OpenGender: Open source gender classification for large scale data analytics

Mingi Ryu

1 Abstract

Despite abundance of commercially available gender inferences services, large scale quantitative analysis of gender bias and inequality benefits greatly from free and open source gender inference tools. Particularly in tech, media, and academia, the large scale of such analysis requires a significant amount of resources to process datasets consisting of millions of names. Recent interest in open source alternatives and the availability of large datasets of names and gender. This paper presents an open source gender classifier based on several machine learning models and gender datasets to achieve comparable performance compared to commercial gender inference APIs.

2 Introduction

2.1 Large-scale gender analysis

While there are many online application programming interfaces (APIs) offering both free and paid options, most are unsuitable for large scale data analytics of more than 10 million data points. The use of gender inference on individuals requires high confidence in the services or the model’s performance. In large scale analysis, this becomes less relevant. Majority of the people have either distinctly feminine or masculine names. Thus, we can analyze the gender bias and imbalance in a large dataset with higher confidence.

2.2 Free and paid APIs

Most if not all publicly available gender inference APIs offer some form of free plan where a limited amount of API requests are provided for free of charge. For example, Gender API provides 100 credits per month[1] which corresponds to mere 100 inferences per month. Genderize.io provides 1,000 names per day[2], roughly 30,000 names per month.

While paid plans offer higher limits, none are suitable for processing millions of names in a single dataset job, which could take from an hour to at most a day. This severely limits the availability to conduct large scale data analytics of gender inequality and bias.

2.3 Open source libraries and tools

On the other hand, open source libraries such as gender-guesser[3] can provide a way for a large-scale gender inference. It can infer thousands of names under a millisecond on a modern CPU since it does not need to make any network calls. However, it is limited to a known set of 45,000 names and underperforms severely[4] compared to publicly available APIs.

While not a direct substitute, open source tools such as Damegender can provide a way to infer gender at scale. It provides various gender inference datasets from more than 20 countries and an international dataset that contains more than 400,000 names. Unfortunately, it does not provide a simple way to use the Python package and the suggested inference pipeline is quite difficult to work in large scale settings.

[1] Based on https://gender-api.com/en/pricing as of November 2022.
[2] Based on https://genderize.io/#rate-limiting as of November 2022.
[3] Available as a Python package on PyPI https://pypi.org/project/gender-guesser/
[4] While it achieves the lowest misclassification rate, it also achieves the highest non-classification compared to publicly available gender inference tools and APIs [Santamaría, 2018]

3 Related works

3.1 Open source datasets

Due to the abundance of publicly available records of names and assigned gender, there has been numerous efforts in collecting gender and name data from various sources such as United States Social Security, Wikipedia, etc. In [Bérubé, N., et al., 2020], the paper presents a free dataset of 130,000 first names and gender collected from Wikipedia. Using the dataset, the paper proposes the Wiki-Gendersort algorithm that assigns a gender based on names that appear in Wikipedia.

In the most recent work of [Menendez, David et al., 2022], the paper presents a collection of names and gender datasets covering more than 20 countries to address the lack of free and open source tools for analyzing gender imbalance. The paper focuses on the free and open source aspect of gender inference whereas prior works have been focused on comparing existing free and paid API services that offer gender identification based on first name or full name.

3.2 Comparison of gender inference tools and APIs

As of 2022, there are more than five commercial gender inference services in the market that offer essentially the same product. Due to the nature of such services, there have been many various analyses of these services. In [Santamaría, 2018], five commercially available gender inference tools and APIs are evaluated against a test dataset of 7,706 gold labels. In [Menendez, David et al., 2020], six of nearly identical services are evaluated against the same dataset.

While commercial gender inference services offer the best performance compared to existing tools, they can be quite limiting and expensive for large scale data analysis. [Mueller, 2016] proposes use of machine learning models based on hand selected features for analyzing names from Twitter achieving comparable performance. In [Menendez, David et al., 2020], a large set of machine learning algorithms are evaluated and compared against commercial services.

