American Economic Journal: Economic Policy 2024, 16(4): 254–279
https://doi.org/10.1257/pol.20220611

How Do You Say Your Name? Difficult-to-Pronounce Names and Labor Market Outcomes[†]

By Qi Ge and Stephen Wu*

We test for labor market discrimination based on an understudied characteristic: name fluency. Analysis of recent economics PhD job candidates indicates that name difficulty is negatively related to the probability of landing an academic or tenure-track position and research productivity of initial institutional placement. Discrimination due to name fluency is also found using experimental data from prior audit studies. Within samples of African Americans (*Bertrand and Mullainathan 2004*) *and ethnic immigrants* (*Oreopoulos 2011*)*, job applicants with less fluent names experience lower callback rates, and name complexity explains roughly between 10 and 50 percent of ethnic name penalties. The results are primarily driven by candidates with weaker résumés, suggesting that cognitive biases may contribute to the penalty of having a difficult-to-pronounce name.* (*JEL* A11, J15, J23, J44, J71)

There is strong evidence of labor market discrimination across many dimensions, including but not limited to age (Neumark, Burn, and Button 2019), gender (Neumark, Bank, and Van Nort 1996; Goldin and Rouse 2000), race and ethnicity (Cross et al. 1990; Turner, Fix, and Struyk 1991), national origin (Pierné 2013), religion (Wright et al. 2013), sexual orientation (Carpenter 2007), and physical appearance (Hamermesh and Biddle 1994). One strand of literature focuses on discrimination based on the racial or ethnic origin of one's name. Much of this research employs audit studies using fictitious résumés, where only names are changed between otherwise similar applicants. Bertrand and Mullainathan (2004) show that those with White-sounding names receive 50 percent more callbacks than otherwise-similar applicants with African American–sounding names, while Oreopoulos (2011) finds substantial discrimination in Canada against job seekers with Indian, Pakistani, and Chinese names relative to those with English names. These types of experimental studies have been repeated in a number of different settings (see Neumark 2018 and Baert 2018 for surveys of related literature). The existence of

---

*Ge: Department of Economics, Vassar College (email: qige@vassar.edu); Wu: Department of Economics, Hamilton College (email: swu@hamilton.edu). C. Kirabo Jackson was coeditor for this article. We are grateful to Michèle Belot, Jeffrey Cross, Wayne Grove, Benjamin Ho, Rodrigo Schneider, seminar participants at Vassar College, and anonymous referees for helpful suggestions and feedback. We also thank John Nunley for providing data, and Matthew Bartok, Kathryn Biederman, Alex Eisert, Jacob Gliedman, Daniel Goldstein, Shijun Hong, Taicheng Jin, James Kaffenbarger, Erin Kuo, Dewayne Martin, Maroun Mezher, Griffin Perry, Josue Herrera Rivera, Matthew Surprenant, Kenneth Talarico, Gwendolyn Urbanczyk, Adam Valencia, Fiona Xiang, and Chenyu Zhou for outstanding research assistance. All errors are ours.

[†] Go to https://doi.org/10.1257/pol.20220611 to visit the article page for additional materials and author disclosure statement(s) or to comment in the online discussion forum.

name discrimination along racial or ethnic lines can have significant consequences, as some may decide to change their names (Biavaschi, Giulietti, and Siddique 2017) or engage in résumé "Whitening," where job applicants strategically omit information that could signal their race or ethnicity, such as participation in race-based organizations or affinity groups (Kang et al. 2016).

In this paper, we study the labor market effects of a related but distinct characteristic of names: name fluency. While a significant body of literature documents discrimination against names that signal one's racial or ethnic background, to our knowledge, no studies have tested the hypothesis that having a difficult-to-pronounce name may harm job market outcomes. Although many ethnic-sounding names are difficult to pronounce for those outside of that ethnic group, there is still variation in name fluency within particular groups. For example, most non-Chinese speakers would consider "Chen" to be more familiar and easier to pronounce than "Xiang"; people without a Polish background will generally have more trouble trying to pronounce "Przybyłko" than "Nowak."

Those with less fluent names may face disadvantages in the labor market. Employers may have subconscious bias against those with difficult-to-pronounce names, resulting in more negative evaluations of these applicants. Recruiters will also have an easier time processing and remembering names that are more fluent. Indeed, there is evidence consistent with these theories across various nonlabor market settings. Alter and Oppenheimer (2006) demonstrate that a basket of shares with easy-to-pronounce names yielded an 11 percent higher return than a comparable basket with less fluent stock names. Laham, Koval, and Alter (2012) conduct a series of experiments in Australia and find that those with easier-to-pronounce names are judged more positively by others and also hold more prominent positions in law firms. In Newman et al. (2014), university students in New Zealand were more likely to believe claims from those with easy-to-pronounce names relative to those with more difficult names. And processing fluency can even affect motivation and behavior, as shown in Song and Schwarz (2008), who find that having instructions written in hard-to-read fonts lowers willingness to complete certain tasks.

In this study, we examine multiple sources of data and document significant evidence of labor market discrimination based on name fluency. First, we utilize observational data from the academic labor market by assembling curriculum vitae of over 1,500 economics job market candidates from 96 top-ranked economics PhD programs from the 2016–2017 and 2017–2018 job market cycles and find that name fluency is significantly related to job market outcomes. Candidates with difficult-to-pronounce names are less likely to be initially placed into an academic or tenure-track position, and they are placed in jobs at institutions with lower research productivity as ranked by the Research Papers in Economics (RePEc) database. Our results are consistent and robust across three separate ways of measuring pronunciation difficulty: an algorithmic rating based on commonality of letter and phoneme combinations, a survey-based measure that records the median time it takes individuals to pronounce a name, and a purely subjective measure based on individual ratings. These results hold after controlling for a large number of covariates, including PhD institution and country of origin.

We augment these observational findings by employing experimental data from two prior audit studies discussed above. Analysis of replication data for Bertrand

and Mullainathan (2004) shows that job applicants with less fluent names have lower callback rates, and even within the sample of résumés with distinctly African American names, name fluency is strongly correlated with callback rates. We obtain similar results using replication data for another prior audit study by Oreopoulos (2011). Once again, job applicants are less likely to be called back when they have names that are difficult to pronounce, and when restricting the sample to immigrants with ethnically Indian, Pakistani, and Chinese names, those with less fluent names are significantly less likely to be called for a job interview. Name complexity explains between roughly 10 and 50 percent of the ethnic name penalty documented in Bertrand and Mullainathan (2004) and Oreopoulos (2011). Our results have important implications for audit studies looking to test for name-based racial and ethnic discrimination. In particular, researchers should consider selecting names that are similar in pronunciation difficulty when generating fabricated résumés, and empirical analysis should also control for a name's pronunciation difficulty.

We also explore possible mechanisms for our findings by testing for heterogeneity in the size and strength of name penalties according to the profiles of job applicants. Across all three data sources, those with weaker résumés encounter more discrimination due to name pronunciation than their counterparts with stronger résumés. This evidence is consistent with the idea that processing and remembering less fluent names entail mental costs, and these mental costs are only worth spending on higher-quality candidates.

There are several unique aspects of this paper. We test for labor market discrimination due to an understudied characteristic: name fluency. In doing so, our work also introduces key methodological innovations. Unlike most research on name-based discrimination, we employ observational data using actual names and résumés, in addition to our analysis of prior audit studies that randomly assign names to fictitious applicants. Our data also incorporate hiring outcomes, which complements the analysis of callback rates that are the focus of most earlier research. Moreover, we contribute to the literature on word fluency by incorporating three independent ways of rating pronunciation difficulty. One measure is similar to what has been used before: independent subjective assessments. The other two measures are novel in their approaches: an algorithm that provides a relatively objective way to rate name fluency, and the median time it takes people to pronounce someone's name. These innovative measures can be used by others studying the impacts of textual fluency. Finally, our analysis documents a mechanism for discrimination that is different than what other research has emphasized. While the vast majority of prior work has focused on the role of preferences or statistical discrimination in explaining racial differences in labor market outcomes, we find that the *cognitive* cost associated with hard-to-pronounce names is an important channel through which name-based discrimination operates.

I. Observational Data and Empirical Strategy

A. *Economics PhD Job Market Candidates*

As an initial test of the impact of name pronunciation on job market outcomes, we employ data on economics PhD candidates who entered the job market in either

2016–2017 or 2017–2018.[1] We manually collected curriculum vitae from professional websites or departmental placement pages for PhD students from the 96 top-ranked economics doctoral programs, as ranked by the *US News & World Report* (USNWR) in 2017.[2] Although these data represent only one specific labor market, there are advantages to studying the job market for economics PhDs. We have a fairly comprehensive pool of candidates searching for jobs requiring a doctoral degree in economics, given the large number of graduate programs included in our sample. In addition, curriculum vitae are publicly available and tend to follow consistent formats, providing reliable information about candidates' research productivity and teaching experience. Furthermore, the ranking of departments according to research productivity offers one measure of job placement quality, albeit an incomplete and imperfect one.

Initial job market outcomes are primarily obtained from departmental web pages that list the placement histories of their graduates. When departments do not list specific names in their placement outcomes, we search for this information on personal websites or publicly available LinkedIn profiles.[3] We categorize job placement into three categories: academia, government or research think tanks, and the private sector. Within academic positions, we distinguish between tenure-track jobs and other academic positions such as postdoctoral fellowships or visiting (non-tenure-track) positions.[4]

For academic, governmental, and think tank positions, placement outcomes are ranked based on institutional research productivity from the RePEc database, as published in February 2019.[5] Compared to alternative indices on economics research productivity, such as those introduced in Coupé (2003) that focus exclusively on economics departments in colleges and universities, the RePEc productivity ranking covers a broader range of institutions and thus offers a more comprehensive indicator of placement quality. In particular, the RePEc index provides explicit productivity scores from 1 to 389 (in descending order of productivity) for the top 5 percent of all economics institutions around the world. Institutions ranked in the top 6–10 percent are only grouped by single percentiles and are not ranked within these groupings. For the purposes of this study, all institutions ranked in the top sixth percentile are coded as having a RePEc productivity score of 400. Similarly, the remainder of institutions ranked in the top 10 percent are coded as having a RePEc score of 500 (seventh percentile), 600 (eighth percentile), 700 (ninth percentile), or 800 (tenth percentile). Institutions not listed among the top 10 percent are coded as having a RePEc score of 1,000. For academic positions, we only use the rank of an institution in the top 10 percent when the individual has been placed in a tenure-track job. Candidates placed in

---

[1] These data are identical to those of Ge, Wu, and Zhou (2021a) that have been used to study the impact of student–advisor matching on job placement outcomes.

[2] The 2017 USNWR listing of top graduate programs in economics rates programs all the way down to number 90, but because of ties, the list includes a total of 96 programs.

[3] Out of the 96 USNWR-ranked economics PhD programs considered in our sample, 82 list their placement outcomes with candidates' names attached. The 14 programs that do not list specific names in their placement records are fairly evenly distributed across the spectrum of the USNWR ranking.

[4] If a candidate lists a postdoc as the initial placement as well as a tenure-track position that will start upon completion of the postdoc, we use the tenure-track position as the job market outcome.

[5] Given the stability of the RePEc rankings over time, the particular choice of the ranking date does not impact our findings.

postdoctoral fellowships or non-tenure-track academic positions are given a RePEc score of 1,000, even if the institution is ranked among the top 10 percent. To maximize the number of observations, individuals placed in the private sector or those with missing job information have been given an imputed RePEc score of 1,000. Although much of our analysis focuses on binary measures such as whether or not someone initially placed in an academic or tenure-track job, when analyzing the quality of placement outcomes, we employ the continuous RePEc scores and estimate tobit regressions with right censoring at 1,000. We also show that our results are robust to excluding individuals placed in the private sector or those without job placement information.

B. *Name Fluency Ratings*

To measure the difficulty in pronouncing individual names, we use several different methods: a computer-generated algorithm that assesses the difficulty of pronouncing various words, a rating based on the median time it takes for people to pronounce a particular name, and a subjective measure based on three independent raters. We discuss each of these methods in more detail below.

In prior research on word fluency, a common way to rate pronunciation difficulty is to have individuals use their own criteria to decide whether a word is easy or difficult to pronounce (Laham, Koval, and Alter 2012; Newman et al. 2014). While this type of rating scheme is straightforward, it is prone to potential bias, as an individual's subjective perception of the difficulty in pronouncing a name may not equate to how hard it actually is to pronounce a name in practice. Therefore, our primary measure of name fluency is based on a more objective method where we created an algorithm to rate the difficulty of pronouncing words from the vantage point of a native English speaker.

