Demonym gazetteer

Alex Pardo and David Sanchez

29th, May 2014

Contents

0.1 Introduction

Demonym or gentilic, is a term for the residents of a locality. How to generate the demonym from a country or city name is not an easy task, and in order to automatize this process we are proposing a system which which will be capable to do it. The next document will show the description of the problem, our system approach, the evaluation and the discussions of the results. Finally, we will explain some possible improvements in our work as a future work.

0.2 Attached files

The structure of the project is separated into two files: demonym.py and functions.py.

- **Demonym.py**: Contains the main code which it calls all the methods from function.py. The methods are the following:
- **Functions.py**: Contains all the methods of the system.
  - *demonyms_downloadDemonymsWP:* Extracts pairs of country-demonym from wp.
  - *demonyms_showHistogram:* Parses the rule file and extracts adding and replacing rules. Finally, shows the histogram.
  - *demonyms_parseFile:* Parses the demonym file.
  - *demonyms_generateRules:* Extracts the rule for a given pair country, demonym.
  - *demonyms_parseCitiesAndCountries:* Download a list of cities and a list of countries from Wikipedia.
  - *demonyms_generateDemonym:* Generates all the possible demonyms from a given place.
  - *demonyms_matchCandidates:* Finds the given demonym candidates in the WP of the given place.
  - *demonyms_findWholeWord:* Finds a whole word.
  - *demonyms_findDemonyms:* Extracts the rules from the files, reads the sample (i.e places) and perfoms the matching.
  - *demonyms_guessEncoding:* Detects coding and transforms to Unicode

Finally, data files need for the system are the **cities.csv**, **countries.csv**, and **demonyms.csv**.

0.3 Description of the problem

As the document mentions in the introduction, the generation of the demonym from a location like country or city is not an easy task. There exist a lot of possible rules and different cases. Some examples of this

process are:

- Africa – African
- Croatia – Croatian (also ”Croat”)
- North / South Korea – North / South Korean
- Hanoi (Vietnam) – Hanoian
- Iran – Iranian (also ”Irani” or ”Persian”)
- Florence – Florentine (also Latin ”Florentia”)

- Ann Arbor – Ann Arborite
- Israel – Israelite (also ”Israeli”)
- Netherlands – Netherlander

As the examples above show, some demonyms are simple a process in which is needed to add or replace ending letters in the original country or city. However, irregular cases are the ones that complicate the task of extracting general rules since are special cases and require a specific norms. We have to assume that we are not going to be able to generate those demonyms that are more difficult because are irregular forms or just because there are few cases.

0.4 Approach

Our system follows the general flow in this type of systems which it is based on two phases: training and test (Figure 1).

In the training phase the system obtains a database of examples (in this case from Wikipedia) of country demonyms[1] and extracts rules from them. In order to remove special cases, a threshold is used in order to filter out those rules which appear less than 3 times.

We have defined two types of rules:

- Adding rules: in which only a suffix is added to the location.
- Substitution rules: in which some letters of the location are substituted by a suffix.

An example of this two types of rules in the pair Africa - African will be:

- ”a” - ”an”
- ” ” - ”n”

Once all the rules are generated and stored, the system extracts two test datasets also from Wikipedia; one is a list of cities in the world and the other a list of countries. Although the source is completely different from the one used in the training phase, some of the locations might be repeated but since the rules do not take into account the location but only the ending of the words it should not influence in the results.

When we have our two test sets, for every location we generate all the demonyms applying all the possible rules and find them in the Wikipedia page of the country/city. Notice that in almost all the city/country pages the demonym appears several times. In some cases, it never appears. Since we are not able to distinguish this two cases, in order to simplify the problem we are going to assume that all the pages have the demonym in the text and it is always correct.

This approach is the most direct and the commonly followed in every machine learning problem. Some decisions were taken in order to simplify the problem and are taken into account in the evaluation made. There are several ways to tackle this problem (e.g. using more complex suffixation techniques) but it makes more sense to begin with the simplest approach and if the results are not as good as expected, increase the complexity of the model. Several times, using a more complex model difficult the task of learning and the results are even worst than with a simpler one.

