Damegender: Writing and Comparing Gender Detection Tools

David Arroyo Menéndez
GSYC
Universidad Rey Juan Carlos I
d.arroyome@alumnos.urjc.es

Jesús González Barahona
GSYC
Universidad Rey Juan Carlos I
jgb@gsyc.es

Abstract

The variable sex (male or female) is one of most used variables for any study in sociology, but this variable can be hidden in Internet communities. The gender detection from a name is an important problem in Natural Language Processing to decide if a string labeled as name is classified as male or female. An engineer will find useful make gender detection from a name retrieving information from social networks, mailing lists, instant messaging, software repositories, papers, etc. To achieve gender equality and empower all women and girls is a goal in sustanaible development in United Nations, so to measure the gender gap is a previous step to find solutions to reduce it.

Nowadays, there are several Application Programming Interfaces to guess gender from a name. This kind of software has the database based on propietary databases and the software is not free, so some scientific works are difficult to reproduce.

In this paper, we are envisioning how to solve these problems, offering a solution with a free license and open data names from official census useful to replace, use and/or compare these apis with very good results. This tool provides Machine Learning to predict strings, that's useful to guess diminutives or nicknames.

In: A. Editor, B. Coeditor (eds.): Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at http://ceur-ws.org

1 Introduction

There are different ways to detect gender from a person name and perhaps a surname: census, wikipedia, self-references in trust websites, ... The most common way to detect gender from a name is the Application Programming Interfaces with a good popularity, for example, genderapi, namsor, genderize, ... [?]

The problems addressed are:

- Evaluate quality/price with different commercial solutions.
- Think about solutions using free licenses.
- Treatment with names without census, for example, nicknames, diminutives, ...
- Massive gender detection from Internet, for example, mailing lists, software repositories, ...

In this paper, these problems are faced writing a Python solution for:

- To evaluate quality of different solutions applying metrics suggested by [?]
- To understand the current technology in detail, I have developed a tool guessing gender from a name giving support to Spanish and English from the open data census provides by the states.
- To fix the problem with nicknames and diminutives, we have developed a machine learning solution to strings not found in the census dataset.
- To do proof-of-concept tests applying Perceval to detect gender in mailing lists and software repositories.

In Section 2, we explain the current solutions to the problems. In Section 3, we present the results evaluating the current Application Programming Interfaces with our software. In Section 4, we discuss attempts and problems releasing with a free license a gender detection from name program. In Section 5, we discuss how to obtain Open Datasets counting names and gender. In Section 6, we describe our machine learning solution. In Section 7, we describe general implementation details. Finally, in Section 8 we summarize our findings, and describe extensions to the work that we are currently exploring.

3. Application Programming Interfaces

Market

We have reproduced to [?] and updated on 27/06/2019 and we are showing the results in 1.

[]@lllllll@ Feature Gender API genderguesser genderize.io NameAPI NamSor Damegender-Database size $431*10^6 48.528114 * 10^6 1.428.3454407 * 10^6 197.271$[1] *Regulardataupdatesyesnoyesyesyesyes, devU[illegible]nesyesnono* *Latinalphabetspartiallynopartiallyyesyesno Geo − localizationyesnonoyesyesnoExistslocaleyesyesyesyesyes[illegible]probabilisti*

All solutions have increased the database size from [?]. NameAPI and GenderAPI is reaching more features. The tools with a free license have not many features, so for now that will not be the trend in many situations. Today, one good solution quality and price is Namsor, which provides unlimited names through an Application Programming Interface with many features in the task to detect gender from the name.

Reproducing accuracies and confusion matrix

[?] explains different ways to determine gender from a name by humans and it gives 7000 names applying these methods. In this dataset the gender is classified as male, female or unknown. We have used this dataset, but only male and female to these experiments. We are showing the results in the next table:

[]@lllllll@ API Accuracy Precision F1score RecallGenderapi 0.9687686966482124 0.9717050018254838 0.9637877964874163 1.0Genderize 0.926775 0.9761303240374678 0.9655113956503119 1.0Damegender (SVC)[2] 0.8791969539633091 0.9718767935718385 0.9718767935718385 1.0Namsor 0.8672551055728626 0.9730097087378641 0.9236866359447006 1.0Nameapi 0.8301886792452831 0.97420272191753 0.9054181612233341 1.0Gender Guesser 0.7743554248139817 0.9848151408450704 0.8715900233826968 1.0Different accuracies measures

In 1 Genderapi and Genderize are obtaining the best results, although all solutions is reaching results better than 0.8 except Gender Guesser.