This algorithm develops two measures of difficulty in pronouncing words: one based on sequences of letters and the other based on sequences of phonemes, or distinct units of sound. Occurrence dictionaries of common letter and phoneme combinations were then used to develop one difficulty rating based on the frequency of letter combinations and another rating based on the frequency of phoneme combinations embedded in various names,[6] resulting in two relatively objective ratings of fluency/familiarity for first names and two ratings for last names. We then obtain a weighted average of the letter-based score and the phoneme-based score using a weighting scheme derived from neural network learning to measure the difficulty of pronouncing a particular first name or last name, and these scores are scaled between 0 and 100.[7] Our primary name difficulty measure is the sum of the first name complexity score and the last name complexity score, providing a composite measure of the pronunciation difficulty for each individual's full name, but we also conduct additional analysis using first name complexity and last name complexity as separate variables.[8] Research in

---

[6] More specifically, the letter combination algorithm would first assess the commonality of individual letters, then consecutive groupings of two letters, all the way up to *n* letters, where *n* equals the number of letters in the entire word. Similarly, frequency of "phoneme *n*-grams" were calculated, starting with sequences of one phoneme, then two phonemes, up to *n* phonemes, where *n* is the total number of phonemes in a specific name. These "*n*-gram" calculations are common in linguistic assessment of word complexity.

[7] The two subrating schemes are highly correlated with one another.

[8] We show annotated code for our name fluency algorithm in a separate technical online Appendix, and the complete files necessary to execute the algorithm are available upon request.

computational linguistics has shown that these types of fluency measures correlate well with human judgments of word complexity for a number of languages, including English (Wilson and Raaijmakers 2008; Bojar et al. 2016; Popović 2018).

To provide a sense of how names of a particular racial/ethnic origin might vary according to this algorithmic rating, some examples of Chinese given names from our economics PhD sample and their associated difficulty include Bin (one standard deviation easier than the average for the entire sample), Dede (average difficulty), Yunfan (one standard deviation above average), and Zhiyi (two standard deviations above average). Analogous examples of Indian surnames from our PhD data and their associated difficulty include Gupta (almost one standard deviation easier than average), Singh (average difficulty), Chaturvedi (one standard deviation above average), and Seetharam (two standard deviations above average). Examples of distinctively African American first names taken from the audit study by Bertrand and Mullainathan (2004) include Ebony (one standard deviation easier than the average), Jermaine (average difficulty), Aisha (one standard deviation above average), and Lakisha (two standard deviations above average).

As a second way to rate pronunciation difficulty, we hired ten undergraduate research assistants to independently go through a series of surveys of first names and another series of surveys of last names of the job market candidates in our data and attempt to pronounce each name.[9] Each screen for a particular survey contained one name; we asked the research assistants to read through the name and immediately click to advance to the next screen for the next name.[10] This procedure was then repeated until a particular survey of names was completed. Each survey contained timing functions that kept track of the length of time that each person spent on a particular screen before moving onto the next name. We summed the time it took each research assistant to pronounce a candidate's first name and last name to obtain the total time spent on the candidate's full name. We then used the median time (out of all research assistants) spent on a candidate's full name as our second measure of pronunciation difficulty. The full instructions on this method are included in online Appendix E.

As a third measure of name complexity, we implement the subjective method common in prior research, in spite of its shortcomings. In following this method, three individuals conducted independent ratings for each first name in a binary manner: the name was deemed either difficult to pronounce or easy to pronounce. Independent ratings of last names were done in the same fashion. This rating system provided a number of different possible ways to code the name difficulty, but we primarily used a dichotomous variable that was equal to one if at least two raters deemed a candidate's first or last name as being difficult to pronounce.[11]

We recognize that there are several possible sources of measurement error in rating the fluency of names. The algorithm-based rating is based on the perspective of

---

[9] We assembled a fairly diverse group of research assistants across race, ethnicity, gender, and country of origin. The research assistants were also blind to the hypotheses of our study.

[10] We specifically grouped names such that each individual survey contained only first names or only last names. Given that no full names were ever listed together, the likelihood that any of the research assistants would recognize a particular candidate is very low.

[11] We also tried alternative schemes, such as having at least one person rate a first or last name as being difficult, having all three consider a name as difficult, or summing the total number of raters that deemed a candidate to have a difficult-to-pronounce first or last name. Results are consistent across different methods.

a native English speaker, and search committees may have individuals from a variety of native countries. Similarly, the team of survey-taking research assistants, while reflecting diversity across a number of demographic characteristics, may or may not reflect the typical makeup of a hiring committee. And subjective ratings are always potentially flawed for reasons discussed earlier. Nonetheless, we believe that the use of three independent ways of rating pronunciation difficulty provides a comprehensive way to test for labor market discrimination due to name fluency.

C. *Empirical Strategy*

As an initial test of the hypothesis that difficulty of name pronunciation affects a job candidate's initial placement, we estimate the following probit model:

$$Pr\left(Placement_i = 1\right) = \Phi\left(\delta' Pronounce_i + \phi' X_i + \theta' Z_i\right) \tag{1}$$

where $Placement_i$ denotes dichotomous placement outcome variables for candidate i, including (i) an indicator that is equal to one if the candidate is placed in academia (including tenure-track, visiting, or postdoc positions) and (ii) an indicator that is equal to one if the candidate is placed into a tenure-track position.

X_i is a vector of job candidate characteristics in line with prior literature (e.g., Krueger and Wu 2000; Athey et al. 2007; and Sullivan, Dubnicki, and Dutkowsky 2018, among others) that includes gender, whether one attended an elite US undergraduate institution,[12] prior graduate degree(s), number of publications, number of publications in top-five economics journals,[13] number of papers currently under revise-and-resubmit status, number of revise-and-resubmit invitations at top-five journals, and dichotomous variables for the receipt of a teaching award and for having independent instructor experience. In addition, Z_i captures the following characteristics of candidate i's primary dissertation advisor: the number of publications in top-five journals, having editorial experience at an academic journal, and having editorial experience at a top-five journal. Finally, we control for a range of fixed effects that may affect placement outcomes, including region of undergraduate degree (United States and Canada, Latin America and the Caribbean, Eastern Europe, Western Europe, South Asia and the Middle East, East and Southeast Asia, Australia, and Africa), subfield (theory, macro/finance, econometrics, and applied micro), job market cycle (2016–2017 versus 2017–2018), and PhD institution.

Our independent variable of interest, $Pronounce_i$, represents the difficulty of pronouncing a candidate's name, measured in several different ways as described earlier. One potential confounding issue in measuring name fluency is that longer names may be more difficult and take longer to pronounce, all else constant. To distinguish between the effects of name length and name difficulty, we control for the number of letters in a candidate's first and last names.

---

[12] Similar to Athey et al. (2007), we define elite undergraduate institutions as (i) Ivy League universities, (ii) other top 15 national universities according to the 2019 USNWR college ranking, and (iii) top-five liberal arts colleges, as rated by the USNWR.

[13] We use the *Quarterly Journal of Economics*, the *American Economic Review*, the *Review of Economic Studies*, the *Journal of Political Economy*, and *Econometrica* as our group of top-five journals.

As a second measure of placement outcomes, we also investigate the determinants of job placement quality based on the RePEc database's ranking of economics institutions according to their research productivity. For our main specifications, we impute the placement quality of candidates placed in the private sector or candidates obtaining non-tenure-track jobs at academic institutions with a RePEc score of 1,000 and run a tobit specification with an upper limit at 1,000, but we also reestimate our regressions by excluding these observations and obtain similar results. Similar to equation (1), our tobit models include job candidate characteristics, advisor characteristics, name fluency rating, and name length, as well as fixed effects for region, subfield, job market cycle, and PhD program.

D. *Descriptive Statistics*

Table 1 presents the summary statistics for our sample that contains a total of 1,510 candidates from the top 96 ranked economics PhD programs who were on the job market in either 2016–2017 or 2017–2018. Most of the job market candidate characteristics are in line with prior studies. Seventy percent of our sample is male, slightly over half completed a prior graduate degree, about 40 percent completed their undergraduate education in the United States, and close to 10 percent attended an elite US undergraduate institution. The majority of job market candidates have limited publication experience as revealed by the mean number of publications or papers receiving a revise-and-resubmit invitation, both of which are less than one. In terms of job market outcomes, 65 percent of candidates are initially placed into academic jobs, with the majority of these being tenure-track positions. As explained earlier, those who are placed in private sector positions are assigned the highest (worst) RePEc ranking of 1,000. The mean imputed RePEc ranking is close to 800, equivalent to the top 10 percent of economics institutions around the world, but the median is 1,000.[14]

Turning to our measures of name fluency, we find that the computer-generated algorithm produces a wide range of pronunciation ratings for full names, with a low of 0 and a high of 200, and mean and median values of approximately 80. The average time it takes to pronounce a candidate's full name is approximately 2.5 seconds, though this also has a wide range, from just over 1.5 seconds to nearly 6 seconds. Slightly less than 30 percent of names were subjectively rated as having a difficult-to-pronounce first or last name by at least two out of three individual raters.

II. Results for Observational Data

A. *Baseline Findings*

Table 2 shows the probit marginal effect estimates of the likelihood of being initially placed in an academic job (columns 1–2) or a tenure-track job (columns 3–4), and tobit estimates for the research ranking of initial placement (columns 5–6). In column 1 (column 3), we see that getting an academic (tenure-track) job is positively

---

[14] The mean and median RePEc rankings without imputation ($N = 910$) are 635 and 1,000, respectively.

Table 1—Sample of Economics PhD Job Candidates: Summary Statistics

	Mean (1)	Median (2)	Std. dev. (3)	Min (4)	Max (5)
Candidate characteristics					
Male	0.697	1	0.460	0	1
US undergrad	0.415	0	0.493	0	1
Elite US undergrad	0.089	0	0.284	0	1
Prior graduate degree	0.552	1	0.497	0	1
PhD rank	32.934	27	25.812	1	90
Number of revise-resubmits	0.160	0	0.465	0	5
Number of top-five rev-resubs	0.022	0	0.151	0	2
Number of publications	0.674	0	1.303	0	17
Number of top-five pubs	0.011	0	0.102	0	1
Teaching award	0.206	0	0.405	0	1
Instructor experience	0.521	1	0.500	0	1
Advisor characteristics					
Number of top-five pubs	5.584	3	7.826	0	61
Top-five journal editor exp	0.125	0	0.330	0	1
Editor experience	0.784	1	0.412	0	1
Placement outcomes					
Academia	0.647	1	0.478	0	1
Tenure track	0.460	0	0.499	0	1
Visiting	0.078	0	0.268	0	1
Postdoc	0.109	0	0.311	0	1
Government/think tank	0.152	0	0.359	0	1
Industry	0.201	0	0.401	0	1
US job	0.701	1	0.458	0	1
Imputed RePEc ranking	779.852	1,000	361.874	1	1,000
Name difficulty					
Algorithm rating	78.978	81.865	44.582	0	200
Pronunciation time	2.525	2.429	0.537	1.539	5.818
Subjectively difficult	0.286	0	0.452	0	1

Notes: The sample includes 1,510 PhD candidates from the 96 top-ranked economics doctoral programs who entered the job market in the 2016–2017 and 2017–2018 job market cycles.

related to the number of publications and revise-and-resubmit invitations and to having completed a prior graduate degree or receiving a teaching award, though the last coefficient is not quite statistically significant for the tenure-track regression. Column 5 exhibits negative and significant coefficients for prior graduate degrees, number of papers under revise-and-resubmit status, number of top-five revise-and-resubmit papers, and winning a teaching award, implying that these factors are all correlated with placing in more productive research institutions as a smaller RePEc score corresponds to an institution with greater research output.[15] Having an advisor who is a journal editor increases the probability of being placed in an academic job (column 1), but it is not significantly related to the other outcome variables. We do not find significant gender differences for any job outcomes.

We now turn to the focus of this study: the effect of name fluency on job outcomes. Our algorithmic measure of name fluency is normalized in all regressions

---

[15] As described earlier, our main specifications include a full set of PhD program fixed effects, which slightly reduces the sample size in the probit regressions from the original 1,510 to 1,469 (columns 1–2) and 1,499 (columns 3–4) due to collinearity.