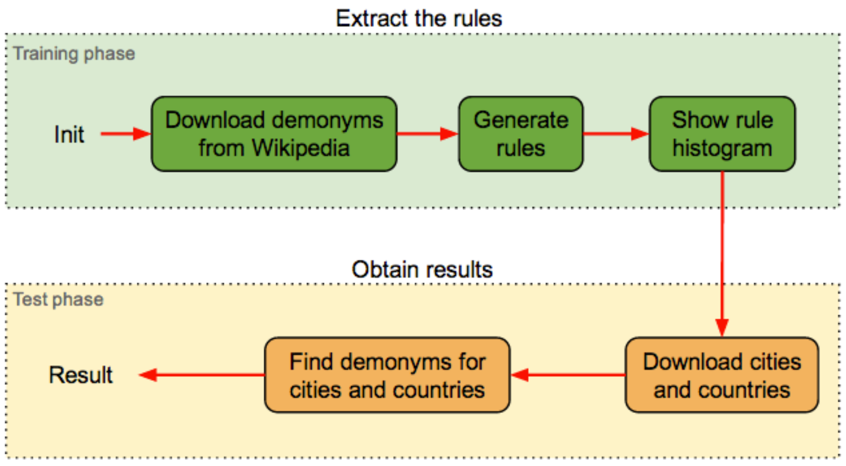
Extract the rules
Training phase
Init
Download demonyms from Wikipedia
Generate rules
Show rule histogram
Obtain results
Test phase
Result
Find demonyms for cities and countries
Download cities and countries

Figure 1: Main architecture of our system.

[1]See http://en.wikipedia.org/wiki/Demonym

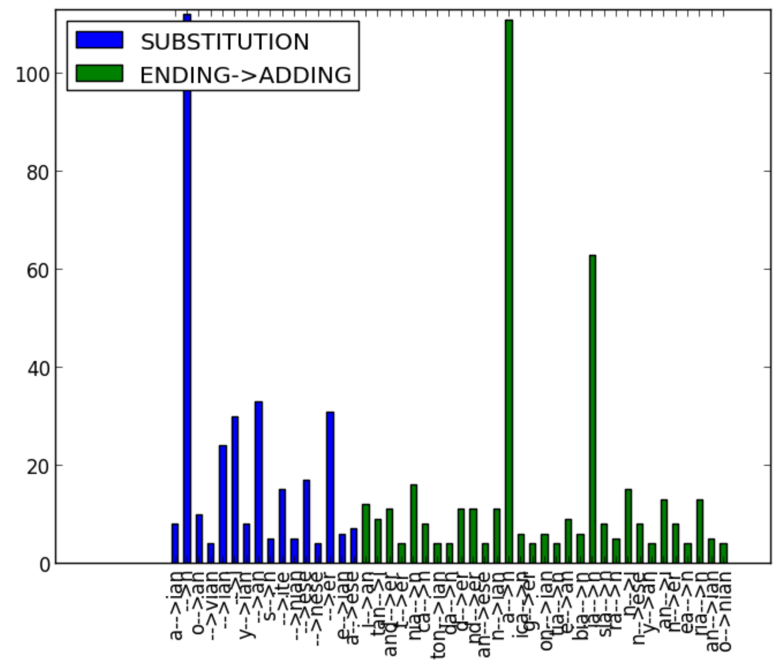

Figure 2: Histogram of the rules.

0.5 Evaluation

As it has been explained in the previous section, we generate all the demonyms using the possible rules and try to find them in the Wikipedia page of the city/country. For simplicity we assume that the correct demonym is always present in the Wikipedia page and that there are no mistakes.

In order to evaluate the algorithm, we simply count the number of demonyms found and the total number of locations we have and compute an Accuracy measure (here we do not have negative results):

$$AC = \frac{true\_positive}{total\_population}$$

As we are assuming that all the examples have a positive match in the text, in this case the Precision measure will have the same value.

0.6 Results

After performing the tests, we have the following results (See *Execution results appendix* for the complete output):

- Cities: 89 matchings over 1751 cities, 5.08% of accuracy.
- Countries: 55 matchings over 89 countries, 61.8% of accuracy.

0.7 Discussion

After looking at the results, one can see that the obtained accuracy in countries is acceptable but the value for cities is really low.

There are three considerations that have to be taken into account, first of all the system is trained only using countries and the obtained rules can be restricted to this types of locations; second, there are several cities with a really short page in Wikipedia (e.g. small cities of Africa or Asia) where the demonym is not appearing; and third, there are several cases of irregular forms, deriving for example from latin forms (Tarragona, Tarraconense).

All this factors are decreasing the accuracy of the system and have to be taken into account in the evaluation.

0.8 Future work

As a results shows, the accuracy of the cities is very lower and one improvement could try to obtain better results with the cities. The system had trained using only countries and maybe we could add other trained version of the system using only cities. The behaviour of the transformations needed to convert the country or city into the demonym sometimes differ if you are using cities and countries. For this reason we could include one system trained with the countries and other with the cities.

Other important improvement will be increase the complexity of the rules models because our rules models are very simple now. Also, the analyse of the word could be deeper using lexemes and so on.

Finally, we could try to generalize the rules in order to reduce the rules which they are very specific.

