Damegender: Writing and Comparing Gender Detection Tools

David Arroyo Menéndez

April 28, 2020

Damegender: Writing and Comparing Gender Detection Tools

David Arroyo Menéndez[1]

[1]**Universidad Rey Juan Carlos**

ABSTRACT

The gender detection from a name is useful to make gender studies from social networks, mailing lists, software repositories, papers, etc. These studies are required to measure the gender gap and 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 tool with a free license to use and compare these apis. This tool provides Machine Learning to predict strings, that's useful to guess diminutives or nicknames.

Keywords: Gender gap, Gender detection tools, Software repositories

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, ... Santamaría and Mihaljević (2018)

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 being tackling writing a Python solution for:

- To evaluate quality of different solutions applying metrics suggested by Santamaría and Mihaljević (2018)
- 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 census.
- 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.

2. STATE OF ART

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.

Santamaría and Mihaljević (2018) 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, ...).

Datasets

In Berners-Lee et al. (2001) 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.

Janssen et al. (2012) 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 Krawetz (2006) or Loper and Bird (2002) 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, ...).

Wikidata Vrandečić and Krötzsch (2014) provides a semantic and open database about Wikipedia allowing retrieve information with Sparql, such as names and gender.

Santamaría and Mihaljević (2018) describes different ways to build a dataset on hand looking for names in papers, scientific websites, wikipedia, biographies, photos, ...)

Free Software

Before Damegender, only Krawetz (2006) 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 (Ranjan et al. (2017)), from hand written (Liwicki et al. (2011)), or from speeches (Koppel et al. (2002)).

Massive Gender Detection

There are good studies measuring gender in Internet. Some studies are about gender gap in general (Robles et al. (2014), Holman et al. (2018), Dollar and Gatti (1999)), Twitter (Burger et al. (2011), Mislove et al. (2011)) Stackoverflow (Vasilescu et al. (2012)), Wikipedia (Antin et al. (2011), Hill and Shaw (2013)), Github (Vasilescu et al. (2015)) ...

3. APPLICATION PROGRAMMING INTERFACES

Market

We have reproduced to Santamaría and Mihaljević (2018) and updated on 27/06/2019 and we are showing the results in 1

Feature	Gender API	genderguesser	genderize.io	NameAPI	NamSor	Damegender
Database size	$431*10^6$	48.528	$114*10^6$	1.428.345	$4407*10^6$	57.282
Regular data updates	yes	no	yes	yes	yes	yes, dev
Unstructured strings	yes	no	no	yes	no	yes
Handles surnames	yes	no	no	yes	yes	yes
Non-Latin alphabets	partially	no	partially	yes	yes	no
Geo-localization	yes	no	no	yes	yes	no
Exists locale	yes	yes	yes	yes	yes	yes
Assingment type	probabilistic	binary	probabilistic	probabilistic	probabilistic	prob
Free params	total, prob	gender	total, prob	confidence	scale	total, prob
Guessing with ML	no	no	no	no	no	yes
Free license	no	yes	no	no	no	yes
API	yes	no	yes	yes	yes	future
free requests limited	yes (200)	unlimited	yes (1000)	yes	yes	unlimited

Table 1. Features and gender detection tools by name

All solutions have increased the database size from Santamaría and Mihaljević (2018). 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.

Measuring success and errors in gender detection tools from the name

To guess the sex, we have an true idea (example: female) and we obtain a result with a method (example: using an api, querying a dataset or with a machine learning model). The guessed result could be male, female or perhaps unknown. To remember some vocabulary:

True positive is finding a value guessed as true if the value in the data source is positive.

True negative is finding a value guessed as true if the the value in the data source is negative.

False positive is finding a value guessed as false if the the value in the data source is positive.

False negative is finding a value guessed as false if the the value in the data source is negative.

In ISO (1994), we can find a vocabulary for measure true, false, success and errors. We can make a summary in the gender name context about mathematical concepts:

Precision is about true positives between true positives plus false positives

```
(femalefemale + malemale ) /
(femalefemale + malemale + femalemale)
```

Recall is about true positives between true positives plus false negatives.

```
(femalefemale + malemale ) /
(femalefemale + malemale + malefemale + femaleundefined + maleundefined)
```

Accuray is about true positives between all.

```
(femalefemale + malemale ) /
(femalefemale + malemale + malefemale +
 + femalemale + femaleundefined + maleundefined)
```

The **F1 score** is the harmonic mean of precision and recall taking both metrics into account in the following equation:

```
2 * (
(precision * recall) /
(precision + recall))
```