Table 2—Name Fluency and Placement Outcomes: Algorithm Rating

	Academia (1)	Academia (2)	TT (3)	TT (4)	RePEc_Imputed (5)	RePEc_Imputed (6)
Candidate characteristics						
Male	−0.001	−0.001	−0.018	−0.017	−21.575	−21.957
	(0.029)	(0.029)	(0.031)	(0.031)	(57.951)	(58.059)
Elite US undergrad	0.051	0.050	0.043	0.040	−45.538	−43.752
	(0.045)	(0.045)	(0.050)	(0.050)	(88.017)	(87.767)
Prior graduate degree	0.096	0.095	0.081	0.080	−133.562	−132.712
	(0.031)	(0.031)	(0.033)	(0.033)	(62.361)	(62.531)
Number of revise-resubmits	0.065	0.067	0.109	0.111	−139.899	−140.820
	(0.035)	(0.035)	(0.034)	(0.034)	(57.295)	(57.560)
No. of top-five rev-resubs	0.146	0.149	0.130	0.135	−286.184	−287.994
	(0.107)	(0.108)	(0.104)	(0.105)	(134.393)	(135.524)
Number of publications	0.029	0.029	0.025	0.026	−26.548	−26.946
	(0.012)	(0.012)	(0.012)	(0.013)	(19.931)	(19.992)
Number of top-five pubs	0.271	0.274	0.148	0.148	−161.312	−159.318
	(0.069)	(0.067)	(0.129)	(0.130)	(164.210)	(165.784)
Teaching award	0.076	0.077	0.061	0.063	−114.694	−115.333
	(0.034)	(0.034)	(0.037)	(0.038)	(64.041)	(64.103)
Instructor experience	0.013	0.013	−0.008	−0.007	57.480	56.622
	(0.032)	(0.032)	(0.033)	(0.033)	(60.609)	(60.696)
Advisor characteristics						
Number of top-five pubs	0.004	0.004	−0.000	−0.000	−1.995	−2.100
	(0.003)	(0.003)	(0.002)	(0.002)	(3.984)	(4.000)
Top-five journal editor exp	−0.052	−0.055	0.015	0.011	−0.329	1.741
	(0.053)	(0.053)	(0.052)	(0.052)	(78.595)	(78.581)
Editor experience	0.096	0.095	0.048	0.047	6.678	6.753
	(0.036)	(0.036)	(0.036)	(0.036)	(74.361)	(74.360)
Name difficulty						
Algorithm rating: Full name	−0.030		−0.006		67.201	
	(0.016)		(0.017)		(31.688)	
Algorithm rating: First name		−0.030		−0.002		47.456
		(0.017)		(0.017)		(32.297)
Algorithm rating: Last name		−0.011		−0.006		51.148
		(0.018)		(0.018)		(34.316)
Observations	1,469	1,469	1,499	1,499	1,510	1,510
Control for name length	Yes	Yes	Yes	Yes	Yes	Yes
Subfield/program FE	Yes	Yes	Yes	Yes	Yes	Yes
Region/job mark cycle FE	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The coefficients in columns 1–4 are marginal effects of probit regressions. The dependent variable in columns 1–2 (3–4) is a dichotomous variable for being placed in an academic (tenure-track) position. Columns 5–6 are estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of 1,000, the highest (worst) ranking. All tobit regressions are censored with an upper limit of 1,000. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based subrating schemes, where the weights are derived from neural network learning. Robust standard errors are in parentheses.

for ease of interpretation. The coefficient in column 1 implies that a one standard deviation increase in pronunciation difficulty decreases a candidate's likelihood of landing an academic job by 3 percentage points, and this coefficient has a p-value of 0.06. The analogous effect of a one standard deviation increase in name difficulty on the likelihood of obtaining a tenure-track position (column 3) is 0.6 percentage points, though this coefficient is not precisely estimated. In terms of the ranking of an individual's initial placement (column 5), a one standard deviation increase in name difficulty is correlated with an initial job placement that is 67 spots worse in the RePEc ranking, and the coefficient is significant at the 5 percent level. This name penalty translates to a difference between a job in the economics departments at Princeton University versus that of the University of California, Irvine, for example, or the difference between a job at Duke University's Fuqua School of Business and a job at the University of Essex.[16]

Next, we test to see whether there are differences between the fluency effects of first names and last names. Column 2 suggests that hard-to-pronounce first names and last names are both negatively correlated with placing in academia, though the coefficient for last names has a large standard error. The marginal effects for a one standard deviation increase in name difficulty of first and last names is -3.0 and -1.1 percentage points, respectively, though a t-test fails to reject the statistical equivalence of the two coefficients. In column 4, neither of the coefficients for first-name or last-name difficulty is statistically significant in predicting tenure-track placements. With regards to the institutional ranking of initial placement, pronunciation difficulty of first and last names has statistically indistinguishable coefficients of 47 and 51, respectively.

Table 3 shows results using two alternative ways to rate pronunciation difficulty: the normalized median time it took research assistants to pronounce a job candidate's full name (columns 1, 3, and 5), and an indicator variable equal to one if at least two independent raters deemed the first name or the last name to be difficult to pronounce and to zero otherwise (columns 2, 4, and 6). These results are even stronger and larger in magnitude than results using our algorithmic measure of name fluency, and we also observe that pronunciation time significantly decreases the likelihood of a getting a tenure-track position, in addition to affecting chances of getting an academic job and the research ranking of initial placement. Taken together, the combined results paint a clear and consistent picture: those with difficult-to-pronounce names are penalized in the academic job market.

B *Potential Mechanisms*

While we are unable to precisely identify the mechanisms through which discrimination based on name fluency acts, we can speculate about some possibilities. For job searches at academic, governmental, and research institutions, initial screening generally involves committees getting together to discuss names of potential candidates, which may lead to subconscious discrimination against names that are harder to pronounce. Extensive research in psychology and other social sciences has demonstrated implicit bias occurring in a number of realms, including

---

[16] As discussed in Section IIB, the algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based subrating schemes, where the weights are derived from neural network learning. As a robustness check, we also consider an alternative algorithm rating based on an arithmetic average of the aforementioned subrating schemes and reestimate our main specifications. The results, as presented in Table A1 in the online Appendix, are similar to our baseline estimates.

Table 3—Name Fluency and Placement Outcomes: Pronunciation Time and Subjective Rating

	Academia (1)	Academia (2)	TT (3)	TT (4)	RePEc_Imputed (5)	RePEc_Imputed (6)
Pronunciation time: Full name	−0.074 (0.018)		−0.047 (0.018)		82.734 (33.930)	
Subjectively difficult name		−0.116 (0.033)		−0.055 (0.033)		82.110 (60.897)
Observations	1,469	1,469	1,499	1,499	1,510	1,510
Control for name length	Yes	Yes	Yes	Yes	Yes	Yes
Other controls	Yes	Yes	Yes	Yes	Yes	Yes
Subfield/program FE	Yes	Yes	Yes	Yes	Yes	Yes
Region/job mark cycle FE	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The coefficients in columns 1–4 are marginal effects of probit regressions. The dependent variable in columns 1–2 (3–4) is a dichotomous variable for being placed in an academic (tenure-track) position. Columns 5–6 are estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of 1,000, the highest (worst) ranking. All tobit regressions are censored with an upper limit of 1,000. The pronunciation time rating is a survey-based measure that records the median time that it takes individuals to pronounce a name. The subjective difficulty rating is based on individuals' independent subjective assessments. Robust standard errors are in parentheses.

judges in courtrooms (Kang et al. 2011), physicians and the provision of health care (Chapman, Kaatz, and Carnes 2013), and teachers in the classroom (Staats 2016).

Another possibility is that there are mental costs to processing and remembering less fluent names, and that these mental costs are only worth spending on higher-quality candidates. The theory of cognitive load posits that there is a certain amount of working memory in the human brain, so increases in cognitive load can impair learning and task performance (Plass, Moreno, and Brünken 2010; Sweller 2011). Of particular relevance is the work by Nordstrom, Williams, and LeBreton (1996), who find that cognitively demanding environments impede employers' ability to generate interview questions, thus hampering the evaluation of job candidates. Nevertheless, there is also research describing ways that people can manage and develop strategies to overcome increased cognitive load (Bannert 2002; Ang, Zaphiris, and Mahmood 2007; Paas and Sweller 2012).

One way to distinguish between these two possibilities is to test whether the penalty for having a difficult-to-pronounce name depends on the strength of the job candidate's profile. Research in behavioral economics suggests that individuals are more likely to develop heuristics when tasks involve greater uncertainty and more mental effort (Tversky and Kahneman 1974; Kahneman et al. 1982). Compared to candidates with strong résumés, job seekers with weak résumés present more uncertainty to employers in terms of their unobserved quality and necessitate more screening effort, making employers more likely to resort to mental shortcuts. One such heuristic may involve name pronunciation because of the additional mental costs associated with processing and remembering less fluent names. If cognitive load is largely responsible for name-based discrimination, candidates with weaker profiles will experience more name-based bias than those with stronger profiles, but if name penalties are primarily due to statistical or taste-based discrimination, we would not expect to see this heterogeneity.

Table 4—Name Fluency and Placement Outcomes by Program Rankings

	Top-ranked program			Lower-ranked program		
	Academia (1)	TT (2)	RePEc_Imputed (3)	Academia (4)	TT (5)	RePEc_Imputed (6)
Algorithm rating: Full name	−0.010	−0.006	66.205	−0.054	0.010	188.168
	(0.025)	(0.027)	(40.304)	(0.030)	(0.033)	(97.241)
Observations	594	594	594	364	385	398
Control for name length	Yes	Yes	Yes	Yes	Yes	Yes
Other controls	Yes	Yes	Yes	Yes	Yes	Yes
Subfield/program FE	Yes	Yes	Yes	Yes	Yes	Yes
Region/job market cycle FE	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The coefficients in columns 1–2 and 4–5 are marginal effects of probit regressions. The dependent variable in columns 1 and 4 (2 and 5) is a dichotomous variable for being placed in an academic (tenure-track) position. Columns 3 and 6 are estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of 1,000, the highest (worst) ranking. All tobit regressions are censored with an upper limit of 1,000. Top-ranked (lower-ranked) programs are those ranked within the top 20 (outside of the top 50) in the 2017 edition of the USNWR Best Economics Schools. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based subrating schemes, where the weights are derived from neural network learning. Robust standard errors are in parentheses.

To test these hypotheses, we use the economics PhD program ranking published by the USNWR as a proxy for the strength of one's profile and conduct separate analyses for applicants coming from top- versus lower-ranked programs, where we define top-ranked (lower-ranked) programs as those ranked within the top 20 (outside of the top 50) in the 2017 edition of the USNWR ranking.[17] Table 4 reports the estimated coefficients for the algorithm-based name difficulty measure. Candidates from less prestigious institutions experience a much larger name penalty on the likelihood of being placed into an academic position (column 4 versus column 1) and on the quality of placement outcomes (column 6 versus column 3) than those from top-ranked programs, though the differences between coefficients are not statistically significant at conventional levels. These results imply that the observed name pronunciation penalty may work through the channel of increased mental costs necessary to process and remember candidates with difficult names, and that hiring committees are only willing to expend this energy on candidates with strong credentials.

C. *Robustness Checks*

We perform a number of robustness checks to our main specifications, including using the raw RePEc rankings without imputations as an alternative measure for placement quality, estimating multinomial logit regressions for different types of placements, employing an ordered probit model based on categories of the RePEc ranking of job placements, conducting separate analyses by gender, and assessing the potential effect due to the uniqueness or commonality of names.

---

[17] When accounting for ties in the USNWR ranking, there are actually 52 institutions ranked within the top 50.

We also account for the possibility that students who have advisors or other committee members from the same country might be less likely to feel the need to Americanize/Anglicize their (first) names. Ge, Wu and Zhou (2021b) document a beneficial impact of student–graduate-committee matching, in the form of country of origin and native language, on students' initial placement outcomes in the economics PhD job market, which could lead to a downward bias in the estimate of the magnitude of the name fluency effect. To address this, we reestimate our baseline specifications and add controls for student–graduate-committee matching based on country (United States versus non–United States) or native language (English versus other), but results on name fluency are unchanged. Another potential concern is that difficult-to-pronounce names may be concentrated in a few countries, and the lack of success that individuals from these countries have in finding prestigious academic jobs is not necessarily linked to their names but to more general discrimination due to national origin. All regressions shown in our tables have controlled for the region of one's undergraduate school, but we have also estimated specifications that include a full set of individual country effects, and the results are largely identical. To streamline the discussion in the main text, we present these additional analyses in online Appendixes A and B.

