Demographer: Extremely Simple Name Demographics

Rebecca Knowles
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
rknowles@jhu.edu

Josh Carroll
Qntfy
Crownsville, MD 21032
josh@qntfy.com

Mark Dredze
Human Language Technology
Center of Excellence
Johns Hopkins University
Baltimore, MD 21211
mdredze@cs.jhu.edu

Abstract

The lack of demographic information available when conducting passive analysis of social media content can make it difficult to compare results to traditional survey results. We present DEMOGRAPHER,[1] a tool that predicts gender from names, using name lists and a classifier with simple character-level features. By relying only on a name, our tool can make predictions even without extensive user-authored content. We compare DEMOGRAPHER to other available tools and discuss differences in performance. In particular, we show that DEMOGRAPHER performs well on Twitter data, making it useful for simple and rapid social media demographic inference.

1 Introduction

To study the attitudes and behaviors of a population, social science research often relies on surveys. Due to a variety of factors, including cost, speed, and coverage, many studies have turned to new sources of survey data over traditional methods like phone or in-person interviews. These include web-based data sources, such as internet surveys or panels, as well as *passive analysis of social media content*. The latter is particularly attractive since it does not require active recruitment or engagement of a survey population. Rather, it builds on data that can be collected from social media platforms.

Many major social media platforms, such as Twitter, lack demographic and location characteristics available for traditional surveys. The lack of these data prevents comparisons to traditional survey results. There have been a number of attempts to *automatically* infer user attributes from available social media data, such as a collection of messages for a user. These efforts have led to author attribute, or demographic, inference (Mislove et al., 2011; Volkova et al., 2015b; Burger et al., 2011; Volkova et al., 2015a; Pennacchiotti and Popescu, 2011; Rao and Yarowsky, 2010; Rao et al., 2010; Schwartz et al., 2013; Ciot et al., 2013; Alowibdi et al., 2013; Culotta et al., 2015) and geolocation tasks (Eisenstein et al., 2010; Han et al., 2014; Rout et al., 2013; Compton et al., 2014; Cha et al., 2015; Jurgens et al., 2015; Rahimi et al., 2016).

A limitation of these content analysis methods is their reliance on multiple messages for each user (or, in the case of social network based methods, data about multiple followers or friends for each user of interest). For example, we may wish to better understand the demographics of users who tweet a particular hashtag. While having tens or hundreds of messages for each user can improve prediction accuracy, collecting more data for every user of interest may be prohibitive either in terms of API access, or in terms of the time required. In this vein, several papers have dealt with the task of geolocation from a single tweet, relying on the user's profile location, time, tweet content and other factors to make a decision (Osborne et al., 2014; Dredze et al., 2016). This includes tools like Carmen (Dredze et al., 2013) and TwoFishes.[2] For demographic prediction, several papers have explored using names to infer gender and ethnicity (Rao et al., 2011; Liu and Ruths,

[1] https://bitbucket.org/mdredze/demographer

[2] http://twofishes.net/

Proceedings of 2016 EMNLP Workshop on Natural Language Processing and Computational Social Science, pages 108–113,
Austin, TX, November 5, 2016.

2013; Bergsma et al., 2013; Chang et al., 2010), although there has not been an analysis of the efficacy of such tools using names alone on Twitter.

This paper surveys existing software tools for determining a user's gender based on their name. We compare these tools in terms of accuracy on annotated datasets and coverage of a random collection of tweets. Additionally, we introduce a new tool DEMOGRAPHER which makes predictions for gender based on names. Our goal is to provide a guide for researchers as to software tools are most effective for this setting. We describe DEMOGRAPHER and then provide comparisons to other tools.

2 Demographer

DEMOGRAPHER is a Python tool for predicting the gender[3] of a Twitter user based only on the name[4] of the user as provided in the profile. It is designed to be a lightweight and fast tool that gives accurate predictions when possible, and withholds predictions otherwise. DEMOGRAPHER relies on two underlying methods: name lists that associate names with genders, and a classifier that uses features of a name to make predictions. These can also be combined to produce a single prediction given a name.

The tool is modular so that new methods can be added, and the existing methods can be retrained given new data sources.

Not every first name (given name) is strongly associated with a gender, but many common names can identify gender with high accuracy. DEMOGRAPHER captures this through the use of name lists, which assign each first name to a single gender, or provide statistics on the gender breakdown for a name. Additionally, name morphology can indicate the gender of new or uncommon names (for example, names containing the string "anna" are often associated with *Female*). We use these ideas to implement the following methods for name classification.