4 Research and methods

4.1 Task description

Give a list of first names with gender identification, find a model that is able to assign gender labels to each name using only the name itself.

4.2 Performance metrics

Since gender inference is a binary classification problem with one or more extra output classes such as “unknown”, errorCoded, errorCodedWithoutNA, and naCoded are used instead of precision and recall [Santamaría, 2018] [Wais, 2016]. These metrics are derived based on the confusion matrix to better incorporate non-classification into the metrics.

		predicted class: male	predicted class: female	predicted class: unknown
true class	male	m_m	m_f	m_u
true class	female	f_m	f_f	f_u

For errorCoded, non-classification is treated as a misclassification. For errorCodedWithoutNA, non-classification is ignored. For naCoded, the proportion of non-classifications over the total number of assignments are computed.

$$\begin{aligned}
\text{errorCoded} &= \frac{f_m+m_f+m_u+f_u}{m_m+f_m+m_f+f_f+m_u+f_u}, \\
\text{errorCodedWithoutNA} &= \frac{f_m+m_f}{m_m+f_m+m_f+f_f}, \\
\text{naCoded} &= \frac{m_u+f_u}{m_m+f_m+m_f+f_f+m_u+f_u}, \\
\text{errorGenderBias} &= \frac{m_f-f_m}{m_m+f_m+m_f+f_f}.
\end{aligned}$$

Another important metric is errorGenderBias. It estimates whether there are more females misclassified as male, or vice versa. Ifit is positive, then it is biased towards women such that it predicts more women than in the real data. While it is most desirable to consider all of the

aforementioned performance metrics, some are more important than others. In [Santamaría, 2018], weightedError is introduced as one of the extensions of the metrics.

$$\text{weightedError}_w = \frac{\text{f}_\text{m} + \text{m}_\text{f} + w * (\text{m}_\text{u} + \text{f}_\text{u})}{\text{m}_\text{m} + \text{f}_\text{m} + \text{m}_\text{f} + \text{f}_\text{f} + w * (\text{m}_\text{u} + \text{f}_\text{u})}.$$

4.3 State of the art

In [Santamaría, 2018], five gender inference services are benchmarked with a test dataset consisting of four different datasets[5].

Table 1: Performance metrics from [Santamaría, 2018]

	errorCoded	errorCodedWithoutNA	errorGenderBias	naCoded
Gender API	0.0789	0.0503	−0.0111	0.0301
gender-guesser	0.2224	0.0264	0.0022	0.2012
genderize.io	0.1428	0.0502	0.0222	0.0974
NameAPI	0.1794	0.0342	0.0037	0.1504

Based on [Table 1], Gender API achieves the lowest errorCoded of 0.0789, making it the best performing gender inference service. While gender-guesser achieves the lowest errorrCodedWithoutNA of 0.0264 and the lowest errorGenderBias of 0.0022, it achieves the highest errorrCodedWithoutNA due to high non-classification.

In [Sebo, 2021], four different gender inference services are compared with a test dataset that contains four physician datasets. As mentioned above, errorCoded, errorCodedWithoutNA, naCoded, and errorGenderBias performance metrics are used to evaluate the performance of the services.

Table 2: Performance metrics from [Sebo, 2021]

[5] The data sources are zbMATH, genderizeR, PubMed, and WoS. Refer to [Santamaría, 2018] for more details about each data source.

	errorCoded	errorCodedWithoutNA	naCoded	errorGenderBias
Gender API	0.0276	0.0211	0.0066	-0.0117
NamSor	0.0305	0.0305	0.0000	0.0013
Wiki-Gendersort	0.0959	0.0299	0.0680	-0.0086
genderize.io	0.2815	0.0243	0.2635	-0.0099

Based on [Table 2], Gender API still performs the best when compared to other gender inference services. Similar to gender-guesser, Wiki-Gendersort suffers from non-classification despite having comparable misclassification.

