rate of errors. YP-results are shown in the tables below (correct pairs: Table 6, wrong pairs: Table 7).

**Table 6**. Results LD=1, correct pairs.

| City extracted | | | No city extr. |
|---|---|---|---|
| Correct | | Not correct | |
| YP=LDP | YP≠LDP | | |
| 41.48% | 19.60% | 28.99% | 9.93% |

In the cases 'YP≠LDP' the YP -results were different to the standard but also correct (munie→ München (YP), munich (LDP)).

**Table 7.** Results LD=1, wrong pairs.

| City extracted | | | No city extr. |
|---|---|---|---|
| Correct | Not correct | | |
| | Var =city | Var≠city | |
| 22.54% | 38.73% | 30.04% | 8.69% |

'Var=city': YP gives the variant as city—this is a hint for the variant not being a wrong typed form of the standard but another city with a name written similar to the standard of the pair.

So—given the Variants identified by LDP—YP performs correct with a rate of ~48.68% and is therefore less correct than LDP. Though LDP performs

better than YP here, it is obliged to YP for some help. As mentioned above it can detect lots of LDP-errors, so LDP performs correct with a rate of ~74.06% after having taken that aspect into consideration (or even better taking into account v_i/s_i).

*Other information in the city-field (396 items)*
Sometimes it is given in addition to the city-information, sometimes there is only this "other information" given. Here the YP should extract a city in the first case and should have no hit in the latter case.

**Table 8.** Other information in the city-field.

| City extr. correct | City extr. not correct | No city extr. |
|---|---|---|
| 32.07% | 27.78% | 40.15% |

YP doesn't perform very well in this case (Table 8). Expecting address-data as input it often assigns a city if there is no city information included (e.g. 'Kloster' → YP: city='Gronau') or not clearly identifiable (e.g. input is a street name).

*Only postal code given (143 items)*
In 140 cases YP could find the city name. In the left three cases the given numbers are no valid postal codes in Germany, therefore it is correct to have no result; this gives a clue of the presence of an error.

*Languages*
The output language can be given as a parameter to YP so you can get the results in the desired language, e.g. language='German' will show the result 'München' for the input 'Munich'.

**Conclusion**
It is recognizable from the examples above that YP can help in many cases (though it is not the best choice for all of them)—always keeping in mind the difficulties and characteristics of this tool. Some examples for typical difficulties/characteristics are:

- YP tries to interpret all given data as city-information. Especially if there is no city information in the input string this leads to errors like 'Communication Engineering Lab.' → YP: city='Laboe' or 'International University' → YP: city= 'Bruchsal', but there are also cases in which YP finds the correct city in this case: 'Charité Campus Mitte'→ YP: city='Berlin', 'Forschungsinstitut und Naturmuseum Senckenberg' → YP: city='Frankfurt am Main'.
- The input string should contain country information; otherwise there are many results in other countries. If the country information is not correct YP still shows the right country information in many cases (as shown above).
- YP seems to have problems with too many or too few blanks; in nearly all cases where splitting was not possible, the input string contained an additional blank or a blank was missing (in the city names or postal codes or between city name and postal code). There are also problems with dots, for example: 'Gaswärme-Institut e.V. Essen' → YP: city='Evessen'.
- The YP-output (city name) is not completely standardized, for example: 'Albert Ludwigs University of Freiburg' → YP: city='Freiburg, while 'Albert-Ludwig-Universität Freiburg' → YP:city='Freiburg im Breisgau'.
- As you can see above, there are cases in which YP can't find a city at all.

So YP is certainly a good tool for supporting the cleaning of address-data but does not offer a complete solution to the problem. While it performs well in case of language differences/translations and complementing cities to given postal codes it can only give hints or additional information in other cases (typing errors, missing data)—but this is also very useful. Procedures combined with YP can lead to much better results than without YP (as shown in the LDP-example above).

This is work in progress. The next steps include testing YP also on other countries, combining YP with other methods for cleaning data and an attempt to combine YP and the found YP-combined methods to an algorithm that works as well as possible for all occurring problems/errors of address-data in order to receive a procedure delivering a good preparation of address-data.

**References**
Placefinder: http://developer.yahoo.com/geo/placefinder/ (10.05.2012).