[]@lllll@ APIs gender male female undefined Genderapi male 3589 155 67 female 211 1734 23 Damegender (SVC)[1] male 3663 147 0 female 551 1497 0 Genderguesser male 3326 139 346 female 78 1686 204 Namsor male 3325 139 346 female 78 1686 204 Genderize male 3157 242 412 female 75 1742 151 Nameapi male 2627 674 507 female 667 1061 240 Confusion matrix tables by APIs

With Damegender has been done a comparison about confusion matrix tables depending the API (see 1). If we compare these results with the results obtained in [?], we can understand that the results are similar.

Genderapi has similar results, but it is being improved the undefined results. In Genderguesser is we are obtaining different results and it is strange, because the software has not modified from some years ago. In Genderize we are obtaining the same results. In Nameapi the guessed results is changing from male to female with more errors. In Namsor the results is so similar. Damegender is not guessing undefined because we predict with machine learning if the string is not in the database.

The most important tools Namsor, Genderapi and Genderize are improving the accuracies with respect the previous comparison.

[]@llll@ API error code error code without na na coded error gender biasDamegender (SVC)[1] 0.121 0.121 0.0 -0.07GenderApi 0.167 0.167 0.0 -0.167Gender Guesser 0.225 0.027 0.204 0.003Genderize 0.276 0.261 0.0204 -0.0084 Namsor 0.332 0.262 0.095 0.01 Nameapi 0.361 0.267 0.129 0.001 APIs and Errors

In the table it is possible to observe a high index of errors in Nameapi and Namsor and a low index of errors in GenderApi and Damegender.

Datasets

We can divide the next options choosing a dataset: (1) a census published with a free license (open census way), (2) a dataset done by scientist with a paper in a magazine (scientific way), (3) a dataset released with a free license in a free software package (free software way), (4) a dataset retrieved from commercial Application Programming Interfaces (commercial api way).

```
$ python3 main.py David --total="ine"
David gender is male
```

[1] This cipher is the sum of names in United States of America, United Kingdom, Uruguay and Spain

[2] SVC is the acronym of Support Vector Classification, the Machine Learning algorithm that Damegender was using with this results

```
363559  males for David from INE.es
0 females for David from INE.es
```

In Damegender, we are including Open Data census about names and gender, such as INE.es or USA and United Kingdom (births and dies). We want datasets provided by the software package to increment the speed retrieving data.

From the user final point of view, the value of using Open Data is give a good explanation when we are asking about the gender from a name (number of males and females using a specific name in a country) versus a probability created by the way explained in [?] or similar.

From the scientific point of view, the value of using Open Data is to allow that the experiment can be reviewed by peers on an automatic and legal way (using proprietary data the reviewer should request it separately to make the review).

A second approach is to build the dataset reviewing the names in scientific personal sites, Wikipedia, ... [?]. This approach is valid, but it consumes many time and efforts, although could be useful if there not a legal way to build the dataset.

A third approach is using a dataset from a popular free software solution. For instance, Natural Language Tool Kit is providing 8000 labeled english names. The classification is male or female. The problem again is about don't retrieve data with the social science quality of National Statistics Institutes. Another example is Gender Guesser a good dataset for international names with different categories to define the probability. This approach is similar to use a dataset released with a paper in a journal, the advantage is to understand and add new names with a solid criteria accepted by the scientific community.

We are using the census approach as base of truth to distinguish if a name is male or female in a geographical area. Generally, a name has a strong weight to determine if it's a male or a female on this way, for instance, David is registered 365196 times as male and 0 times as female in Spain National Institute of Statistics.