In Damegender, we are using accuracy.py with the different measures (precision, recall, accuracy and f1 score) in different apis from an input. For instance:

```
$ python3 accuracy.py --api="damegender" --measure="recall"
  --csv=files/names/allnoundefined.csv
$ python3 accuracy.py --api="damegender" --measure="precision"
  --csv=files/names/allnoundefined.csv
```

Error coded defines if the true is different than the guessed. That's divide the number of elements with errors by the total number of elements:

```
(femalemale + malefemale + maleundefined + femaleundefined) /
(malemale + femalemale + malefemale +
 + femalefemale + maleundefined + femaleundefined)
```

Error coded without na defines if the true is different than the guessed, but without undefined results. That's divide the number of elements with undefined errors by the total number of elements

```
(maleundefined + femaleundefined) /
(malemale + femalemale + malefemale +
femalefemale + maleundefined + femaleundefined)
```

Error gender bias allows to understand if the error is bigger than guessing males than females or viceversa. That's males guessed as females minus females guessed as males and this number divided by the total number of elements not guessed as undefined.

```
(malefemale - femalemale) /
(malemale + femalemale + malefemale + femalefemale)
```

The weighted error defines if the true is different than the guessed, but giving a weight to the guessed as undefined.

```
(femalemale + malefemale +
+ w * (maleundefined + femaleundefined)) /
(malemale + femalemale + malefemale + femalefemale +
+ w * (maleundefined + femaleundefined))
```

In Damegender, we have coded errors.py to implement the different definitions in different apis.

```
$ python3 errors.py --api="damegender" --csv=files/names/allnoundefined.csv
Damegender with files/names/allnoundefined.csv has:
+ The error code: 0.2547594323295258
+ The error code without na: 0.2547594323295258
+ The na coded: 0.0
+ The error gender bias: -0.04949809622706819
```

In the **confusion matrix** the rows of the datasource element are true and in the columns the elements are identified as guess.

```
[[ 2, 0, 0]
 [ 0, 5, 0]]
```

It means, I have 2 females true and I've guessed 2 females and I've 5 males true and I've guessed 5 males. I don't have errors in my classifier.

```
[[ 2  1  0]
[ 2 14  0]
```

It means, I have 2 females true and I've guessed 2 females and I've 14 males true and I've guessed 14 males. 1 female was considered male, 2 males was considered female.

In Damegender, we have coded confusion.py to implement this concept with the different apis.

```
$ python3 confusion.py --csv="files/names/allnoundefined.csv"
```

Reproducing accuracies and confusion matrix

Santamaría and Mihaljević (2018) 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 2

API	Accuracy	Precision	F1score	Recall
Genderapi	0.9687686966482124	0.9717050018254838	0.9637877964874163	1.0
Genderize	0.926775	0.9761303240374678	0.9655113956503119	1.0
Namsor	0.8672551055728626	0.9730097087378641	0.9236866359447006	1.0
Nameapi	0.8301886792452831	0.97420272191753	0.9054181612233341	1.0
Gender Guesser	0.7743554248139817	0.9848151408450704	0.8715900233826968	1.0
Damegender	0.7452405676704742	0.8789548887528067	0.8789548887528067	1.0

Table 2. Different accuracies measures

In 2 Genderapi and Genderize are obtaining the best results and the free solutions (Gender Guesser and Damegender) the worst results. We hope improve Damegender augmenting languages.

APIs	gender	male	female	undefined
Genderapi	male	3589	155	67
	female	211	1734	23
Genderguesser	male	3326	139	346
	female	78	1686	204
Genderize	male	3157	242	412
	female	75	1742	151
Nameapi	male	2627	674	507
	female	667	1061	240
Namsor	male	3325	139	346
	female	78	1686	204
Damegender	male	3033	778	0
	female	276	1692	0

Table 3. Confusion matrix tables by APIs

With Damegender has been done a comparison about confusion matrix tables depending the API (see 3). If we compare these results with the results obtained in Santamaría and Mihaljević (2018), 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.

API	error code	error code without na	na coded	error gender bias
Genderize	0.0727	0.053	0.02	-0.008
GenderApi	0.167	0.167	0.0	-0.167
Namsor	0.167	0.167	0.0	0.167
Damegender	0.255	0.255	0.0	-0.049
Gender Guesser	0.225	0.027	0.204	0.003
Nameapi	0.361	0.267	0.129	0.001

API	error code	error code without na	na coded	error gender bias

Table 4. APIs and Errors

In the table it is possible to observe a high index of errors in nameapi and a low index of errors in Genderize, GenderApi and Namsor.

4. 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
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.

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 Santamaría and Mihaljević (2018) 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 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 363559 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 Santamaría and Mihaljević (2018).