In [Menendez, David et al., 2020], the test dataset from [Santamaría, 2018] is used to evaluate its Damegender model and other gender inference services. Unfortunately, the result from the paper is heavily skewed towards Damegender and the model seems to have been trained on the entire test set. In an attempt to reproduce the result, the performance metrics were recalculated based on the provided confusion matrix.

Table 3: Performance metrics from [Menendez, David et al., 2020][6]

	errorCoded	errorCodedWithoutNA	errorGenderBias	naCoded
Damegender	0.1192	0.1192	-0.0690	0.0
Gender API	0.0789	0.0643	-0.0098	0.0156
genderize.io	0.1522	0.0608	0.03200	0.0974
NameAPI	0.3615	0.2667	0.00139	0.1293

According to the recalculated metrics, the result is consistent with other papers for the most part. Gender API outperforms other services by large margin, which is consistent with the results from

[6] In [Menendez, David et al., 2020], the performance metrics (Table 4) presents a widely incorrect result that favors Damegender. In this paper, we recalculated the metrics from confusion matrix result (Table 3) in [Menendez, David et al., 2020] to obtain more sensible results. Furthermore, Genderguesser and Namesor (v1) were omitted as they had nearly identical confusion matrices.

[Santamaría, 2018]. However, the performance metrics for the Damegender are a bit hard to evaluate due to the lack of “unknown” class in its predictions.

4.4 Feature engineering

For manual feature extraction, there are many ways to encode a person’s name. For example, [Mueller, 2016] introduced a number of characteristics that were identified in prior onomastic research and works. The paper identified that the ending character was the strongest predictor of gender identification where the majority of feminine names ended with a vowel and the majority of mascinline names ended with a consonant.

Table 4: Set of characters that can be extracted from a name. [Mueller, 2016]

Number of syllables	Female names tend to have more syllables than their male counterparts.
Number of consonants	Male names tend to contain more consonants than female names.
Number of vowels	Female first name tend to contain more vowels than male names.
Vowel brightness	Female names contain more brightly emphasized vowels than male names.
Ending character	Female names end more often with a vowel while male names tend to end with a consonant
Number of Bouba consonants	Female name can be identified by counting the voiced consonants “b”, “l”, “m”, and “n”.
Number of Bouba vowels	Female name can be identified by counting the rounded vowels “u”, “o”, and “6”.
Number of Kiki consonants	Male name can be identified by counting the voiceless stop consonants “k”, “p”, and “t”.
Number of Kiki vowels	Male name can be identified by counting the unrounded vowels “i”, “e”, “ɛ”, and “2”.

While the majority of these features can be extracted from the written name alone, some of the features identified required the pronunciation of the name. Since pronunciation of a name depends on spoken languages, it is highly sensitive to which country the name is from and the pronunciation of the region.

In [Mahmood, 2019], Term Frequency-Inverse Document Frequency (TF-IDF) was used to automatically extract features from tweet text. TF-IDF is widely used in many NLP tasks for its ease of use and performance. Since a tweet text consists of a higher number of words compared to a person's first name, it can utilize word level features to extract unique words in a document. However, the majority of the first names are a single word name and a very few have more than two words. Character level TF-IDF can be used in place of word level TF-IDF, but the limited number of unique characters and the lack of positional encoding of a certain set of characters makes it challenging to utilize TF-IDF in the context of gender identification.

4.5 Models

Since our task is limited to a few tokens per input mostly consisting of a unique set of characters, simple ML algorithms are sufficient. In [Menendez, David et al., 2020], eight different machine learning algorithms are evaluated against the test set from [Santamaría, 2018].

Table 5: Comparison of different ML algorithms [Menendez, David et al., 2020].