III. Experimental Data from Prior Audit Studies

Although our analysis of economics PhD candidates controls for a large set of characteristics related to the individuals and their graduate schools and advisors, name fluency may still correlate with unobserved variables that could also impact job placement. As an additional way to test the hypothesis that name pronunciation affects job prospects, we incorporate experimental data from two important and well-cited audit studies that analyze how callback rates for job applicants are impacted by having either a distinctively African American name (Bertrand and Mullainathan 2004) or being a foreigner with an ethnic-sounding name (Oreopoulos 2011).[18]

The experiment by Bertrand and Mullainathan (2004) involved sending fictitious résumés to job postings in the areas of sales, administrative support, clerical services, and customer services in the cities of Boston and Chicago, where applicants were randomly assigned either distinctively White-sounding names or distinctively Black-sounding names.[19] Meanwhile, Oreopoulos (2011) conducted an audit study in Canada on skilled immigrant workers, with a particular focus on those with Indian, Pakistani, and Chinese backgrounds. In his data, all jobs

---

[18] As another test, we also investigate labor market effects of name fluency using data from Nunley et al. (2015a), who perform an audit study to examine racial discrimination in the labor market for recent college graduates. Our results, shown in Table A17 in the online Appendix, indicate that after controlling for race and other résumé characteristics, name complexity is negatively related to the probability of receiving a callback, with the coefficient marginally significant at the 10 percent level. Nevertheless, a major shortcoming of these data for the purpose of our study is that in Nunley et al. (2015b), they use only 8 unique names (2 for each gender–race combination), far fewer than the number used by Bertrand and Mullainathan (2004) (36 names) or Oreopoulos (2011) (44 names), limiting statistical power once controls for race and gender are included. We therefore leave the details of this analysis to online Appendix D in order to streamline the presentation in the main text.

[19] Note that the publicly available dataset for Bertrand and Mullainathan (2004) contains first names but not last names. Our analysis on name difficulty for these data will therefore be limited to first names, which are also the focus for their paper.

required a bachelor's degree or higher, and all job applicants possessed at least this level of education, so this serves as a nice complement to the data from Bertrand and Mullainathan (2019). For each of the audit studies, four résumés were sent to every job advertisement, with résumés varying by name; ethnicity; and either quality (Bertrand and Mullainathan) or source of undergraduate degree, Canadian or foreign (Oreopoulos), so all regressions use standard errors that are clustered at the level of the job identification number.

A. *Correlations between Name Difficulty and Callback Rates*

Of particular benefit for our study is the fact that within each racial/ethnic group, there are many different names randomly assigned to the various job applicants, so we can apply our previously described algorithm to generate ratings of name pronunciation. As a preliminary way to measure the relationship between name difficulty and callback rates, we calculate a simple Pearson correlation between callback rates and algorithmic ratings of name difficulty for the 18 different Black-sounding names in Bertrand and Mullainathan (2004). In the top panel of Table 5, which shows some examples of names and their respective difficulty ratings,[20] we see that three of the most familiar/fluent names (Ebony, Kenya, Leroy) have relatively high callback rates of 9.62, 8.67, and 9.38 percent, respectively, while three of the least familiar/fluent names (Aisha, Keisha, and Lakisha) have relatively low callback rates of 2.22, 3.83, and 5.5 percent, respectively. The overall correlation coefficient between name difficulty and callback rates among all Black applicants is -0.488, which is statistically significant with a p-value of 0.04.

One potential concern is that names may signal additional information about a candidate beyond their racial identity, such as their socioeconomic background, as shown in Fryer and Levitt (2004). Some researchers have scrutinized the choice of names in various audit studies testing for discrimination (Darolia et al. 2016), particularly those focusing on distinctively African American names, given the potential to conflate the effect of socioeconomic status with race (Hansen and Tummers 2020). Several recent papers provide guidance for those conducting audit studies on how to more effectively isolate the pure effect of racial and ethnic discrimination by choosing names that signal different races but do not vary by perceived socioeconomic status (Gaddis 2017a, b; Barlow and Lahey 2018; Crabtree et al. 2022). If having a distinctively Black-sounding name signals lower socioeconomic status, then the observed name penalty may be due to bias related to perceived socioeconomic status. This hypothesis would imply that those with names that convey a worse social background would have lower callback rates. But the data from Bertrand and Mullainathan (2019) demonstrate the opposite: across the set of Black-sounding names, mother's education and callback rates are actually negatively correlated.[21]

---

[20] The complete list of Black and ethnic immigrant names, along with their fluency ratings, can be found in Table A11 in the online Appendix.

[21] As defined in Bertrand and Mullainathan (2004), mother's education for a given first name is based on the percent of babies born under that first name in the state of Massachusetts between 1970 and 1986 whose mother had completed at least a high school degree.

Table 5—Examples of Black/Ethnic Immigrant Names and Callback Rates in Bertrand and Mullainathan (2004) and Oreopoulos (2011)

	Name difficulty	Percent callback
Bertrand and Mullainathan (2004)		
Black		
Ebony	−0.973	9.62
Kenya	−0.973	8.67
Leroy	−0.523	9.38
Aisha	1.148	2.22
Keisha	1.547	3.83
Lakisha	2.161	5.50
Correlation among Black applicants	−0.488 [0.040]	
Oreopoulos (2011)		
Indian		
Tara Singh	−0.603	10.29
Priyanka Kaur	2.557	7.61
Pakistani		
Sana Khan	0.392	8.82
Rabab Saeed	3.142	4.26
Chinese		
Min Liu	−0.671	11.34
Xiuying Zhang	1.511	7.42
Correlation among Ind/Pak/Chn applicants	−0.594 [0.002]	

Notes: Examples of Black and ethnic immigrant names are taken from publicly available replication data for Bertrand and Mullainathan (2004) and Oreopoulos (2011), respectively. The reported correlations are between name difficulty ratings and callback rates. The correlation in the top panel is computed using all Black job applicants in Bertrand and Mullainathan (2004), while the correlation in the bottom panel is calculated using all job applicants with ethnically Indian, Pakistani, and Chinese names in Oreopoulos (2011). p-values for correlations are in brackets.

And in terms of our study of name pronunciation, we find no significant relationship between name difficulty and mother's education across the set of African American names, so there is not strong support for the social background theory in our context.

The remainder of Table 5 shows some examples of ethnic names, taken from the study by Oreopoulos (2011), for applicants from India, Pakistan, and China, as well as their respective difficulty ratings and callback rates. Within Indian names, a relatively easy-to-pronounce name such as Tara Singh has a noticeably higher callback rate (10.29 percent) than that for a harder name like Priyanka Kaur (7.61 percent). There are similar examples of this relationship between name fluency and callback rates for those with Pakistani names (Sana Khan, callback rate of 8.82 percent, versus Rabab Saeed, callback rate of 4.26 percent) and Chinese names (Min Liu, callback rate of 11.3 percent, versus Xiuying Zhang, callback rate of 7.4 percent). Within each of these groups, the rating of name difficulty is negatively related to callback rates, with correlations between name difficulty and callback rates of −0.755, −0.588, and −0.338 for the sample of Indian, Pakistani, and Chinese names, respectively. Pooling these three ethnic groups together, the correlation coefficient is −0.594, which has a p-value of less than 0.01 for this sample of 24 observations.

Table 6—Name Fluency and Callback Rates: Experimental Data from Bertrand and Mullainathan (2004)

	All applicants				Black applicants	
	Callback (1)	Callback (2)	Callback (3)	Callback (4)	Callback (5)	Callback (6)
Black	−0.032		−0.018	−0.015		
	(0.006)		(0.006)	(0.006)		
Algorithm rating: First name		−0.017	−0.011	−0.012	−0.011	−0.014
		(0.004)	(0.005)	(0.005)	(0.003)	(0.003)
Observations	4,870	4,870	4,870	4,870	2,435	2,435
Control for name length	No	No	No	Yes	No	Yes
Control for gender	No	No	No	Yes	No	Yes
Control for résumé characteristics	No	No	No	Yes	No	Yes

Notes: The sample is derived from publicly available replication data for Bertrand and Mullainathan (2004). Columns 1–4 include all job applicants, while columns 5-6 focus on Black applicants. The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based subrating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

B. *Baseline Regression Results*

We now turn to our main regression results for the data derived from two prior audit studies, as presented in Tables 6 and 7.[22] In the first column of Table 6, we replicate and show the main finding of Bertrand and Mullainathan (2004): those with Black-sounding names have significantly lower callback rates than those with White-sounding names. Specifically, the marginal effect of the probit regression shows that the average callback rate is 3.2 percentage points lower for Black job applicants than for White applicants, representing a very large effect, as the average callback rates for Black and White applicants are only 6.45 and 9.65 percent, respectively.

In column 2, we estimate a probit regression that only includes the algorithmic rating of name fluency (once again, normalized for ease of interpretation) as an independent variable, and we find a statistically large and significant effect. A one standard deviation increase in name difficulty is correlated with a 1.7 percentage point decrease in the callback rate. Given that name fluency is correlated with race, we include both of these factors in the regression shown in column 3. The magnitude of the coefficient on the indicator variable for being a Black applicant is now reduced to -0.018 (p-value < 0.01), and the coefficient on the standardized measure of name difficulty is -0.011 (p-value $= 0.02$). This suggests that one important mechanism through which name-based racial discrimination works is the fluency of the name, as nearly *half* of the penalty for having a distinctively Black-sounding name is explained by name complexity. In column 4, we control for gender as well as a set of additional résumé characteristics used in Bertrand and Mullainathan (2004).[23] Name difficulty

---

[22] Full estimates on all covariates are reported in Tables A12 and A13 in the online Appendix.

[23] These résumé characteristics include education; number of jobs listed on résumé; a quadratic specification for years of work experience, honors received, volunteering, and military experience; and dummies for working in school, listing an email address on résumé, computer skills, and other special skills.

Table 7—Name Fluency and Callback Rates: Experimental Data from Oreopoulos (2011)

	All applicants				Ind/Pak/Chn applicants	
	Callback (1)	Callback (2)	Callback (3)	Callback (4)	Callback (5)	Callback (6)
Indian	−0.046		−0.036	−0.035		0.002
	(0.005)		(0.007)	(0.009)		(0.008)
Pakistani	−0.057		−0.049	−0.049		−0.015
	(0.007)		(0.008)	(0.009)		(0.012)
Chinese	−0.041		−0.038	−0.035		
	(0.005)		(0.006)	(0.009)		
Chinese Canadian	−0.053		−0.053	−0.050		
	(0.006)		(0.006)	(0.007)		
Greek	−0.031		−0.018	−0.018		
	(0.012)		(0.015)	(0.016)		
British	−0.024		−0.024	−0.023		
	(0.008)		(0.008)	(0.008)		
Algorithm rating: Full name		−0.014	−0.008	−0.007	−0.008	−0.007
		(0.003)	(0.004)	(0.004)	(0.003)	(0.004)
Observations	12,910	12,910	12,910	12,910	7,158	7,158
Control for name length	No	No	No	Yes	No	Yes
Control for gender	No	No	No	Yes	No	Yes
Control for résumé characteristics	No	No	No	Yes	No	Yes

Notes: The sample is derived from publicly available replication data for Oreopoulos (2011). Columns 1–4 include all job applicants, while columns 5–6 focus on applicants with ethnically Indian, Pakistani, and Chinese names. The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based subrating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

continues to be an important and significant factor in explaining callback rates, and the indicator variable for Black names remains negative and significant.

To more clearly distinguish discrimination due to name fluency from discrimination based on perceived race or ethnicity, we restrict our analysis to the sample of African American job applicants and show these results in the last two columns.[24] In column 5, we see that even within the sample of African American candidates, those with less fluent/familiar sounding names are less likely to receive callbacks. The coefficient of -0.011 (p-value < 0.01) implies that a one standard deviation increase in name difficulty leads to a decrease in the rate of callbacks by 1.1 percentage points, which represents a 20 percent change in the overall callback rate for Black applicants. After controlling for gender and additional résumé characteristics in column 6, the impact of name fluency actually increases (coefficient of -0.014) and remains statistically significant (p-value < 0.01).