Many countries don't provide Open Data census about gender and names, but we envisioned build a Dataset about names and gender free and universal working from Gender Guesser dataset and Wikidata as solution. Perhaps, to complete this work we need automate humans process described in [?].

The last approach is based on to trust on commercial solutions, such as we trust on search engines to make searches in Internet (black box). In Damegender we can download json files from main commercial Application Programming Interfaces (API) solutions (genderapi, genderize, namsor, nameapi, ...). One user can build proprietary datasets on this way using an average weighted by the precision or accuracy of each Application Programming Interface measured with Damegender with an open dataset as base of truth.

2 Machine Learning

We have developed a script infofeatures.py with our datasets to analyze data about features. The datasets used in this experiment was retrieved from official datasets from national statistical institutions in Spain, Uruguay, United Kingdom, USA. The features used are: first letter, last letter, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, vocals, consonants, first letter, first letter vocal, last letter vocal, last letter consonant, last letter a. The choosing of features was verified with Principal Component Analysis.

Take a look to the most informative features results with the different datasets: []@lllllll@ Dataset Char A Last Char A Last Char O Last Char Consonant Last Char Vocal First Char Consonant First Char Vocal Uruguay (F) [3] 0.816 0.456 0.007 0.287 0.712 0.823 0.177 Uruguay (M) [3] 0.643 0.249 0.062 0.766 0.234 0.771 0.228 Spain (F) [3] 0.922 0.588 0.03 0.271 0.728 0.772 0.228 Spain (M) [3] 0.818 0.03 0.268 0.569 0.43 0.763 0.236 UK (F) [3] 0.825 0.374 0.013 0.322 0.674 0.765 0.235 UK (M) [3] 0.716 0.036 0.039 0.78 0.218 0.799 0.2 USA (F) [3] 0.816 0.456 0.007 0.287 0.712 0.823 0.177 USA (M) [3] 0.643 0.02 0.061 0.765 0.234 0.84 0.159 Informative Features in Different Countries

The countries where the main language is spanish (Uruguay + Spain) and english (USA + United Kingdom + Australia) where is having very similar variation with the features chosen between males and females with these datasets (remember that this datasets are being extracted from official statistics provided by the states). In Canada, a country french centric has different rules with this features.

The letter a is varying 0.2 from males to females in (USA and Uruguay) and 0.1 from males to females (United Kingdom, Australia and Spain). The last letter a is varying 0.5 from males to females in (Australia, Spain) around 0.4 in (USA, United Kingdom) and 0.2 in Uruguay. The last letter o from females to males is varying 0.2 in (Spain, Australia) and is equal in (Uruguay, USA, United Kingdom). For the last letter consonant all countries is giving the result that is for males, with results from 0.2 to 0.5: Uruguay and USA (0.5), United Kingdom (0.4), Australia and Spain (0.3). So last letter vocal is reverse tha last letter consonant. First letter consonant or first letter vocal is

[3] F is for females and M is for males

a non significative feature due to so similar results in english and spanish.

The success with the different algorithms is showed in the next table:

[]@lllll@ Machine Learning Algorithm Accuracy Precision F1score Recall

Support Vector Machines 0.879 0.972 0.972 1.0 Random Forest 0.862 0.902 0.902 1.0 NLTK (Bayes) 0.862 0.902 0.902 1.0 Multinomial Navie Bayes 0.782 0.791 0.791 1.0 Tree 0.764 0.821 0.796 1.0 Stochastic Gradient Distribution 0.709 0.943 0.815 1.0 Gaussian Naive Bayes 0.709 0.968 0.887 1.0 Bernoulli Naive Bayes 0.699 0.965 0.816 1.0 AdaBoost 0.698 0.965 0.815 1.0 Multi Layer Perceptron 0.677 0.819 0.755 1.0

Machine Learning Algorithms and accuracies measures

The results in 2 shows that using algorithms as Support Vector Machines or Random Forest against a scientific dataset created by independent researchers where it is possible to reach results similar to another commercial solutions about gender detection tools. Our classifier is binary (only male and female).

We were doing this experiment with NLTK and INE datasets with accuracies reaching accuracies until 0.745. So it makes sense expect better results in random datasets augmenting languages and countries. Due to our solution is not providing arabic or chinesse alphabets, yet.

