[Artifact Presentation] Damegender:
Writing and Comparing Gender Detection Tools

David Arroyo Menéndez, Jesus González-Barahona, Gregorio Robles

Grupo de Sistemas y Comunicaciones (GSyC)

Universidad Rey Juan Carlos, Madrid, Spain

{d.arroyome@alumnos, jgb@gsyc, grex@gsyc}.urjc.es

Abstract

Diversity in software development teams has been identified as one of the main ingredients of a more productive and healthy software community. Thus, the interest of the research community in identifying who is contributing to a project has increased in the last years. In the software domain, and although other types of diversity exist, this is particularly important for the case of gender. Given the large amount of publicly available data on the software development process that can be retrieved and analyzed from the Internet (e.g., GitHub, StackOverflow), the importance of having methods and tools that help to identify the participation of women within large amounts of data is desirable. In this paper we present a free software tool, called `damegender`, which we have conceived to infer the gender from the name of a participant. `damegender` is based on open databases from official census and uses Machine Learning. An experimental branch is currently under development for those cases where instead of names we have diminutives or nicknames. We have compared `damegender` with other name-to-gender inference tools, obtaining good results. We hope the `damegender` tool can become a cornerstone for the scientific advancement of the study of gender (including the gender gap) and diversity in Information Technology (IT),

and in particular in the free software community.

1 Introduction

In recent times, many research investigations have been made on gender diversity in the IT domain. Examples of these efforts range from the participation in Twitter [BHKZ11, MLA+11], in Wikipedia [AYCN11, HS13], in science [HSFH18, DG99], and more specifically in the software domain in StackOverflow [VCS12, VPR+15], GitHub [VPR+15], and free software development [RARS+14, IHSR18, LC19, CPT+19].

The interest on gender diversity has become more and more relevant, and so does the identification methods that allow us to perform comprehensive studies on gender representation, given the large amounts of data available, in particular from collaborative environments.

The most used method to infer the gender of participants is by looking at their name. However, there are some studies about Twitter and GitHub where not only names are used to detect gender. In those studies we can find gender detection tools that infer the gender from faces in images [RPC17], from hand-written annotations [LSB11], or from speeches [KAS02].

Currently, there are different ways to detect gender from a person's name (and maybe other personal data, such as the surname, the geographic location, etc.). A first one is based on data extracted from the Wikipedia, self-references in trust websites, searches in Google Images, among others. Another way to do it is by using one of the existing Application Programming Interfaces (APIs). This paper is about the latter, focusing on their possibilities and limitations. Therefore, (i) we evaluate the quality (and accessibility, including price) of different existing solutions; (ii) we discuss their limitations; and (iii) we investigate what happens with those names not included in a census, for

example, nicknames or diminutives.

As a result, we contribute with: (i) an evaluation of the quality of different solutions applying well-known metrics;

(ii) a tool, called `damegender`, guessing gender from a name giving support to Spanish and English from the open data census provides by the states built to understand current technologies in detail; this tool has been compared with APIs using an international dataset with good results; and

(iii) a machine learning solution to strings not found in the census dataset to approach the problem with nicknames and diminutives;

The remainder of this paper is structured as follows:

In Section 2 we explain `damegender` as solution to the problem of name-to-gender inference. Section 3 introduces the Open Datasets used as the *golden truth*. Section 4 presents a feature comparison with other name-to-gender inference services and tools. Section 5 reports on values of accuracy and offers a confusion matrix using a scientific dataset. Section 6 is about how we use Machine Learning in `damegender`. Section 7 discusses limitations and further research, and concludes the paper.

2 Damegender

`damegender`[1] is a gender detection tool released under a free software license (in particular, the GNU General Public License v3.0). It has been implemented in Python to take advantage of many other free software tools used in the scientific domain, such as the Natural Language Toolkit (NLTK) for natural language processing [LB02], Scikit for machine learning [PVG+11], Numpy nor numerical computation [VDWCV11], and Matplotlib to visualize results [Hun07]. At its current point it is linked to Perceval [DCRGB18], a tool specialized in retrieving and gathering data from software repositories, such as git and mailing lists. `damegender` is a Python package that can be easily installed using PIP (the package installer for Python) from the console.

The main reason for developing `damegender` is that there are not many free software tools that help in the identification of gender. Before `damegender`, only `Gender guesser`[2] offered a free software solution in this field [Kra06], and the project (i) has not been active for more than three years now, (ii) lies technologically well behind other solutions. The best contribution of `Gender guesser` is the dataset containing 48,528 names with a good classification by countries[3].