Next, we perform a similar analysis using the replication data for Oreopoulos (2011). We begin by estimating a probit regression of callback rates, where the only independent variables are a set of indicators for various backgrounds based

---

[24] While there is a large range in pronunciation difficulty for the sample of Black-sounding names, the names in the White-sounding sample are all very easy to pronounce, so there is insufficient variation to conduct a separate analysis of these job applicants.

on ethnicity and national origin, and the results are shown in column 1 of Table 7. Relative to Canadians with English-sounding names, all other groups have lower callback rates, ranging from a decrease of 2.4 percentage points for applicants from Britain, to a decrease of 5.7 percentage points for Pakistani applicants. These are very large effects, given that the average callback rate for all job seekers is roughly 10 percent. In column 2, we see that name fluency matters, with a one standard deviation increase in name difficulty corresponding to a 1.4 percentage point decrease in the callback rate. When including controls for both ethnicity and name difficulty together, we see in column 3 that both remain significant, though the magnitudes of the coefficients on the Chinese, Pakistani, and Indian ethnicity indicators are decreased by 0.3, 0.8 and 1.0 percentage points, respectively, representing 7, 14, and 22 percent of the original effects. Once again, this implies that racial discrimination based on one's name is partly working through the difficulty of pronouncing (and potentially processing and remembering) that name. After controlling for gender and other résumé characteristics in column 4,[25] the coefficient on name difficulty now has a p-value of just over 0.1.[26]

We also restrict the sample to foreign applicants with ethnically Indian, Pakistani, or Chinese names, to see if name-based discrimination holds within these particular groups of job seekers. After controlling for ethnic background, the coefficient on name difficulty is negative and significant at the 1 (10) percent level without (with) control variables, as shown in column 5 (6) of Table 7. In Table A14 in the online Appendix, we show results for each of these ethnic groups separately. Name difficulty is negatively related to callback rates for each individual group, with a one standard deviation increase in difficulty associated with a decrease in callback rates of 0.6, 1.2, and 3.1 percentage points for Indian, Pakistani, and Chinese applicants, respectively, though these coefficients are not precisely estimated.[27]

C. *Potential Mechanisms*

In documenting the relationship between name difficulty and callback rates within a particular minority group, an important issue is understanding the mechanism through which this discrimination occurs. As discussed earlier, one possibility is that there is some general bias (conscious and/or subconscious) toward those with names that are hard to pronounce. But when we conduct separate analyses for candidates with high-quality and low-quality résumés, we find interesting differences. In the top panel of Table 8, when we divide the sample of Black job applicants by résumé quality (as determined by Bertrand and Mullainathan 2004),[28] we

---

[25] These résumé characteristics closely follow those employed in Oreopoulos (2011) and include quality of undergraduate programs, extracurricular activities, fluency in foreign languages, Master's degrees, additional credentials, prior work experience, list of references, and work authorization in Canada.

[26] Results from Table A13 in the online Appendix also show that the complexity of first and last names has a negative association with the likelihood of receiving a callback, but these coefficients are not statistically significant after the inclusion of the other control variables.

[27] In separate analyses discussed in online Appendix C, we test for potential gender differences in the name fluency effect, and we do not find any statistically significant differences for either of the audit studies.

[28] Specifically, Bertrand and Mullainathan (2004) subjectively categorize résumés as either high or low quality based on criteria such as labor market experience, career profile, existence of gaps in employment, and skills listed.

find somewhat mixed evidence of name-based discrimination toward Black applicants with high-quality résumés (columns 1–2), but we observe large and significant effects of name fluency on callback rates for Black applicants with low-quality résumés (columns 3–4). The difference in effects between the two groups of candidates is statistically significant (p-value $=$ 0.02 when comparing columns 2 and 4) and corroborates analogous findings from our analysis of PhD job candidates.

In Oreopoulos (2019), there is no subjective measure of quality provided in the data, so we partition the sample based on whether the candidate holds a Master's degree or not. As shown in the bottom panel of Table 8, there is greater discrimination due to name fluency among Indian, Pakistani, and Chinese applicants without a Master's degree, relative to those with a Master's degree. The point estimates are twice as large for those without a Master's degree relative to those with a Master's degree, though the differences in regression coefficients are not statistically significant.

Taken together, the heterogeneity in name effects is consistent with the idea that there are mental costs to processing and remembering names that are less familiar or more difficult to pronounce, and that these mental costs are only worth spending on higher-quality candidates. As a result, discrimination based on name fluency will mainly occur for those with lower-quality résumés, as employers develop quick heuristics to make decisions about these candidates, which they may not necessarily do for those with better résumés.

D. *Combined Audit Studies*

Finally, we stack the data from the two audit studies and run a series of regressions using the combined sample. While Bertrand and Mullainathan (2004) and Oreopoulos (2011) do not employ identical sets of controls, they have a number of variables in common, including gender, educational background, skills,[29] and name length, and we conduct pooled regressions using variables available in both datasets. Because Bertrand and Mullainathan (2004) only provide first names in their replication data, for consistency we only use the algorithmic rating for first names in Oreopoulos (2019), in spite of the fact that first and last names are available for the latter study.[30]

Column 1 of Table 9 reports the results from a specification that only includes an indicator variable equal to one if a job applicant is Black, Indian, Pakistani, or Chinese, and to zero otherwise. Relative to candidates with White-sounding names, those with a Black or ethnic immigrant name are 3.1 percentage points less likely to receive a callback, where the baseline callback rate is just over 9 percent. In column 2, where the regression only includes the algorithmic rating of first-name difficulty, we see that a one standard deviation increase in first-name difficulty decreases the likelihood of receiving a callback by 1.6 percentage points. When we

---

[29] Specifically, we include computer skills and other special skills (such as a certification degree or foreign language skills) from Bertrand and Mullainathan (2004) and language skills and extracurricular activities from Oreopoulos (2011).

[30] Nonetheless, regressions where we use the first-name difficulty from Bertrand and Mullainathan (2019) and overall name difficulty from Oreopoulos (2019) yield similar results.

Table 8—Name Fluency and Callback Rates by Résumé Quality: Experimental Data from Bertrand and Mullainathan (2004) and Oreopoulos (2011)

	Black applicants			
	High-quality résumé		Low-quality résumé	
	Callback (1)	Callback (2)	Callback (3)	Callback (4)
Bertrand and Mullainathan (2004)				
Algorithm rating: First name	−0.004	−0.009	−0.017	−0.019
	(0.005)	(0.004)	(0.003)	(0.003)
Observations	1,223	1,223	1,212	1,208
Control for name length	No	Yes	No	Yes
Control for gender	No	Yes	No	Yes
Control for résumé characteristics	No	Yes	No	Yes

	Ind/Pak/Chn applicants			
	Master's		No Master's	
	Callback (5)	Callback (6)	Callback (7)	Callback (8)
Oreopoulos (2011)				
Algorithm rating: Full name	−0.005	−0.003	−0.008	−0.008
	(0.008)	(0.010)	(0.003)	(0.005)
Observations	1,129	1,129	6,029	6,029
Control for name length	No	Yes	No	Yes
Control for ethnicity	No	Yes	No	Yes
Control for gender	No	Yes	No	Yes
Control for résumé characteristics	No	Yes	No	Yes

Notes: The samples are derived from publicly available replication data for Bertrand and Mullainathan (2004) and Oreopoulos (2011). The top panel focuses on Black job applicants, while the bottom panel focuses on job applicants with ethnically Indian, Pakistani, and Chinese names. The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. Résumé quality is determined based on a subjective measure in Bertrand and Mullainathan (2004) and whether one holds a Master's degree in Oreopoulos (2011). The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based subrating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

run a specification that includes both an indicator for Black/ethnic immigrant and a measure of name difficulty in column 3, the racial/ethnic coefficient (−0.019) is reduced by approximately 40 percent compared to its counterpart in column 1, while the name difficulty coefficient remains large (−0.010) and statistically significant. In column 4, we add a set of common control variables, but the coefficients on the racial/ethnic indicator and name difficulty are not greatly altered.[31]

---

[31] To further assess how much of the ethnic name penalty can be explained by name fluency, we analyze the subset of low-quality applicants (for each of the separate datasets as well as the pooled data) and run regressions similar to those in the top panel of Table 9, and these results are reported in Table A16 in the online Appendix. Among the sample of applicants with low-quality résumés in Bertrand and Mullainathan (2004), nearly all of the ethnic name penalty is explained by name fluency, while for the sample of applicants without a Master's degree in Oreopoulos (2011), name fluency explains 23, 14, and 7 percent of the ethnic name penalty for Indians, Pakistanis, and Chinese, respectively. For the pooled sample of low-quality applicants, name difficulty explains roughly 50 percent of the ethnic name penalty.

Table 9—Name Fluency and Callback Rates: Combining Experimental Data from Bertrand and Mullainathan (2004) and Oreopoulos (2011)

	Callback (1)	Callback (2)	Callback (3)	Callback (4)
All applicants				
Black/immigrant (Ind/Pak/Chn)	−0.031		−0.019	−0.023
	(0.004)		(0.005)	(0.006)
Algorithm rating: First name		−0.016	−0.010	−0.008
		(0.002)	(0.003)	(0.004)
Observations	17,780	17,780	17,780	17,780
Control for name length	No	No	No	Yes
Control for gender	No	No	No	Yes
Control for education	No	No	No	Yes
Control for skills	No	No	No	Yes

	All	High-quality résumé	Low-quality résumé
	Callback (5)	Callback (6)	Callback (7)
Black/immigrant (Ind/Pak/Chn)			
Algorithm rating: First name	−0.007	0.001	−0.010
	(0.004)	(0.007)	(0.004)
Observations	9,593	2,352	7,241
Control for name length	Yes	Yes	Yes
Control for gender	Yes	Yes	Yes
Control for education	Yes	Yes	Yes
Control for skills	Yes	Yes	Yes

Notes: The sample combines publicly available replication data for Bertrand and Mullainathan (2004) and Oreopoulos (2011). The top panel includes all job applicants from both studies, while the bottom panel focuses specifically on Black and ethnic immigrant applicants. The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. Résumé quality is determined based on a subjective measure in Bertrand and Mullainathan (2004) and whether one holds a Master's degree in Oreopoulos (2011). The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based subrating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

In the bottom panel of Table 9, we restrict the sample to Black job candidates and immigrants from India, Pakistan, and China. Even with this set of job applicants, we see a significant effect of name difficulty on job opportunities, with a one standard deviation increase in difficulty corresponding to a 0.7 percentage point decrease in the job callback rate. When we further divide this sample of non-White candidates into those with high-quality résumés and low-quality résumés based on the criteria that we outline earlier for each dataset, the coefficient on name difficulty is significantly larger for those with low-quality résumés (−0.01 with a p-value of 0.02, as shown in column 7) than for those with high-quality résumés (coefficient of −0.001 and statistically insignificant, as shown in column 6).

We also test to see whether name fluency effects are different across the two datasets. Although there is no difference in the overall name fluency effect between the two samples, when focusing on the low-quality résumés for the restricted sample of Black/immigrant applicants, there is a statistically significant difference (p-value < 0.01) in the size of the coefficients, with a much larger penalty for having a difficult-to-pronounce name within the sample of Black applicants than within the sample of ethnic Indian, Pakistani, and Chinese immigrants.

IV. Conclusions

This paper analyzes two recent cohorts of economics PhD candidates and finds strong evidence for labor market discrimination against individuals with names that are hard to pronounce. A one standard deviation increase in the difficulty of a candidate's full name lowers the likelihood of obtaining an academic (tenure-track) job by somewhere between 3 and 12 (1 and 6) percentage points and results in placement in an institution that is ranked about 70−80 spots lower on RePEc's research productivity ranking. We obtain strong and consistent results across three ways of measuring name fluency, and the results persist after the inclusion of a comprehensive set of control variables.

We also employ experimental data from prior audit studies and find that labor market discrimination based on one's perceived race or ethnicity can be partially explained by the difficulty of a job applicant's name. Using randomly assigned White-sounding and Black-sounding names (Bertrand and Mullainathan 2004), we find that even after accounting for perceived race, applicants with more difficult-to-pronounce names receive fewer callbacks, and name difficulty explains almost half of the racial gap. Within the set of Black applicants, those with more difficult names are less likely to get called. We obtain similar results using applicants to job postings in Canada (Oreopoulos 2011). Once again, controlling for ethnic background, having a less fluent name corresponds to a lower probability of being called back for a job, and incorporating name difficulty in the empirical analysis reduces the ethnic gaps by 7 to 22 percent. And within the sample of ethnically Indian, Pakistani, and Chinese applicants, name difficulty is negatively related to callback rates.

For all three data sources, name penalties are strongest and most consistently found for weaker job candidates, suggesting that there are mental costs to processing and remembering less fluent names, and that these mental costs are only worth spending on higher quality candidates. This mechanism of discrimination differs from those studied in most prior research, which has primarily focused on the role of preferences or statistical discrimination in explaining racial disparities in labor market outcomes.