So, it's possible infer that Damegender provides a good solution for nicknames, diminutives, or similar.

3 State of Art

3.1 Comparing Commercial Solutions

A standard commercial Application Programming Interface (API) can guess the gender for a single name or a list of names (from a CSV file or an API call). To express geolocalization you can give surnames, a country ISO code, or a language. Generally, you can give a probability and a counter associated to a name and gender in a certain population.

[?] are proposing a good metrics set to classify these commercial Application Programming Interfaces (features, measuring errors and success, ...). The features observed are: Database size (January 2018), Regular data updates, Handles unstructured full name strings, Handles surnames, Handles non-Latin alphabets, Implicit geo-localization, Assignment type, Free parameters, Open source, Application Programming Interface, Monthly free requests, Monthly subscription cost (100,000 requests/month)).

In the commercial tools is being used different ways to express probability (confidence, scale, accuracy, precision, recall, ...).

3.1.1 Datasets

In [?] a world was envisioned where public structured data could be interconnected with software agents to process these data, perhaps only recovering information, but mixed with distributed artificial intelligence would give a big jump to the semantic richness to the web.

[?] shows serious profits for the states adopting Open Data in three categories (1) political and social, (2) economical, (3) operational and technical. So, Open Data is a breakthrough towards the Semantic Web.

We can find Open Data about names and gender in census of citizens in states and commercial solutions. Free software packages such as [?] or [?] is providing good datasets about names and gender. So, Damegender incorporates different lists of names from free software solutions wrote before (Natural Language ToolKit, Gender Guesser, ...) and from Open Data census (United Kingdom, USA, Spain, Uruguay, ...).

Wikidata [?] provides a semantic and open database about Wikipedia allowing retrieve information with Sparql, such as names and gender.

[?] describes different ways to build a dataset on hand looking for names in papers, scientific websites, wikipedia, biographies, photos, ...)

Free Software

Before Damegender, only [?] was competing as Free Software solution with the main commercial Application Programming Interfaces about gender detection from the name. The best contribution is the dataset containing 48528 names with a good classification by countries.

More software about gender

In some studies, for example, about Twitter or Github, some people can choose between different ways to detect gender (not only names). So, we can find gender detection tools from faces in images ([?]), from hand written ([?]), or from speeches ([?]).

Massive Gender Detection

There are good studies measuring gender in Internet. Some studies are about gender gap in general ([?], [?], [?]), Twitter ([?], [?]) Stackoverflow ([?]), Wikipedia ([?], [?]), Github ([?]) ...

Implementation

We have chosen Python free software tools with a good scientific impact. Natural Language Toolkit for Natural Language Processing [?]. Scikit for Machine Learning [?]. Numpy for Numerical Computation [?]. Matplotlib to visualize results [?]. And Perceval [?] to retrieve information in mailing lists and repositories.

The current result is a Python package contributed to pip to be used from the console.

The software is using an oriented to objects design with unit testing for classes and methods using nose and unit testing for Python commands using Bash.

A summary of current features in the software are:

[noitemsep]To deduce the gender about a name in Spanish or English (current status) from open census in local. To decide about males and females in strings using different machine learning algorithms. To use the main solutions in gender detection (genderize, genderapi, namsor, nameapi and gender guesser) from a command. To research about why a name is related to males or females with statistics. We provide Python commands about study and compare gender solutions with: confusion matrix, accuracies, error measures. And to decide about features: statistical feature weight, principal component analysis, ... To determine gender gap in free software repositories or mailing lists (proof of concept)

Conclusions

The market of gender detection tools is dominated by companies based on payment services through Application Programming Interfaces with good results. This market could be modified due to Free Software tools and Open Data giving more explicative results for the user.

Although machine learning techniques are not new in this field, we are giving an approach to guess strings not found in a dataset that currently is classified as unknown and the humans trend to think in gender terms many strings calling it as nicknames or diminutives.

These previous advances in computer science could be giving support to study the gender gap in repositories and mailing lists. Another future work is to create a free and universal dataset with support for all languages and cultures.