---

[1] https://github.com/davidam/damegender
[2] https://github.com/lead-ratings/gender-guesser
[3] https://raw.githubusercontent.com/lead-ratings/gender-guesser/master/gender_guesser/data/nam_dict.txt

3 Datasets

Name-to-gender inference services and tools apply several methods for estimating the gender from a given name. As a starting point, however, all of them rely on a dataset that contains information on what gender a name usually can be attributed to.

There are several sources to create these databases, being the most common: (1) a census published with a free license (open census way), (2) a dataset released with a free license in a free software package (free software way), (3) a dataset retrieved from commercial APIs (commercial API way), and (4) a dataset which is the result of an investigation and that has been released publicly (scientific way).

In `damegender`, we are including Open Data census about names and gender, usually offered by governments, such as the ones from Spain (by means of INE.es, the Spanish National Statistics Institute), Uruguay, USA and UK.

Some Open Datasets, such the one offered by INE.es or the government of the US offer support for surnames and how they are related to ethnicity. In particular, the dataset from the government of the US offers a probability of the race, and the Spanish INE.es gives the number of people with a surname with a nationality different to the Spanish nationality.

Generally, a name has a strong weight to determine if it is a male or a female using this procedure. For instance, David is registered 365,196 times as male, but 0 times as female in the data offered by the Spain National Institute of Statistics. There are names that heavily depend on the region. For instance, Andrea would be considered a female name in Germany, but a male name in Italy. Hence, we are using the census approach as base of truth to distinguish if a name is male or female in a geographical area. However, many countries do not provide Open Data census about gender and names, although our hope is that they will do in the near future.

We have evaluated to include data from the second option (datasets released with a free license). For instance, the Natural Language Tool Kit offers 8,000 labeled English names classified as male or female. Another example is `Gender Guesser`, which provides a good dataset for international names with different categories to define the probability. Although these datasets can be incorporated into `damegender`, the problem with them is that, in general, we have observed that they do not have the quality of National Statistics Institutes.

The third approach is to create the dataset from existing, usually commercial solutions. Here we have to trust their results, in the same way we trust search engines when we make searches in Internet. This is be-

Service / Tool − >	Gender API	gender-guesser	genderize.io	NameAPI	NamSor	damegender
Database size	431M	45K	114M	1M	4G	57K
Regular data updates	Yes	no	No	Yes	Yes	Yes
Unstructured full name strings	Yes	No	No	Yes	No	Yes
Surnames	Yes	No	No	Yes	Yes	Yes
Non-Latin alphabets	Partial	No	Partial	Yes	Yes	No
Implicit geo-localization	Yes	No	No	Yes	Yes	No
Exists locale	Yes	Yes	Yes	Yes	Yes	Yes
Assignment type	P	B	P	P	P	P
Free parameters	T,P	G	P,C	T	S	T,C
Prediction	No	No	No	No	No	Yes
Free license	No	Yes	No	No	No	Yes
REST API	Yes	No	Yes	Yes	Yes	Planned
Limits number of requests	Yes (200)	∞	Yes	Yes	Yes	∞
Subscription (100K requests/month)	79	0	7	150	80	0

Table 1: Comparison of the different features that name-to-gender inference services and tools offer. Assignment type = {P: Probabilistic; B: Binary}. Free Parameters = {T: total_names; P: probability; C: count; G: gender; T: trust; S: scale }. The subscription price is given in euro.

cause commercial APIs can be seen as a black box – we do not know where the data comes from and how it has been treated. But, at least, we can measure its quality, as we will do next. So, we will see that –at this point– commercial APIs offer better results than other solutions. That is why `damegender` gives the possibility to include data from them. It is possible to download JSON files from the main commercial name-to-gender inference API solutions (e.g., `genderapi`, `genderize`, `namsor`, `nameapi`) and use it as the dataset. There are certain uses that are currently only available in such tools.

As a final goal, we envision to build a free dataset with names and gender, that builds on top of `Gender Guesser` and that can be made available as Wikidata with a free license. Perhaps, to complete this work, we need to combine an automated with a manual process as described in [SM18].

4 Feature comparison with other tools

Standard commercial name-to-gender inference APIs usually guess the gender for a single name or a list of names (from a CSV file or an API call). To express geolocalization the user can also give surnames, a country ISO code, or specify a language. Generally, you can give a probability and a counter associated to a name and gender in a certain population.