The results of this paper have important and wide-ranging implications. In designing audit studies to test for racial and ethnic discrimination based on names, researchers should consider choosing names that are roughly equivalent in terms of pronunciation difficulty, and name fluency should be included as a control variable in all regression analysis. Our results can also be applied to many settings. While we find strong evidence for the existence of discrimination based on name pronunciation across several different types of labor markets, name fluency could matter in other contexts such as housing markets, access to receiving health care, or teachers' differential treatment of students.

What are some potential ways to combat this bias? One practical implication for search committees and recruiters is to be cognizant of another potential source of bias that exists in the hiring process. While it is impossible to entirely eliminate all sources of bias and labor market discrimination, our analysis suggests that with sufficient attention and effort, the bias against difficult-to-pronounce names can be minimized. Perhaps individuals who are reading job applications could use an online pronunciation tool or ask a colleague about how to pronounce an unfamiliar name. A more time-intensive intervention that may be feasible in some labor markets would be to have an administrator redact names of job applicants before hiring committee members review them to select those that will receive an initial interview.[32] Finally, phonetic spelling instructions may be helpful in reducing cognitive costs of employers who have difficulty figuring out how to process and/or pronounce job applicants' names. Studies that test for the effectiveness of these types of interventions would be a fruitful avenue for future research.

REFERENCES

Alter, Adam L., and Daniel M. Oppenheimer. 2006. "Predicting Short-Term Stock Fluctuations by Using Processing Fluency." *Proceedings of the National Academy of Sciences* 103 (24): 9369–72.

Ang, Chee Siang, Panayiotis Zaphiris, and Shumaila Mahmood. 2007. "A Model of Cognitive Loads in Massively Multiplayer Online Role Playing Games." *Interacting with Computers* 19 (2): 167–79.

Athey, Susan, Lawrence F. Katz, Alan B. Krueger, Steven Levitt, and James Poterba. 2007. "What Does Performance in Graduate School Predict? Graduate Economics Education and Student Outcomes." *American Economic Review* 97 (2): 512–18.

Baert, Stijn. 2018. "Hiring Discrimination: An Overview of (Almost) All Correspondence Experiments since 2005." In *Audit Studies: Behind the Scenes with Theory, Method, and Nuance*, edited by S. Michael Gaddis, 63–77. Cham, Switzerland: Springer.

Bannert, Maria. 2002. "Managing Cognitive Load—Recent Trends in Cognitive Load Theory." *Learning and Instruction* 12 (1): 139–46.

Barlow, M. Rose, and Joanna N. Lahey. 2018. "What Race Is Lacey? Intersecting Perceptions of Racial Minority Status and Social Class." *Social Science Quarterly* 99 (5): 1680–98.

Behaghel, Luc, Bruno Crépon, and Thomas Le Barbanchon. 2015. "Unintended Effects of Anonymous Résumés." *American Economic Journal: Applied Economics* 7 (3): 1–27.

Bertrand, Marianne, and Sendhil Mullainathan. 2004. "Are Emily and Greg More Employable than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination." *American Economic Review* 94 (4): 991–1013.

Bertrand, Marianne, and Sendhil Mullainathan. 2019. *Data and Code for: "Are Emily and Greg More Employable than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination."* Nashville, TN: American Economic Association; distributed by Inter-university Consortium for Political and Social Research, Ann Arbor, MI. https://doi.org/10.3886/E116023V1.

Biavaschi, Costanza, Corrado Giulietti, and Zahra Siddique. 2017. "The Economic Payoff of Name Americanization." *Journal of Labor Economics* 35 (4): 1089–116.

Bojar, Ondřej, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. "Results of the WMT16 Metrics Shared Task." In *Proceedings of the First Conference on Machine Translation, Vol. 2, Shared Task Papers*, edited by Ondřej Bojar, Christan Buck, Rajen Chatterjee, Liane Guillou, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, et al, 199–231. Berlin, Germany: Association for Computational Linguistics.

Carpenter, Christopher S. 2007. "Revisiting the Income Penalty for Behaviorally Gay Men: Evidence from NHANES III." *Labour Economics* 14 (1): 25–34.

Chapman, Elizabeth N., Anna Kaatz, and Molly Carnes. 2013. "Physicians and Implicit Bias: How Doctors May Unwittingly Perpetuate Health Care Disparities." *Journal of General Internal Medicine* 28 (11): 1504–10.

---

[32] While making résumés anonymous may decrease the cognitive costs of processing and remembering difficult names, recent research has shown that such practice may have other consequences, as removing names could make it harder to discuss candidates in general (Krause, Rinne and Zimmermann 2012; Behaghel, Crépon and Le Barbanchon 2015).

Coupé, Tom. 2003. "Revealed Performances: Worldwide Rankings of Economists and Economics Departments, 1990–2000." *Journal of the European Economic Association* 1 (6): 1309–45.

Crabtree, Charles, S. Michael Gaddis, John B. Holbein, and Edvard Nergår Larsen. 2022. "Racially Distinctive Names Signal Both Race/Ethnicity and Social Class." *Sociological Science* 9 (18): 454–72.

Cross, Harry, Genevieve M. Kenney, Jane Mell, and Wendy Zimmermann. 1990. *Employer Hiring Practices: Differential Treatment of Hispanic and Anglo Job Seekers*. Washington, DC: Urban Institute Press.

Darolia, Rajeev, Cory Koedel, Paco Martorell, Katie Wilson, and Francisco Perez-Arce. 2016. "Race and Gender Effects on Employer Interest in Job Applicants: New Evidence from a Resume Field Experiment." *Applied Economics Letters* 23 (12): 853–56.

Fryer, Roland G., Jr., and Steven D. Levitt. 2004. "The Causes and Consequences of Distinctively Black Names." *Quarterly Journal of Economics* 119 (3): 767–805.

Gaddis, S. Michael. 2017a. "How Black Are Lakisha and Jamal? Racial Perceptions from Names Used in Correspondence Audit Studies." *Sociological Science* 4 (19): 469–89.

Gaddis, S. Michael. 2017b. "Racial/Ethnic Perceptions from Hispanic Names: Selecting Names to Test for Discrimination." *Socius: Sociological Research for a Dynamic World* 3: 2378023117737193.

Ge, Qi, Stephen Wu, and Chenyu Zhou. 2021a. *Data for: "Sharing Common Roots: Student-Graduate Committee Matching and Job Market Outcomes."* Hoboken, NJ: Wiley Publishing. https://doi.org/10.1002/soej.12531 (accessed December 21, 2021).

Ge, Qi, Stephen Wu, and Chenyu Zhou. 2021b. "Sharing Common Roots: Student-Graduate Committee Matching and Job Market Outcomes." *Southern Economic Journal* 88 (2): 828–56.

Goldin, Claudia, and Cecilia Rouse. 2000. "Orchestrating Impartiality: The Impact of 'Blind' Auditions on Female Musicians." *American Economic Review* 90 (4): 715–41.

Hamermesh, Daniel S., and Jeff E. Biddle. 1994. "Beauty and the Labor Market." *American Economic Review* 84 (5): 1174–94.

Hansen, Jesper Asring, and Lars Tummers. 2020. "A Systematic Review of Field Experiments in Public Administration." *Public Administration Review* 80 (6): 921–31.

Kahneman, Daniel, Stewart Paul Slovic, Paul Slovic, and Amos Tversky. 1982. *Judgment under Uncertainty: Heuristics and Biases*. Cambridge, UK: Cambridge University Press.

Kang, Jerry, Mark Bennett, Devon Carbado, Pam Casey, Nilanjana Dasgupta, David Faigman, Rachel Godsil, et al. 2011. "Implicit Bias in the Courtroom." *UCLA Law Review* 59: 1124.

Kang, Sonia K., Katherine A. DeCelles, András Tilcsik, and Sora Jun. 2016. "Whitened Résumés: Race and Self-Presentation in the Labor Market." *Administrative Science Quarterly* 61 (3): 469–502.

Krause, Annabelle, Ulf Rinne, and Klaus F. Zimmermann. 2012. "Anonymous Job Applications of Fresh PhD Economists." *Economics Letters* 117 (2): 441–44.

Krueger, Alan B., and Stephen Wu. 2000. "Forecasting Job Placements of Economics Graduate Students." *Journal of Economic Education* 31 (1): 81–94.

Laham, Simon M., Peter Koval, and Adam L. Alter. 2012. "The Name-Pronunciation Effect: Why People Like Mr. Smith More than Mr. Colquhoun." *Journal of Experimental Social Psychology* 48 (3): 752–56.

Neumark, David. 2018. "Experimental Research on Labor Market Discrimination." *Journal of Economic Literature* 56 (3): 799–866.

Neumark, David, Roy J. Bank, and Kyle D. Van Nort. 1996. "Sex Discrimination in Restaurant Hiring: An Audit Study." *Quarterly Journal of Economics* 111 (3): 915–41.

Neumark, David, Ian Burn, and Patrick Button. 2019. "Is It Harder for Older Workers to Find Jobs? New and Improved Evidence from a Field Experiment." *Journal of Political Economy* 127 (2): 922–70.

Newman, Eryn J., Mevagh Sanson, Emily K. Miller, Adele Quigley-McBride, Jeffrey L. Foster, Daniel M. Bernstein, and Maryanne Garry. 2014. "People with Easier to Pronounce Names Promote Truthiness of Claims." *PLoS One* 9 (2): e88671.

Nordstrom, Cynthia R., Karen B. Williams, and James M. LeBreton. 1996. "The Effect of Cognitive Load on the Processing of Employment Selection Information." *Basic and Applied Social Psychology* 18 (3): 305–18.

Nunley, John M., Adam Pugh, Nicholas Romero, and R. Alan Seals. 2015a. *Data for: "Racial Discrimination in the Labor Market for Recent College Graduates: Evidence from a Field Experiment."* Berlin, Germany: De Gruyter. http://doi.org/10.1515/bejeap-2014-0082 (accessed January 5, 2023).

Nunley, John M., Adam Pugh, Nicholas Romero, and R. Alan Seals. 2015b. “Racial Discrimination in the Labor Market for Recent College Graduates: Evidence from a Field Experiment.” *B. E. Journal of Economic Analysis & Policy* 15 (3): 1093–125.

Oreopoulos, Philip. 2011. “Why Do Skilled Immigrants Struggle in the Labor Market? A Field Experiment with Thirteen Thousand Resumes.” *American Economic Journal: Economic Policy* 3 (4): 148–71.

Oreopoulos, Philip. 2019. *Data and Code for: “Why Do Skilled Immigrants Struggle in the Labor Market? A Field Experiment with Thirteen Thousand Resumes.”* Nashville, TN: American Economic Association; distributed by Inter-university Consortium for Political and Social Research, Ann Arbor, MI. https://doi.org/10.3886/E114770V1.

Paas, Fred, and John Sweller. 2012. “An Evolutionary Upgrade of Cognitive Load Theory: Using the Human Motor System and Collaboration to Support the Learning of Complex Cognitive Tasks.” *Educational Psychology Review* 24 (1): 27–45.

Pierné, Guillaume. 2013. “Hiring Discrimination Based on National Origin and Religious Closeness: Results from a Field Experiment in the Paris Area.” *IZA Journal of Labor Economics* 2: 4.

Plass, Jan L., Roxana Moreno, and Roland Brünken, eds. 2010. *Cognitive Load Theory*. Cambridge, UK: Cambridge University Press.

Popović, Maja. 2018. “Complex Word Identification Using Character N-grams.” In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, edited by Joel Tetreault, Jill Burstein, Ekaterina Kochmar, Claudia Leacock, and Helen Yannakoudakis, 341–48. New Orleans: Association for Computational Linguistics.

Song, Hyunjin, and Norbert Schwarz. 2008. “If It’s Hard to Read, It’s Hard to Do: Processing Fluency Affects Effort Prediction and Motivation.” *Psychological Science* 19 (10): 986–88.

Staats, Cheryl. 2016. “Understanding Implicit Bias: What Educators Should Know.” *American Educator* 39 (4): 29–33.

Sullivan, Ryan S., Alissa Dubnicki, and Donald H. Dutkowsky. 2018. “Research, Teaching, and ‘Other’: What Determines Job Placement of Economics PhDs?” *Applied Economics* 50 (32): 3477–92.

Sweller, John. 2011. “Cognitive Load Theory.” In *Psychology of Learning and Motivation*, Vol. 55, edited by Jose P. Mestre and Brian H. Ross, 37–76. Cambridge, MA: Academic Press.

Turner, Margery Austin, Michael Fix, and Raymond J. Struyk. 1991. *Opportunities Denied, Opportunities Diminished: Racial Discrimination in Hiring*. Washington, DC: Urban Institute Press.