Appendix

Execution results

Here we have the output of an execution of the whole system:

```
############################
 Extracting rules from WP
############################
410 triples have been obtained


#####################################
Total number of rules:  791
Number of rules (occurences >= 3) :  47


###################################
 Downloading cities and contries
###################################
List of cities in Algeria
17
List of cities and towns in Angola
17
List of cities in Benin
25
List of cities in Botswana
25
List of cities in Burkina Faso
30
List of cities in Burundi
38
List of municipalities in Cameroon
39
List of cities in Cape Verde
42
```

```
List of cities in the Central African Republic
51
List of cities in Chad
55
List of cities in the Comoros
List of cities in the Democratic Republic of the Congo
187
List of cities in the Republic of the Congo
213
List of cities in Djibouti
223
List of cities in Egypt
223
List of towns and villages in Egypt
223
List of cities in Equatorial Guinea
227
List of cities in Eritrea
241
List of cities and towns in Ethiopia
257
List of cities in Gabon
List of cities in the Gambia
257
List of cities in Ghana
314
List of cities in Guinea
List of cities in Kenya
368
List of cities in Lesotho
List of cities in Liberia
368
List of cities in Libya
368
List of cities in Madagascar
368
List of cities in Malawi
381
List of cities in Mali
407
List of cities in Mauritania
413
List of cities in Mauritius
428
List of cities in Morocco
589
List of cities in Mozambique
589
List of cities in Namibia
598
List of cities in Niger
614
List of cities in Nigeria
662
List of cities in Rwanda
672
List of cities in Senegal
696
```

```
List of cities in Seychelles
696
List of cities in Sierra Leone
721
List of cities in Somalia
721
List of cities in South Africa
727
List of cities and towns in the Eastern Cape
735
List of cities and towns in the Free State
748
List of cities and towns in Gauteng
754
List of cities and towns in Limpopo
756
List of cities and towns in Mpumalanga
758
List of cities and towns in the North West Province
767
List of cities and towns in the Northern Cape
771
List of cities and towns in the Western Cape
779
List of cities in South Sudan
795
List of cities in Sudan
816
List of cities in Swaziland
816
List of cities in Tanzania
816
List of cities in Togo
827
List of cities in Tunisia
833
List of cities and towns in Uganda
928
List of cities in Zambia
938
List of cities and towns in Zimbabwe
958
List of cities by continent
958
List of cities in Africa
958
List of cities in North America
964
List of cities in South America
980
List of cities in Europe
980
List of cities in Oceania
980
List of cities in Australia
980
List of cities and towns in Fiji
982
```

```
List of cities in Indonesia
1053
List of cities and villages in Kiribati
List of cities in Papua New Guinea
1053
List of cities in the Marshall Islands
1053
List of cities in Nauru
1100
List of cities in New Zealand
1112
List of cities in Palau
1112
List of cities in Tonga
1112
List of villages and neighbourhoods in Tuvalu
1112
List of cities in Vanuatu
List of cities in Albania
1116
List of cities in Andorra
1144
List of cities and towns in Austria
1159
List of cities and towns in Belarus
1159
List of cities in Belgium
1206
List of cities in Bosnia and Herzegovina
1206
List of cities and towns in Bulgaria
1206
List of cities in Croatia
1272
List of cities in the Czech Republic
1279
List of cities in Denmark by population
1279
List of cities and towns in Estonia
1279
List of cities and towns in Finland
1363
List of cities in Greece
1432
List of cities and towns in Hungary
1445
List of cities and towns in Iceland
1453
List of cities in Ireland
1454
List of cities in Italy by population
1454
List of cities in Kosovo
1455
List of cities in Latvia
1460
List of municipalities in Liechtenstein
1472
```

```
List of cities in Lithuania
1513
List of cities in Luxembourg
1514
List of cities in the Republic of Macedonia
1529
List of cities in Malta
1529
List of cities in Moldova
1555
List of cities in Montenegro
List of cities in the Netherlands by province
1560
List of cities and towns in Poland
1590
List of cities in Portugal
1591
List of cities and towns in Romania
1705
List of cities and towns in Russia
1705
List of cities in San Marino
1705
List of cities in Serbia
1718
List of cities and towns in Slovakia
1718
List of cities in Slovenia
1719
List of municipalities of Spain
1755
List of urban areas in Sweden
1755
List of cities in Switzerland
1847
List of cities in Ukraine
1882
List of cities in the United Kingdom
1882

####################
 Obtaining results
####################
There are 31 add rules and  16 replace rules.
--> Results for countries :

Number of matchings: 55 ; number of samples: 89 ( 61.8 %)

--> Results for cities :
Number of matchings: 89 ; number of samples: 1751 ( 5.08 %)
```