Name list This predictor uses a given name list to build a mapping between name and gender. We assign scores for female and male based on what fraction of times that name was associated with females and males (respectively) in the name list. This model is limited by its data source; it makes no predictions for names not included in the name list. Other tools in our comparison also take this approach.

Classifier We extract features based on prefix and suffix of the name (up to character 4-grams, and including whether the first and final letters are vowels) and the entire name. We train a linear SVM with L2 regularization. For training, we assume names are associated with their most frequent gender. This model increases the coverage with a modest reduction in accuracy. When combined with a threshold (below which the model would make no prediction), this model has high precision but low recall.

3 Other Tools

For comparison, we evaluate four publicly available gender prediction tools. More detailed descriptions can be found at their respective webpages.

Gender.c We implement and test a Python version of the gender prediction tool described in Michael (2007), which uses a name list with both gender and country information. The original software is written in C and the name list contains 32,254 names and name popularity by country.

Gender Guesser Pérez (2016) uses the same data set as Gender.c, and performs quite similarly (in terms of accuracy and coverage).

Gender Detector Vanetta (2016) draws on US Social Security Administration data (which we also use for training DEMOGRAPHER), as well as data from other global sources, as collected by Open Gender Tracking's Global Name Data project.[5]

Genderize IO Strømgren (2016) resolves first names to gender based on information from user profiles from several social networks. The tool is accessed via a web API, and results include gender, probability, and confidence expressed as a count. According to the website, when we ran our experiments the tool included 216,286 distinct names from 79 countries and 89 languages. It provides limited free access and larger query volumes for a fee.

Localization Several tools include the option to provide a locale for a name to improve accuracy. For example, Jean is typically male in French and female

[3] We focus on gender as a social or cultural categorization.

[4] Note that we mean "name" and *not* "username."

[5] https://github.com/OpenGenderTracking/globalnamedata

in English. We excluded localization since locale is not universally available for all users. We leave it to future work to explore its impact on accuracy.

4 Data

4.1 Training Data

We train the classifier in DEMOGRAPHER and take as our name list Social Security data (Social Security Administration, 2016), which contains 68,357 unique names. The data is divided by year, with counts of the number of male and female children given each name in each year. Since it only includes names of Americans with Social Security records, it may not generalize internationally.

4.2 Evaluation Data

Wikidata We extracted 2,279,678 names with associated gender from Wikidata.[6] We use 100,000 for development, 100,000 for test, and reserve the rest for training in future work. While data for other genders is available on Wikidata, we selected only names that were associated with either *Male* or *Female*. This matches the labels available in the SSA data used for training, as well as the other gender prediction tools we compare against. This dataset is skewed heavily male (more than 4 names labeled male for every female), so we also report results on a balanced (subsampled) version.

Annotated Twitter These names are drawn from the "name" field from a subset of 58,046 still publicly tweeting users from the Burger et al. (2011) dataset (user IDs released with Volkova et al. (2013)). Of these, 30,364 are labeled *Female* and 27,682 are labeled *Male*. The gender labels are obtained by following links to Twitter users' blogger profile information (containing structured gender self-identification information).

Unannotated Twitter Since the annotated Twitter data contains predominantly English speakers (and who may not be representative of the general Twitter population who do not link to external websites), we also evaluate model coverage over a sample of Twitter data: the 1% feed from July 2016 from containing 655,963 tweets and 526,256 unique names.

[6] `https://www.wikidata.org/wiki/Wikidata:Database_download` We used the dump from 2016-08-15.

4.3 Processing

All data is lowercased for consistency. For the Twitter data, we use a regular expression to extract the first string of one or more (Latin) alphabetic characters from the name field, if one exists. This may or may not be the user's actual given name (or even a given name at all). Note that most of the tools are do not handle non-Latin scripts, which limits their usefulness in international settings.

5 Results

Table 1 reports results for Wikidata in terms of accuracy (percent of correctly predicted names only including cases where the tool made a prediction), coverage (the percent of the full test set for which the tool made a prediction), F1 (the harmonic mean of accuracy and coverage), and the number of names labeled per second. The corresponding result for the balanced version of the dataset is in parentheses.