Tversky, Amos, and Daniel Kahneman. 1974. “Judgment under Uncertainty: Heuristics and Biases: Biases in Judgments Reveal Some Heuristics of Thinking under Uncertainty.” *Science* 185 (4157): 1124–31.

Wilson, Theresa, and Stephan Raaijmakers. 2008. “Comparing Word, Character, and Phoneme N-grams for Subjective Utterance Recognition.” Paper presented at Interspeech 2008, Brisbane, Australia.

Wright, Bradley R. E., Michael Wallace, John Bailey, and Allen Hyde. 2013. “Religious Affiliation and Hiring Discrimination in New England: A Field Experiment.” *Research in Social Stratification and Mobility* 34: 111–26.

Wu, Stephen, and Qi Ge. 2024. *Data and Code for: “How Do You Say Your Name? Difficult-to-Pronounce Names and Labor Market Outcomes.”* Nashville, TN: American Economic Association; distributed by Inter-university Consortium for Political and Social Research, Ann Arbor, MI. https://doi.org/10.3886/E188001V1.

Online Appendix for "How Do You Say Your Name? Difficult-To-Pronounce Names and Labor Market Outcomes"

Qi Ge* Stephen Wu†

Abstract: This online appendix contains additional empirical analyses complementing the results and discussions presented in the main text. In Appendix A, we perform robustness checks on our baseline findings using observational data from the academic labor market. In Appendix B, we explore the possibility that the uniqueness or commonality of names may affect job outcomes. In Appendix C, we test for heterogeneous effects by gender using experimental data from Bertrand and Mullainathan (2004) and Oreopoulos (2011). In Appendix D, we investigate labor market effects of name fluency using data from Nunley et al. (2015). Lastly, in Appendix E, we include the full instructions for our name fluency surveys.

---

*Department of Economics, Vassar College, Poughkeepsie, NY 12604, *qige@vassar.edu*.

†Department of Economics, Hamilton College, Clinton, NY 13323, *swu@hamilton.edu*.

Appendix A. Robustness Checks

In this section, we present a number of robustness checks on our baseline findings using observational data from the academic labor market. We first consider an alternative placement quality measure based on the raw RePEc ranking that excludes private sector and non-tenure track academic placements. Tobit estimates based on this alternative RePEc measure are reported in Table A2 in the Online Appendix and are similar in direction and statistical significance to the full sample results with imputed RePEc values as shown in Tables 2 and 3. Relative to the original imputed RePEc rankings, the relevant coefficients on name fluency measures are smaller in magnitude for the timing measure (51 vs. 83) and the subjective rating (28 vs. 82) but larger for the algorithmic rating (79 vs. 67).

Next, to assess the robustness of our specifications on placement quality, we estimate multinomial logit regressions for different types of placements and present the estimates in Table A3 in the Online Appendix. We observe that relative to the reference group placement type of government or think tank jobs, the coefficient on name difficulty is significantly negative for being placed into academia, and this result is consistent across different name fluency measures.[1] On the other hand, in separate specifications reported in Table A4 in the Online Appendix, when we further decompose academic job types and set the baseline category as visiting/postdoc, the coefficient on name difficulty for the tenure track category is not significant relative to the baseline. Taken together, this suggests that name fluency impacts the likelihood of being placed into academia relative to industry or government jobs, but does not affect the probability of obtaining a tenure track job, conditional on being placed in academia.

As an additional check on the robustness of the results on placement quality, we estimate an ordered probit model using categories of the imputed RePEc ranking of job placements as the outcome of interest. Given the ordinal nature of RePEc rankings, we categorize

---

[1] The difference between the coefficients for academic and industry positions is statistically significant at the 10%, 1%, and 1% levels for name fluency measures based on algorithmic ratings, pronunciation time, and subjective ratings, respectively.

the ranking of imputed RePEc productivity index into the following five categories for the ordered probit model: 1) RePEc ≤ 50; 2) $50 <$ RePEc ≤ 200; 3) $200 <$ RePEc ≤ 400; 4) $400 <$ RePEc ≤ 800; and 5) RePEc $= 1,000$. The estimates on name fluency measures, as presented in Table A5 in the Online Appendix, are qualitatively similar to our main findings and again suggest that candidates with harder-to-pronounce names tend to be placed in institutions with lower research productivity.

A concern discussed in the main text is that name changes may be endogenous. For example, students who have advisors and committee members from the same country might be less likely to feel the need to Americanize/Anglicize their (first) names. Ge et al. (2021) document a beneficial impact of student-graduate committee matching, in the form of country of origin and native language, on students' initial placement outcomes in the economics PhD job market, which could lead to a downward bias in the estimate of the magnitude of the name fluency effect. To account for this possibility, we re-estimate our baseline specifications and add controls for student-graduate committee matching based on country (U.S. vs. non-U.S.) or native language (English vs. other),[2] and the resulting estimates, as reported in Table A6, remain identical to those in Tables 2 and 3. The decision of whether or not to change one's last name after marriage may also be endogenous, though separate analysis by gender does not reveal any differences in the effects of name fluency. As shown in Table A7, we find similarly sized effects for the sample of male job market candidates (where changing last names is much less common than for females). Furthermore, as seen in Table A8, our results continue to hold when we exclude all candidates with ethnically Chinese names, a group for which individuals are particularly likely to adopt Americanized first names.

Another potential concern is that difficult-to-pronounce names are concentrated in a few countries, and the lack of success that individuals from these countries have in finding prestigious academic jobs is not necessarily linked to their names but from more general

---

[2] Following Ge et al. (2021), we code "country match" as being equal to one when at least one of the student's committee members went to an undergraduate institution in the same country as the student's undergraduate institution. Similarly, we code "language match" as being equal to one when a student's country of origin has the same official language as that of at least one of the committee members.

discrimination due to national origin. All regressions shown in our tables have controlled for the region of one's undergraduate school, but we have also estimated specifications which include a full set of individual country effects, and the results, as presented in Table A9 in the Online Appendix, are largely the same. In addition, we have also run separate regressions for different regions, though the statistical power is reduced in regions with few observations. In general, we observe that the effects of name fluency on placement types and quality are not driven by a particular region of undergraduate degree, as the magnitudes of the effects are large and significant for several different regions.

Appendix B. Common Names

We also explore the possibility that the uniqueness or commonality of names may affect job outcomes. It is likely that those with very common names could be at a disadvantage because they do not stand out from other candidates. Because pronunciation difficulty is likely negatively correlated with commonality of names, our estimates of the name fluency effect might be underestimated. To alleviate this concern, we augment our baseline specifications by controlling for having a common first name or common last name. Due to data constraints, we focus on common names in the U.S. Specifically, we code someone as having a very common name if their first name is among the 50 most common female first names or the 50 most common male first names according to the 1990 U.S. Census, and having a very common last name if their last name is among the 50 most common surnames according to the 2010 U.S. Census.[3]

We present the resulting estimates in Table A10 in the Online Appendix. As shown in columns 1-3 that focus on the full sample of job market candidates, none of the variables for name commonality (i.e., indicator for common first name, indicator for common last name, and their interaction) is statistically significant, and their inclusion does not impact

---

[3] The 1990 and 2010 U.S. Census data respectively represent the most recent data sources for tabulations on common first and last names.

the magnitude or significance of the name difficulty coefficient in any of our regressions. In addition, since the data sources for our common name analysis are based on the U.S. Census, we also conduct a separate analysis for the sample of job market candidates who are from U.S. and Canada. As shown in columns 4-6, the results on placement types and quality as well as the coefficients on common name indicators are qualitatively similar, though larger in magnitude.

Appendix C. Heterogeneous Effects by Gender in Audit Study Data

In this section, we explore potential gender differences in the name fluency effect in the experimental data from Bertrand and Mullainathan (2004) and Oreopoulos (2011). For each of these data sources, we divide the sample by gender and re-estimate our main probit regressions that relate callback rates to algorithmic ratings of name difficulty. All specifications include controls for name length, race/ethnicity, and resume characteristics and use standard errors that are clustered at the job advertisement level.

Table A15 in the Online Appendix reports our estimates for the name fluency effect by gender, with the top and bottom panels focusing on data from Bertrand and Mullainathan (2004) and Oreopoulos (2011), respectively. Columns 1-2 and 5-6 are based on the full sample of each data set, while columns 3-4 and 7-8 focus on the sample of Black job candidates and immigrants from India, Pakistan, and China, respectively. Although the point estimates on the name difficulty measure are somewhat larger and more statistically significant for female applicants across both data sets, the magnitudes of the impacts are not statistically different between the two groups.

In addition, we also compare the algorithmic ratings of names between male and female job applicants and find that there is no consistent and systematic relationship between fluency of names and gender across the two audit study data sources. Specifically, we find that female applicants, on average, have significantly more difficult (first) names than their male counterparts in Bertrand and Mullainathan (2004), while the opposite pattern holds for Oreopoulos (2011).

Overall, we do not find support for significant gender differences in the effect of name fluency based on prior audit study data. Our findings here also support the results in Table A7 in the Online Appendix that document indistinguishable name fluency effects between male and female economics PhD job market candidates.

Appendix D. Experimental Data from Nunley et al. (2015)

As an additional test, we also investigate labor market effects of name fluency using data from Nunley et al. (2015), who perform an audit study to examine racial discrimination in the labor market for recent college graduates. Specifically, Nunley et al. (2015) create fictitious and identical resumes for college-educated entry level job applicants who are randomly assigned one of the eight distinctively White-sounding or African American-sounding names. Similar to Bertrand and Mullainathan (2004) and Oreopoulos (2011), Nunley et al. (2015) also focus on callback rates as their main outcome variable of interest.

Analogous to our analysis of the other two audit studies, we first estimate a probit model of callback rates on implied race of the applicants and report the results in column 1 of Table A17 in the Online Appendix. Consistent with Nunley et al. (2015), we find that the callback rates for job applicants with African American-sounding names are 2.8 percentage points lower compared to those with White-sounding names. When applying our name fluency algorithm to the fictitious first and last names employed in Nunley et al. (2015), we observe in column 2 that the standardized algorithmic name rating is negatively and significantly correlated with callback rates.

In column 3, we include both race and name difficulty measures and find that the magnitude of the coefficient on being a Black applicant is reduced to -0.024 (p-value < 0.01), representing an approximately 15 percent decrease in the racial penalty estimated in column 1. Similar to our findings discussed in Section III.B, this implies that racial discrimination based on one's name partly works through the difficulty of pronouncing (and potentially processing and remembering) that name.

When controlling for name length, gender, as well as additional resume characteristics used in Nunley et al. (2015),[4] we show in column 4 that name difficulty is an important and significant factor in explaining the callback rates, with the coefficient now marginally significant at the 10 percent level, and the indicator variable for Black names remains negative and significant.

It is worth noting that the data from Nunley et al. (2015) uses only eight unique names (two for each gender-race combination), which are far fewer than the number used by Bertrand and Mullainathan (36) or Oreopoulos (44). Despite this important drawback and the resulting limited statistical power, our analysis of this additional experimental data confirms that name complexity is negatively related to the probability of receiving a callback and that an important channel for explaining name-based racial discrimination is through the fluency of one's name. These results are thus consistent with our main findings discussed in Section III.

Appendix E. Instructions for Name Fluency Surveys

Thank you for agreeing to assist with research projects related to the pronunciation of names. I have designed a set of Qualtrics surveys which have a series of names for you to pronounce.

1. Before you start a particular survey, start an audio recording of yourself. Then, you will see a series of names for you to pronounce, with one name per screen. Read through

---

[4] The set of resume characteristics includes college attended, academic major, grade point average, honor's distinction, employment status, socioeconomic status of the applicant's address, and dummies for month and city.

the name, and then click the arrow to advance to the next screen to see the next name. Continue to repeat this until you have finished the survey. You may then stop the recording and save it. You will repeat this process for all of the different groups of names, though you may wish to do break up your work across several different times in the day or the week to complete the work.

2. Please complete a particular group in one sitting without taking any breaks in between. Once you complete that group, then feel free to take as long of a break as you need until you start the next survey, but again, please do not take breaks once you have started a new survey until you complete that one. Names will be separated in groups of approximately 50 (with some groups listed as first names and some groups listed as last names), so perhaps you may want to do a bunch at one time, with short breaks in between each of the individual surveys. Then, you can come back and do another chunk of them at another day/time when you are free.

3. If you are unsure of how to pronounce a particular name, simply do your best to make a guess or sound it out before you click the arrow to advance to the next screen. You should not search the internet to hear a recording of the name, but simply make an attempt at pronouncing it.