	Accuracy	Precision	F1	Recall
Support Vector Machine (SVM)	0.879	0.972	0.972	1.0
Random Forest (RF)	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
Stochastic Gradient Descent	0.709	0.943	0.815	1.0
Guassian Naive Bayes	0.709	0.968	0.887	1.0

Bernoulli Naive Bayes	0.699	0.965	0.815	1.0

According to [Table 5], SVM, RF, and NLTK archives F1 score above 0.9. However, the discrepancies in the aforementioned results regarding performance metrics [Table 3] and suspiciously identical performance between RF and NLTK models calls further investigation into the models.

5 Materials and data sources

5.1 Damegender international dataset

The Damegender [Menendez, David et al., 2022] dataset consists of various CSV and JSON files of names and gender from more than 20 countries. The canonical dataset, interall.csv, contains 447,055 rows of first name, frequency, and gender probabilities. Since the dataset provides probabilities for either male or female, the labels are male or female depending on which one has a higher probability.

5.2 Name gender inference dataset

The name gender inference [Santamaría, 2018] dataset consists of a single test dataset derived from multiple data sources such as zbMath, genderizeR, PubMed, and WoS. It contains 7,706 first names, last names, full names, and gender assignment for each name. The labels are m (male), f (female), and u (unknown).

5.3 Wiki Gendersort dataset

The Wiki Gendersort [Mueller, 2016] dataset contains the gender association to the 694,376 names from Wikipedia. The labels are M (male), F (female), UNI (unisex), UNK (unknown) and INI (initials).

6 Experiments and results

6.1 Feature extraction across different ML models and datasets

To identify the optimal feature extraction method, onomastic features [Mueller, 2016] and term frequency features[7] are evaluated across multiple models[8] and datasets[9].

Table 6: errorCoded performance metric for different feature extraction methods

	Damegender	Wiki_Gendersort	Name gender inference
Onomastic (SVM)	0.403	1.0	0.280
Onomastic (RF)	**0.213**	0.987	**0.168**
Count (SVM)	0.262	0.992	0.226
Count (RF)	0.291	0.945	0.244
TF-IDF (SVM)	0.268	0.997	0.227
TF-IDF (RF)	0.295	**0.926**	0.256

For both Damegender and Name gender inference datasets in Table 6, Random Forest model with onomastic features outperformed other combinations by a significant margin. Due to significant imbalance in the Wiki_Gendersort dataset where “unknown” gender is greater than 50% of the training dataset, many of the models failed to accurately predict “unknown” gender.

6.2 Cross validation and hyperparameters

In the previous experiment (6.1), only the default set of parameters were used for CountVectorizer and TifidfVectorizer. To further improve the performance of term frequency based features, cross validation with relevant parameters are used to pick the most optimal model. For cross validation and model selection, HalvingRandomSearchCV from Sklearn was used with default parameters (5-fold) except for the custom scoring function that incorporates

[7] CountVectorizer and TfidfVectorizer from sklearn.feature_extraction.text with character-level analyzer
[8] Support Vector Machine (SVM) and Random Forest (RF) are used as the default set of models.
[9] Sample size of 7,000 names and gender are split into train (80%) test (20%) split

errorCoded performance metric. Relevant parameters and additional options for the vectorizers are listed in the following table (Table 7).

Table 7: Relevant parameters for Sklearn vectorizers.

Relevant parameters	Defaults	Additional options for cross validation
stripe_accents	None	ascii, unicode
ngram_range	(1,1)	(1,1), (1,2), (1,3), (1,4)
max_df	1.0	0.5, 0.75, 0.9
min_df	1	0.1, 0.01, 0.001
max_features	None	10,000, 20,000, … , 100,000

Similar to the prior experiment (6.1), support vector machine and random forest models were trained and evaluated against three different datasets. For Wiki_Gendersort dataset, the "unknown" labels were omitted to prevent the models failing to learn due to label imbalance.