Tools make different tradeoffs between accuracy, coverage, and speed. Both Gender.c and Gender Guesser have high accuracy and fairly high coverage at high speed (with Gender.c being the fastest of the tools evaluated). Gender Detector has slightly higher accuracy, but this comes at the cost of both coverage and speed (it is second slowest). Genderize IO has the best F1 among all name list based approaches, but stands out for lower accuracy and higher coverage. We show five settings of DEMOGRAPHER: name list only (fast, accurate, but with only fairly high coverage), classifier (slow, and either high coverage with no threshold or high accuracy with a high threshold) and the combined versions, which fall in between the name list and classifier in terms of speed, accuracy, and coverage). The combined demographer with no threshold performs best out of all tools in terms of F1.

Table 2 shows results on Twitter data. The *Coverage* column shows the percentage of the unlabeled Twitter data for which each tool was able to make a prediction. These numbers are quite a bit lower than for Wikidata and the labeled Twitter set (the names in the labeled sample contain less non-Latin alphabet text than those in the unlabeled sample). This may be due to there being many non-names in the Twitter name field, or the use of non-Latin alphabets, which many of the tools do not currently

Tool Name	Accuracy	Coverage	F1	Names/Sec
Gender.c	97.79 (96.03)	81.82 (81.72)	89.09 (88.30)	**58873.6**
Gender Guesser	97.34 (97.12)	83.02 (83.34)	89.61 (89.70)	27691.2
Gender Detector	98.43 (*98.36*)	67.55 (69.91)	80.11 (81.73)	97.8
Genderize IO	85.91 (86.69)	91.96 (92.49)	92.68 (93.11)	13.5*
Demographer: Name list	93.42 (93.74)	80.77 (82.05)	86.89 (87.98)	44445.6
Demographer: Classifier (no threshold)	87.68 (87.09)	**99.99** (*99.99*)	93.43 (93.09)	4239.0
Demographer: Classifier (0.8 threshold)	**99.15** (96.20)	39.17 (24.71)	56.16 (39.32)	
Demographer: Combined (no threshold)	90.42 (90.47)	**99.99** (*99.99*)	**94.97** (*94.99*)	14903.6
Demographer: Combined (0.8 threshold)	94.14 (94.44)	85.80 (85.68)	89.78 (89.84)	

Table 1: *Wikidata*: Tool performance on the test set (balanced test set in parentheses), evaluated in terms of accuracy, coverage, F1, and names per second (averaged over 3 runs). *Note that Genderize IO uses a web API (slower than running locally). In practice, caching locally and sending up to 10 names at once improves speed. This value reflects sending names individually without caching.

Tool Name	Coverage	F1
Gender.c	24.16	71.80
Gender Guesser	25.78	74.82
Gender Detector	35.47	70.56
Genderize IO	45.81	84.06
Dem.:Name list	31.22	79.35
Dem.:Classifier (no thresh.)	**69.73**	89.19
Dem.:Combined (no thresh.)	**69.73**	**90.80**

Table 2: *Twitter data*: Coverage is computed over the unlabeled Twitter data (526,256 unique names) and F1 over the gender-annotated Twitter names.

handle. DEMOGRAPHER provides the best coverage, as it can make predictions for previously unobserved names based on character-level features. For *F1* we report results on gender-annotated Twitter. DEMOGRAPHER, in its combined setting, performs best, with Genderize IO also performing fairly well.

We raise the following concerns, to be addressed in future work. The international nature of the Twitter data takes its toll on our models, as both the name list and classifier are based on US Social Security data. Clearly, more must be done to handle non-Latin scripts and to evaluate improvements based on language or localization (and appropriately localized training data). Our tool also makes the assumption that the user's given name appears first in the name field, that the name contains only characters from the Latin alphabet, and that the user's name (and their actual gender) can be classified as either *Male* or *Female*, all of which are known to be false assumptions and would need to be taken into consideration in situations where it is important to make a correct prediction (or no prediction) for an individual. We know that not all of the "name" fields actually contain names, but we do not know how the use of non-names in that field may be distributed across demographic groups. We did not evaluate whether thresholding had a uniform impact on prediction quality across demographic groups. Failing to produce accurate predictions (or any prediction at all) due to these factors could introduce bias into the sample and subsequent conclusions. One possible way to deal with some of these issues would be to incorporate predictions based on username, such as those as described in Jaech and Ostendorf (2015).

6 Conclusions

We introduce DEMOGRAPHER, a tool that can produce high-accuracy and high-coverage results for gender inference from a given name. Our tool is comparable to or better than existing tools (particularly on Twitter data). Depending on the use case, users may prefer higher accuracy or higher coverage versions, which can be produced by changing thresholds for classification decisions.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1232825. We thank Adrian Benton for data collection, Svitlana Volkova for information about datasets, and the reviewers for their comments and suggestions.