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.

5. MACHINE LEARNING

These results are experimental, we are improving the choosing of features and datasets. The datasets used in this experiment was retrieved from Spain National Institute of Statistics and in Natural Language ToolKit corpus names (this dataset is about english names). 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.

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

Machine Learning Algorithm	Accuracy	Precision	F1score	Recall
Tree Decision	0.745	0.879	0.879	1.0
Stochastic Gradient Distribution	0.703	0.704	0.704	1.0
Gaussian Naive Bayes	0.703	0.704	0.704	1.0
Support Vector Machines	0.698	0.994	0.994	1.0
Multi Layer Perceptron	0.677	0.82	0.756	1.0
Bernoulli Naive Bayes	0.522	0.989	0.747	1.0
Multinomial Naive Bayes	0.406	0.99	0.576	1.0

Table 5. Machine Learning Algorithms and accuracies measures

The results in 5 show that using Tree Decision for English and Spanish is possible to reach results similar to another commercial solutions about gender detection tools. Our classifier is binary (only male and female).

It makes sense expect better results augmenting languages and countries.

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

6. IMPLEMENTATION

We have chosen Python free software tools with a good scientific impact. Natural Language Toolkit for Natural Language Processing Loper and Bird (2002). Scikit for Machine Learning Pedregosa et al. (2011). Numpy for Numerical Computation Van Der Walt et al. (2011). Matplotlib to visualize results Hunter (2007). And Perceval Dueñas et al. (2018) 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:

- 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)

7. 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 tendency 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.

REFERENCES

Antin, J., Yee, R., Cheshire, C., and Nov, O. (2011). Gender differences in wikipedia editing. In *Proceedings of the 7th international symposium on wikis and open collaboration*, pages 11–14. ACM.

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37.

Burger, J. D., Henderson, J., Kim, G., and Zarrella, G. (2011). Discriminating gender on twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1301–1309. Association for Computational Linguistics.

Dollar, D. and Gatti, R. (1999). *Gender inequality, income, and growth: are good times good for women?*, volume 1. Development Research Group, The World Bank Washington, DC.

Dueñas, S., Cosentino, V., Robles, G., and Gonzalez-Barahona, J. M. (2018). Perceval: Software project data at your will. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, pages 1–4. ACM.

Hill, B. M. and Shaw, A. (2013). The wikipedia gender gap revisited: Characterizing survey response bias with propensity score estimation. *PloS one*, 8(6):e65782.

Holman, L., Stuart-Fox, D., and Hauser, C. E. (2018). The gender gap in science: How long until women are equally represented? *PLoS biology*, 16(4):e2004956.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90.

ISO (1994). Accuracy (trueness and precision) of measurement methods and results — part 1: General principles and definitions. ISO 5725-1:1994, International Organization for Standardization, Geneva, Switzerland.

Janssen, M., Charalabidis, Y., and Zuiderwijk, A. (2012). Benefits, adoption barriers and myths of open data and open government. *Information systems management*, 29(4):258–268.

Koppel, M., Argamon, S., and Shimoni, A. R. (2002). Automatically categorizing written texts by author gender. *Literary and linguistic computing*, 17(4):401–412.

Krawetz, N. (2006). Gender guesser.

Liwicki, M., Schlapbach, A., and Bunke, H. (2011). Automatic gender detection using on-line and off-line information. *Pattern Analysis and Applications*, 14(1):87–92.

Loper, E. and Bird, S. (2002). Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Mislove, A., Lehmann, S., Ahn, Y.-Y., Onnela, J.-P., and Rosenquist, J. N. (2011). Understanding the demographics of twitter users. In *Fifth international AAAI conference on weblogs and social media*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Ranjan, R., Patel, V. M., and Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135.

Robles, G., Arjona Reina, L., Serebrenik, A., Vasilescu, B., and González-Barahona, J. M. (2014). Floss 2013: A survey dataset about free software contributors: challenges for curating, sharing, and combining. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 396–399. ACM.

Santamaría, L. and Mihaljević, H. (2018). Comparison and benchmark of name-to-gender inference services. *PeerJ Computer Science*, 4:e156.

Van Der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22.

Vasilescu, B., Capiluppi, A., and Serebrenik, A. (2012). Gender, representation and online participation:

A quantitative study of stackoverflow. In *2012 International Conference on Social Informatics*, pages 332–338. IEEE.

Vasilescu, B., Posnett, D., Ray, B., van den Brand, M. G., Serebrenik, A., Devanbu, P., and Filkov, V. (2015). Gender and tenure diversity in github teams. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 3789–3798. ACM.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.