Santamaría and Mihaljević [SM18] offer a framework to classify gender tools. The features observed in this framework are: (i) database size (as of January 2018), (ii) if there are regular data updates, (iii) if they handle unstructured full name strings, (iv) if they handle surnames, (v) if they handle non-Latin alphabets, (vi) if implicit geolocalization is available, (vii) if locale exists, (viii) the type of assignment, (ix) if free parameters are possible, (x) if they offer prediction, (xi) if the tool is released under an open source license, (xii) if they offer an API, (xiii) the amount of

API	Acc	Prec	F1	Recall
Genderapi	0.969	0.972	0.964	1.0
Genderize	0.927	0.976	0.966	1.0
Damegender (SVC)[4]	0.879	0.972	0.972	1.0
Namsor	0.867	0.973	0.924	1.0
Nameapi	0.830	0.974	0.905	1.0
Gender Guesser	0.774	0.985	0.872	1.0

Table 2: Comparison of measures of the quality of the results for the tools under study.

monthly free requests, and (xiv) the monthly subscription cost (calculated for 100,000 requests/month)).

We have used this comparison framework and have updated it to the current situation and extended it with `damegender`. Results can be found in Table 1. As it can be observed, commercial services have the advantage of owning large datasets, in the order of millions or even of billions of entries. The most advanced ones offer support for non-Latin alphabets. However, they limit the amount of free requests that can be performed and their costs range from 7 euro to 150 euro per month.

5 Reproducing values of accuracy and confusion matrix

In this section, we will analyze the quality of the service of existing name-to-gender inference services and tools. There are different ways to express the probability of a successful identification (e.g., confidence, scale, accuracy, precision, and recall). We can also inspect the confusion matrix to understand where the tools and services succeed or fail, and to analyze the different errors measures (error coded, error coded without not applicable values, error gender bias, not applicable coded) that can happen.

Santamaría and Mihaljević [SM18] explain different ways to determine gender from a name; they offer 7,000 names that can serve as the *golden set* to evaluate them. In their dataset, gender is classified as male,

APIs	gender	male	female	undef
Genderapi	male	3589	155	67
	female	211	1734	23
Damegender	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 3: Confusion matrix tables by APIs

female, or unknown. We have used this dataset, not considering the unknown variable, for our experiments.

The results, using common information retrieval metrics, can be seen in Table 2. Accuracy is the ratio of correctly predicted observation to the total observations. It should be noted that *Accuracy* can be a misleading metric for imbalanced data sets, such as the ones that we usually have in software development projects [GGKR02]. This is because for a sample with 85 negative and 15 positive values, classifying all values as negative would give an accuracy score of 0.85. In those cases, it is better to report other measures, such as the balanced accuracy, which normalizes true positive and true negative predictions by the number of positive and negative samples [Mow05]. As Santamaría and Mihaljević's data does not come from the IT domain, we can use accuracy in our research. Precision is the fraction of relevant instances among the retrieved instances. Recall is the fraction of the total amount of relevant instances that were actually retrieved. In our case, we have left no name out, so recall is 1 for all tools. Precision and recall are sometimes used together in the F1 Score (or f-measure) to provide a single measurement for a system. As can be observed, `Genderapi` and `Genderize` obtain the best results – although all solutions are close and reach results better than 0.8 for accuracy, except for `Gender Guesser`. `damegender` offers relatively good results. Given, however, that its focus is mainly to be used in the IT domain, where the gender is highly imbalanced, we are very positive about its precision, as accuracy is not that relevant as noted above.

We have performed a comparison using a confusion matrix for the software/tools under study (see Table 3). These results are very similar to the ones reported in [SM18], which offers confidence that we have done our comparison correctly. The most important tools (`Namsor`, `Genderapi`, and `Genderize`) improve the values of accuracy with respect to the previous comparison. In particular, `Genderapi` improves the results for *undefined*. For `Genderize`, we obtain exactly the same results. For `Genderguesser`, however,

we obtain different results, which is to some extent not expected, because the software has not been modified for several years. `Nameapi`'s results is changing from male to female with more errors. In `Namsor`, the results are similar. `damegender` is not guessing undefined because we predict with machine learning (using an SVC algorithm) if the string is not in the database.

In Table 4 we can see the different measures for errors in the APIs. *Error coded* defines if the true is different than the guessed one. *Error coded without na* defines if the true is different than the guessed one, but without undefined results. *Error gender bias* allows to understand if the error is bigger for guessing males than females or viceversa. The *weighted error* defines if the true value is different than the guessed one, but giving a weight to the guessed as undefined. The most relevant information is a high index of errors for `Nameapi` and `Namsor`, while `GenderApi` and `damegender` have a low index of errors.