References

Jalal S Alowibdi, Ugo Buy, Paul Yu, et al. 2013. Language independent gender classification on Twitter. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 739–743. IEEE.

Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson, and David Yarowsky. 2013. Broadly improving user classification via communication-based name and location clustering on Twitter. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

John D Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA.

Miriam Cha, Youngjune Gwon, and HT Kung. 2015. Twitter geolocation and regional classification via sparse coding. In *Ninth International AAAI Conference on Web and Social Media*.

Jonathan Chang, Itamar Rosenn, Lars Backstrom, and Cameron Marlow. 2010. epluribus: Ethnicity on social networks. In *International Conference on Weblogs and Social Media (ICWSM)*.

Morgane Ciot, Morgan Sonderegger, and Derek Ruths. 2013. Gender inference of Twitter users in non-English contexts. In *EMNLP*, pages 1136–1145.

Ryan Compton, David Jurgens, and David Allen. 2014. Geotagging one hundred million twitter accounts with total variation minimization. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 393–401. IEEE.

Aron Culotta, Nirmal Ravi Kumar, and Jennifer Cutler. 2015. Predicting the demographics of twitter users from website traffic data. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Mark Dredze, Michael J Paul, Shane Bergsma, and Hieu Tran. 2013. Carmen: A twitter geolocation system with applications to public health. In *AAAI Workshop on Expanding the Boundaries of Health Informatics Using AI (HIAI)*.

Mark Dredze, Miles Osborne, and Prabhanjan Kambadur. 2016. Geolocation for twitter: Timing matters. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2010. A latent variable model for geographic lexical variation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, pages 451–500.

Aaron Jaech and Mari Ostendorf. 2015. What your username says about you. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2032–2037, Lisbon, Portugal, September. Association for Computational Linguistics.

David Jurgens, Tyler Finethy, James McCorriston, Yi Tian Xu, and Derek Ruths. 2015. Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. In *Proceedings of the 9th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Wendy Liu and Derek Ruths. 2013. What's in a name? using first names as features for gender inference in Twitter. In *AAAI Spring Symposium: Analyzing Microtext*.

J. Michael. 2007. 40000 Namen, Anredebestimmung anhand des Vornamens. http://www.heise.de/ct/ftp/07/17/182/.

Alan Mislove, Sune Lehmann, Yong-yeol Ahn, Jukka-pekka Onnela, and J Niels Rosenquist. 2011. Understanding the Demographics of Twitter Users. *Artificial Intelligence*, pages 554–557.

Miles Osborne, Sean Moran, Richard McCreadie, Alexander Von Lunen, Martin D Sykora, Elizabeth Cano, Neil Ireson, Craig Macdonald, Iadh Ounis, Yulan He, et al. 2014. Real-time detection, tracking, and monitoring of automatically discovered events in social media. In *Association for Computational Linguistics (ACL)*.

Marco Pennacchiotti and Ana-Maria Popescu. 2011. A machine learning approach to Twitter user classification. *ICWSM*, 11:281–288.

Israel Saeta Pérez. 2016. Gender-guesser. https://pypi.python.org/pypi/gender-guesser.

Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2016. pigeo: A python geotagging tool. *ACL 2016*, page 127.

Delip Rao and David Yarowsky. 2010. Detecting latent user properties in social media. In *Proc. of the NIPS MLSN Workshop*. Citeseer.

Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. 2010. Classifying latent user attributes in Twitter. In *Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents*.

Delip Rao, Michael Paul, Clay Fink, David Yarowsky, Timothy Oates, and Glen Coppersmith. 2011. Hierarchical bayesian models for latent attribute detection in social media. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Dominic Rout, Kalina Bontcheva, Daniel Preoţiuc-Pietro, and Trevor Cohn. 2013. Where's@ wally?:

a classification approach to geolocating users based on their social ties. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 11–20. ACM.

H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, and Lyle H Ungar. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791.

The United States Social Security Administration. 2016. Baby names. http://www.socialsecurity.gov/OACT/babynames/names.zip.

Casper Strømgren. 2016. Genderize io. https://genderize.io/.

Marcos Vanetta. 2016. Gender detector. https://pypi.python.org/pypi/gender-detector/0.0.4.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Svitlana Volkova, Yoram Bachrach, Michael Armstrong, and Vijay Sharma. 2015a. Inferring latent user properties from texts published in social media (demo). In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI)*, Austin, TX, January.

Svitlana Volkova, Benjamin Van Durme, David Yarowsky, and Yoram Bachrach. 2015b. Tutorial: Social media predictive analytics. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), tutorial*.