Damegender: Writing and Comparing Gender Detection Tools

David Arroyo Menéndez[1]

[1]**Universidad Rey Juan Carlos**

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.

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 faced 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 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.

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 successes, ...). 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)).

It's possible different ways to express probability about the succesful (confidence, scale, accuracy, precision, recall, ...), confusion matrix to understand where it's successes or fails and differente errors measures (error coded, error coded without not applicable values, error gender bias, not applicable coded).

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).

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 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 to build the dataset reviewing the names in scientific personal sites, Wikipedia, ... Santamaría and Mihaljević (2018). 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 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.

Some Open Datasets, such Spain (INE.es), or United States of America is providing support to surnames and the relationship with the ethnicity. For example, in United States of America is giving a probability with the race and in Spain INE.es is giving the number of people with a surname with a nationality different to the spanish nationality. The propietary solutions is allowing infer the gender with the surname as another parameter, due to names such as Andrea changes the gender depending the nationality: in Italia would be male, but in Spain would be female.

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)) ...

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 the next table:

API	Accuracy	Precision	F1score	Recall
Genderapi	0.9687686966482124	0.9717050018254838	0.9637877964874163	1.0
Genderize	0.926775	0.9761303240374678	0.9655113956503119	1.0
Damegender (SVC)[1]	0.8791969539633091	0.9718767935718385	0.9718767935718385	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

Table 1. Different 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.

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

Table 2. Confusion matrix tables by APIs

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

With Damegender has been done a comparison about confusion matrix tables depending the API (see 2). 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
Damegender (SVC)[1]	0.121	0.121	0.0	-0.07
GenderApi	0.167	0.167	0.0	-0.167
Gender Guesser	0.225	0.027	0.204	0.003
Genderize	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

Table 3. 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.

5. 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:

Dataset	char a	last char a	last char o	last char consonant	last char vocal	first char consonant	firs
Uruguay (F) [2]	0.816	0.456	0.007	0.287	0.712	0.823	0.1
Uruguay (M) [3]	0.643	0.249	0.062	0.766	0.234	0.771	0.2
Spain (F) [3]	0.922	0.588	0.03	0.271	0.728	0.772	0.2
Spain (M) [3]	0.818	0.03	0.268	0.569	0.43	0.763	0.2
UK (F) [3]	0.825	0.374	0.013	0.322	0.674	0.765	0.2
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	0.1
USA (M) [3]	0.643	0.02	0.061	0.765	0.234	0.84	0.1

Table 4. Informative Features in Different Countries

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

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 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:

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

Table 5. Machine Learning Algorithms and accuracies measures

The results in 5 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.

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 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.

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.