6 Machine Learning

6.1 Comparing ML algorithms

Table 5 shows the accuracy measures for some Machine Learning algorithms used in our guessing. The best results are given by Support Vector Machines and Random Forest – with those algorithms `damegender` achieves values that are close to more mature, proprietary solutions. It should be noted that as a result of using machine learning techniques, our classifier is binary (male and female), so no *unknown* is given as output.

6.2 Experimenting with some features

We have developed some additional experimental functionality that allows to analyze our database according to some features using machine learning algorithms. To test our approach, we have selected some features of names, such as a being the first letter, a (or o) being the last letter, contains the letter a, first letter is a vowel, last letter is a vowel, last letter is a consonant, or last letter is a. The selection of the features was verified with Principal Component Analysis. The datasets used in this experiment were the ones from the National Institute of Statistics (Spain, Uruguay, United Kingdom, United States of America, Australia and Canada). The most relevant results for the different datasets used are offered in Table 6.

As expected, countries that share language offer similar results, i.e., the variation of the chosen features between males and females is comparable. This is the case for Uruguay and Spain (Spanish), and USA and UK and Australia (English). In Canada, a country

API	error	error w/o 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 4: APIs and Errors

ML Algorithm	Acc	Prec	F1	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 Naive Bayes	0.782	0.791	0.791	1.0
Tree	0.764	0.821	0.796	1.0
Stoch. Gradient Distrib.	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 accuracy measures

that has an ample French-speaking community these features show a different trend.

For instance, the feature *containing the letter a in the name* varies 0.2 from males to females for USA and Uruguay, and 0.1 from males to females for United Kingdom, Australia, and Spain. The feature *the last letter is an a* varies 0.5 from males to females for Australia and Spain, around 0.4 for USA and United Kingdom and 0.2 in Uruguay. The feature *the last letter is an o* varies 0.2 in (Spain, Australia) from females to males and is equal in Uruguay, USA, United Kingdom. For the feature *the last letter is a consonant* all countries give as a result that males do have it more frequently, with results that range from 0.3 to 0.5: Uruguay and USA (0.5), United Kingdom (0.4), Australia and Spain (0.3). So, the feature *the last letter is a vowel* is, as expected, the reverse of the feature *the last letter is a consonant*. The features *the first letter is a consonant* or *the first letter is a vowel* are, however, non-significant as they offer similar results in English and Spanish.

We have done this experiment with the NLTK and INE.es datasets, with the values of accuracy reaching up to 0.745. So it makes sense to expect better results in random datasets if we add new languages and countries. However, our solution is not providing Arabic or Chinese alphabets, yet. The results of this experiment could be used to provide a good solution for nicknames, diminutives, or similar.

7 Limitations and further research

The market of gender detection tools and services is currently dominated by companies based on payment services through Application Programming Interfaces. Without doubt, they offer good results, with high accuracy values. However, their inner working cannot be studied (i.e., they work as a black-box for the outsider) and the fees that have to be paid for using their service are sometimes out of the reach of many researchers. That is why we propose a new tool, called `damegender`, with the aim of having open data of name-to-gender inference, which offers more flexibility and where researchers can build on top of it. This tool is offered under a free software license and is available on GitHub for download and enhancement. As we have shown in this paper, although still incipient, the tool offers good accuracy values based on the use of public databases from government bodies and on the use of machine learning algorithms. Nonetheless, we have to note that `damegender` is still under development, and that it has to be applied to several real repositories to confirm its benefits and address its limitations (such as a small database size of gendered names).

In addition, we have shown a glimpse of how several features of the names could be used to guess the gender if we do not have the real name, but nicknames or diminutives, a situation that is very common in free software development-supporting tools, such as IRC channels. This experiment is at this point very preliminary, and we would like to work more on it.

All in all, we hope the `damegender` tool can become a cornerstone for the scientific advancement of the study of gender (including gender gap) and diversity in the IT, and in particular in the free software community. We have therefore many hopes in linking the output of repositories like the ones that can be fetched by Perceval (git, mbox mailing lists, Gerrit, Bugzilla, etc.). As a result, we envision a free and universal dataset with support for all languages and cultures.