Table 8: errorCoded performance metric for different vectorizers w/ cross validation

	Damegender	Wiki_Gendersort[10]	Name gender inference
Count (SVM)	**0.214**	0.345	**0.163**
Count (RF)	0.253	0.359	0.198
TF-IDF (SVM)	0.217	**0.340**	0.168
TF-IDF (RF)	0.252	0.354	0.183

Based on Table 8, the SVM model with CountVectorizer achieves comparable performance with respect to the RF model with onomastic features in Table 6. However, neither models achieve the performance of commercially available services such as Gender API.

In both 6.1 and 6.2 experiments, a sample of 7,000 labels in order to maintain a fair comparison with respect to the name gender inference dataset. However, both the Damegender and the

[10] "unknown" labels are omitted.

Wiki_gendersort datasets consist of more than 400,000 labels and almost 700,000 labels respectively. In order to gauge the maximum training performance of each model, this experiment utilizes the entire dataset instead of training only on the 80% of 7,000 labels. Similar to previous experiments, each dataset is split into 80% training set and 20% test set. Since training on the entire dataset takes a much longer time, the models are trained without cross validation with the following fixed parameters[11] for both the CountVectorizer and TfidfVectorizer:

- ngram_range=(1, 3)
- max_df=0.75
- min_df=0.01
- max_features=50_000
- strip_accents='ascii'

Table 8: errorCoded performance metric for entire datasets without sampling

	Damegender	Wiki_Gendersort[12]
Onomastic (SVM)	0.414	0.350
Onomastic (RF)	0.157	0.313
Count (SVM)	0.152	0.271
Count (RF)	0.160	0.287
TF-IDF (SVM)	**0.144**	**0.268**
TF-IDF (RF)	0.171	0.301

In Table 8, both of the datasets demonstrate that training on the entire dataset improves the performance across every variation of the models. Surprisingly, models with onomastic features performed worse than models with term frequency features. While onomastic features can account for well known instances of name characteristics, term frequency features can accommodate for the entire dataset with wider distribution of characteristics that are difficult to identify manually.

[11] Best set of parameters identified in 6.2 experiments.
[12] Similar to 6.2, “unknown” labels are dropped from the training dataset.

7 Conclusion

Abundance of commercial gender inference services, availability of large scale gender datasets, and relevant benchmarks greatly benefits the data analysis of gender inequality and bias in tech, media, and academia. While commercial APIs such as Gender API consistently achieves the best performance for gender identification, they are quite costly and limited in large scale quantitative analysis of academic publications, literatures, employment, etc. Using open source names and gender datasets such as Damegender and Wiki_Gendersort, machine learning models with various features achieves comparable performance based on a performance metric that incorporates both misclassification and non-classification.

References

[Bérubé, N., et al., 2020] Bérubé, Nicolas, et al. “Wiki-gendersort: Automatic Gender Detection Using First Names in Wikipedia.” SocArXiv, 14 Mar. 2020. Web.

[Santamaría, 2018] Santamaría, Lucía and Helena Mihaljevic. “Comparison and benchmark of name-to-gender inference services.” *PeerJ Computer Science* 4 (2018): n. Pag.

[Sebo, 2021] Sebo, Paul. “Performance of gender detection tools: a comparative study of name-to-gender inference services.” *Journal of the Medical Library Association : JMLA* 109 (2021): 414 - 421.

[Mahmood, 2019] Mahmood, Asad and Padmini Srinivasan. “Twitter Bots and Gender Detection using Tf-idf.” *CLEF* (2019).

[Menendez, David et al., 2020] Menendez, David et al. “Damegender: Writing and Comparing Gender Detection Tools.” *SATToSE* (2020).

[Menendez, David et al., 2022] Menendez, David et al. “Damegender: Towards an International and Free Dataset about Name, Gender and Frequency” *EasyChair Preprint no. 5763* (2022).

[Mueller, 2016] Mueller, Juergen and Gerd Stumme. “Gender Inference using Statistical Name Characteristics in Twitter.” *Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016* (2016): n. Pag.

[Wais, 2016] Wais, Kamil. “Gender Prediction Methods Based on First Names with genderizeR.” *R J.* 8 (2016): 17.