Acknowledgments

Dataset	contains a	last is a	last is o	last is consonant	last is vowel	1st is consonant	1st is vowel
Uruguay (F)	0.816	0.456	0.007	0.287	0.712	0.823	0.177
Uruguay (M)	0.643	0.249	0.062	0.766	0.234	0.771	0.228
Spain (F)	0.922	0.588	0.030	0.271	0.728	0.772	0.228
Spain (M)	0.818	0.030	0.268	0.569	0.430	0.763	0.236
UK (F)	0.825	0.374	0.013	0.322	0.674	0.765	0.235
UK (M)	0.716	0.036	0.039	0.780	0.218	0.799	0.200
USA (F)	0.816	0.456	0.007	0.287	0.712	0.823	0.177
USA (M)	0.643	0.020	0.061	0.765	0.234	0.840	0.159
Canada (F)	0.659	0.189	0.005	0.591	0.408	0.838	0.160
Canada (M)	0.752	0.220	0.025	0.540	0.456	0.818	0.181
Australia (F)	0.922	0.588	0.033	0.272	0.728	0.772	0.228
Australia (M)	0.818	0.030	0.269	0.570	0.430	0.763	0.237

Table 6: Informative features for different countries. F stands for females, and M for males.

References

[AYCN11]   Judd Antin, Raymond Yee, Coye Cheshire, and Oded Nov. Gender differences in wikipedia editing. In *Proceedings of the 7th international symposium on wikis and open collaboration*, pages 11–14. ACM, 2011.

[BHKZ11]   John D Burger, John Henderson, George Kim, and Guido Zarrella. Discriminating gender on twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1301–1309. Association for Computational Linguistics, 2011.

[CPT+19]   Gemma Catolino, Fabio Palomba, Damian A Tamburri, Alexander Serebrenik, and Filomena Ferrucci. Gender diversity and community smells: insights from the trenches. *IEEE Software*, 37(1):10–16, 2019.

[DCRGB18]   Santiago Dueñas, Valerio Cosentino, Gregorio Robles, and Jesus M Gonzalez-Barahona. Perceval: Software project data at your will. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, pages 1–4. ACM, 2018.

[DG99]   David Dollar and Roberta Gatti. *Gender inequality, income, and growth: are good times good for women?*, volume 1. Development Research Group, The World Bank Washington, DC, 1999.

[GGKR02]   Rishab A Ghosh, Ruediger Glott, Bernhard Krieger, and Gregorio Robles. Free/libre and open source software: Survey and study, 2002.

[HS13]   Benjamin Mako Hill and Aaron Shaw. The wikipedia gender gap revisited: Characterizing survey response bias with propensity score estimation. *PloS one*, 8(6):e65782, 2013.

[HSFH18]   Luke Holman, Devi Stuart-Fox, and Cindy E Hauser. The gender gap in science: How long until women are equally represented? *PLoS biology*, 16(4):e2004956, 2018.

[Hun07]   John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90, 2007.

[IHSR18]   Daniel Izquierdo, Nicole Huesman, Alexander Serebrenik, and Gregorio Robles. Openstack gender diversity report. *IEEE Software*, 36(1):28–33, 2018.

[KAS02]   Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. Automatically categorizing written texts by author gender. *Literary and linguistic computing*, 17(4):401–412, 2002.

[Kra06]   N Krawetz. Gender guesser, 2006.

[LB02]   Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[LC19]   Amanda Lee and Jeffrey C Carver. Floss participants' perceptions about gender and inclusiveness: a survey. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 677–687. IEEE, 2019.

[LSB11]   Marcus Liwicki, Andreas Schlapbach, and Horst Bunke. Automatic gender detection using on-line and off-line information. *Pattern Analysis and Applications*, 14(1):87–92, 2011.

[MLA+11] Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J Niels Rosenquist. Understanding the demographics of twitter users. In *Fifth international AAAI conference on weblogs and social media*, 2011.

[Mow05] Jeffrey P Mower. Prep-mt: predictive rna editor for plant mitochondrial genes. *BMC bioinformatics*, 6(1):96, 2005.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[RARS+14] Gregorio Robles, Laura Arjona Reina, Alexander Serebrenik, Bogdan Vasilescu, and Jesús M González-Barahona. 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, 2014.

[RPC17] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. 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, 2017.

[SM18] Lucía Santamaría and Helena Mihaljević. Comparison and benchmark of name-to-gender inference services. *PeerJ Computer Science*, 4:e156, July 2018.

[VCS12] Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik. Gender, representation and online participation: A quantitative study of stackoverflow. In *2012 International Conference on Social Informatics*, pages 332–338. IEEE, 2012.

[VDWCV11] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22, 2011.

[VPR+15] Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark GJ van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov. 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, 2015.