4. It is possible that you may see some names that are duplicates or are very similar to other names in one of the surveys, but please pronounce each of the names you see on the screen even if you think you have seen that name before.

5. Please complete each survey only one time. To make sure that you do every survey only once, take careful notes about which ones you have completed and which ones you still need to complete. The most logical way would be to complete the surveys in numerical order (perhaps starting with the first names and then the last names).

References

Bertrand, M., and Mullainathan, S. (2004). “Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination.” *American Economic Review*, *94*(4), 991–1013.

Ge, Q., Wu, S., and Zhou, C. (2021). “Sharing common roots: Student-graduate committee matching and job market outcomes.” *Southern Economic Journal*, *88*(2), 828–856.

Nunley, J. M., Pugh, A., Romero, N., and Seals, R. A. (2015). “Racial discrimination in the labor market for recent college graduates: Evidence from a field experiment.” *B.E. Journal of Economic Analysis & Policy*, *15*(3), 1093–1125.

Oreopoulos, P. (2011). “Why do skilled immigrants struggle in the labor market? A field experiment with thirteen thousand resumes.” *American Economic Journal: Economic Policy*, *3*(4), 148–71.

Table A1: Name Fluency and Placement Outcomes: Alternative Algorithm Rating

	(1) Academia	(2) Academia	(3) TT	(4) TT	(5) RePEc_Imputed	(6) RePEc_Imputed
Alternative Algorithm Rating: Full Name	-0.040		-0.019		82.771	
	(0.016)		(0.017)		(31.479)	
Alternative Algorithm Rating: First Name		-0.037		-0.018		56.701
		(0.017)		(0.018)		(32.596)
Alternative Algorithm Rating: Last Name		-0.020		-0.009		68.384
		(0.019)		(0.019)		(36.238)
Observations	1,469	1,469	1,499	1,499	1,510	1,510
Control for Name Length	Yes	Yes	Yes	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The coefficients in columns 1-4 are marginal effects of probit regressions. The dependent variable in columns 1-2 (3-4) is a dichotomous variable for being placed in an academic (tenure track) position. Columns 5-6 are estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of $1,000$, the highest (worst) ranking. All tobit regressions are censored with an upper limit of $1,000$. The alternative algorithm rating for name pronunciation difficulty is based on an arithmetic average of the letter-based and phoneme-based sub-rating schemes. Robust standard errors are in parentheses.

Table A2: Name Fluency and Placement Quality: Tobit Estimates – Raw RePEc Ranking

	(1) RePEc	(2) RePEc	(3) RePEc
Algorithm Rating: Full Name	79.298		
	(24.802)		
Pronunciation Time: Full Name		51.069	
		(26.546)	
Subjectively Difficult: Full Name			28.278
			(49.645)
Observations	910	910	910
Control for Name Length	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes

Notes: The dependent variable across all specifications is the RePEc ranking of the institution of initial job placement, where individuals obtaining private sector jobs are excluded from the sample. All specifications are estimated using a tobit model censored with an upper limit of $1,000$. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. The pronunciation time rating is a survey-based measure that records the median time it takes individuals to pronounce a name. The subjective difficulty rating is based on individuals' independent subjective assessments. Robust standard errors are in parentheses.

Table A3: Name Fluency and Placement Types: Multinomial Logit Estimates

	(1) Academia	(2) Industry
Algorithm Rating: Full Name	-0.217	-0.161
	(0.101)	(0.120)
Observations	1,510	1,510
Control for Name Length	Yes	Yes
Other Controls	Yes	Yes
Subfield/Program FE	Yes	Yes
Region/JM Cycle FE	Yes	Yes

	(3) Academia	(4) Industry
Pronunciation Time: Full Name	-0.273	0.113
	(0.102)	(0.120)
Observations	1,510	1,510
Control for Name Length	Yes	Yes
Other Controls	Yes	Yes
Subfield/Program FE	Yes	Yes
Region/JM Cycle FE	Yes	Yes

	(5) Academia	(6) Industry
Subjectively Difficult: Full Name	-0.411	0.195
	(0.190)	(0.229)
Observations	1,510	1,510
Control for Name Length	Yes	Yes
Other Controls	Yes	Yes
Subfield/Program FE	Yes	Yes
Region/JM Cycle FE	Yes	Yes

Notes: Each panel is estimated using a separate multinomial logit model with the dependent variable being a categorical variable capturing placement types, including academia, government/think tank, and industry (private sector). Government/think tank positions are the baseline category across all specifications. The reported coefficients are in log-odds. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. The pronunciation time rating is a survey-based measure that records the median time it takes individuals to pronounce a name. The subjective difficulty rating is based on individuals' independent subjective assessments. Standard errors are in parentheses.

Table A4: Name Fluency and Placement Types: Multinomial Logit Estimates – Alternative Placement Categories

	(1) TT	(2) Govt/Think Tank	(3) Industry
Algorithm Rating: Full Name	0.100	0.284	0.119
	(0.099)	(0.125)	(0.117)
Observations	1,510	1,510	1,510
Control for Name Length	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes

	(4) TT	(5) Govt/Think Tank	(6) Industry
Pronunciation Time: Full Name	0.062	0.312	0.429
	(0.105)	(0.128)	(0.124)
Observations	1,510	1,510	1,510
Control for Name Length	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes

	(7) TT	(8) Govt/Think Tank	(9) Industry
Subjectively Difficult: Full Name	0.301	0.640	0.838
	(0.203)	(0.247)	(0.236)
Observations	1,510	1,510	1,510
Control for Name Length	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes

Notes: Each panel is estimated using a separate multinomial logit model with the dependent variable being a categorical variable capturing placement types, including tenure track, visiting/postdoc, government/think tank, and industry (private sector). Visiting/postdoc positions are the baseline category across all specifications. The reported coefficients are in log-odds. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based subrating schemes, where the weights are derived from neural network learning. The pronunciation time rating is a survey-based measure that records the median time it takes individuals to pronounce a name. The subjective difficulty rating is based on individuals' independent subjective assessments. Standard errors are in parentheses.

Table A5: Name Fluency and Placement Quality: Ordered Probit Estimates

	(1) RePEc_Imputed	(2) RePEc_Imputed	(3) RePEc_Imputed
Algorithm Rating: Full Name	0.076		
	(0.046)		
Pronunciation Time: Full Name		0.100	
		(0.048)	
Subjectively Difficult: Full Name			0.106
			(0.087)
Observations	1,510	1,510	1,510
Control for Name Length	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes

Notes: All specifications are estimated using an ordered probit model, where the dependent variable is based on the following ordered categories of the imputed RePEc research productivity index: 1) RePEc $\leq$ 50; 2) 50 $<$ RePEc $\leq$ 200; 3) 200 $<$ RePEc $\leq$ 400; 4) 400 $<$ RePEc $\leq$ 800; and 5) RePEc $= 1,000$. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. The pronunciation time rating is a survey-based measure that records the median time it takes individuals to pronounce a name. The subjective difficulty rating is based on individuals' independent subjective assessments. Robust standard errors are in parentheses.

Table A6: Name Fluency and Placement Outcomes: Controlling for Advisor Match

	(1) Academia	(2) Academia	(3) TT	(4) TT	(5) RePEc_Imputed	(6) RePEc_Imputed
Algorithm Rating: Full Name	-0.031	-0.030	-0.005	-0.005	66.572	65.118
	(0.016)	(0.016)	(0.017)	(0.017)	(31.722)	(31.664)
	(7) Academia	(8) Academia	(9) TT	(10) TT	(11) RePEc_Imputed	(12) RePEc_Imputed
Pronunciation Time: Full Name	-0.074	-0.074	-0.047	-0.046	80.610	80.334
	(0.018)	(0.018)	(0.018)	(0.018)	(33.735)	(33.952)
	(13) Academia	(14) Academia	(15) TT	(16) TT	(17) RePEc_Imputed	(18) RePEc_Imputed
Subjectively Difficult: Full Name	-0.117	-0.116	-0.060	-0.057	82.380	83.393
	(0.033)	(0.033)	(0.033)	(0.033)	(61.258)	(60.958)
Observations	1,469	1,469	1,499	1,499	1,510	1,510
Control for Country Match	Yes	No	Yes	No	Yes	No
Control for Language Match	No	Yes	No	Yes	No	Yes
Control for Name Length	Yes	Yes	Yes	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The coefficients in columns 1-4, 7-10, and 13-16 are marginal effects of probit regressions. The dependent variable in columns 1-2, 7-8, and 13-14 (3-4, 9-10 and 15-16) is a dichotomous variable for being placed in an academic (tenure track) position. Columns 5-6, 11-12, and 17-18 are estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of 1,000, the highest (worst) ranking. All tobit regressions are censored with an upper limit of $1,000$. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. The pronunciation time rating is a survey-based measure that records the median time it takes individuals to pronounce a name. The subjective difficulty rating is based on individuals' independent subjective assessments. The country/language match variables are indicator variables based on matching with at least one of the committee members. Robust standard errors are in parentheses.

Table A7: Name Fluency and Placement Outcomes by Gender

	Male Candidates			Female Candidates		
	(1) Academia	(2) TT	(3) RePEc_Imputed	(4) Academia	(5) TT	(6) RePEc_Imputed
Algorithm Rating: Full Name	-0.045	0.009	64.460	-0.036	-0.065	21.069
	(0.022)	(0.022)	(37.267)	(0.033)	(0.034)	(64.742)
Observations	970	1,016	1,053	392	413	457
Control for Name Length	Yes	Yes	Yes	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The coefficients in columns 1-2 and 3-4 are marginal effects of probit regressions. The dependent variable in columns 1 and 4 (2 and 5) is a dichotomous variable for being placed in an academic (tenure track) position. Columns 3 and 6 are estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of 1,000, the highest (worst) ranking. All tobit regressions are censored with an upper limit of $1,000$. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Robust standard errors are in parentheses.

Table A8: Name Fluency and Placement Outcomes: Excluding Candidates With Ethnically Chinese Names

	(1) Academia	(2) TT	(3) RePEc_Imputed
Algorithm Rating: Full Name	-0.037	-0.018	69.262
	(0.019)	(0.020)	(37.206)
Observations	1,094	1,093	1,131
Control for Name Length	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes

Notes: The sample excludes all job market candidates with ethnically Chinese names, regardless of their undergraduate locations. The coefficients in columns 1 and 2 are marginal effects of probit regressions. The dependent variable in column 1 (2) is a dichotomous variable for being placed in an academic (tenure track) position. Column 3 is estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of 1,000, the highest (worst) ranking. All tobit regressions are censored with an upper limit of $1,000$. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Robust standard errors are in parentheses.

Table A9: Name Fluency and Placement Outcomes: Country Fixed Effects

	(1) Academia	(2) TT	(3) RePEc_Imputed
Algorithm Rating: Full Name	-0.033	-0.002	76.923
	(0.017)	(0.018)	(31.260)
Observations	1,416	1,463	1,510
Control for Name Length	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes
Country/JM Cycle FE	Yes	Yes	Yes

Notes: The coefficients in columns 1 and 2 are marginal effects of probit regressions. The dependent variable in column 1 (2) is a dichotomous variable for being placed in an academic (tenure track) position. Column 3 is estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of 1,000, the highest (worst) ranking. All tobit regressions are censored with an upper limit of $1,000$. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Robust standard errors are in parentheses.

Table A10: Name Fluency and Placement Outcomes: Accounting for Common Names

	All Candidates			Candidates from U.S. and Canada		
	(1) Academia	(2) TT	(3) RePEc_Imputed	(4) Academia	(5) TT	(6) RePEc_Imputed
Common First Name	-0.006	-0.041	-44.623	-0.020	-0.040	20.921
	(0.043)	(0.045)	(89.256)	(0.058)	(0.054)	(115.622)
Common Last Name	0.006	-0.076	65.057	0.041	-0.044	101.178
	(0.069)	(0.069)	(124.601)	(0.106)	(0.098)	(172.005)
Common First Name ×	-0.232	-0.179	374.023	-0.119	-0.161	313.511
Common Last Name	(0.168)	(0.138)	(307.825)	(0.211)	(0.156)	(351.275)
Algorithm Rating: Full Name	-0.033	-0.014	69.557	-0.071	-0.048	130.065
	(0.017)	(0.017)	(32.978)	(0.029)	(0.028)	(55.243)
Observations	1,469	1,499	1,510	586	600	648
Control for Name Length	Yes	Yes	Yes	Yes	Yes	Yes
Other Controls	Yes	Yes	Yes	Yes	Yes	Yes
Subfield/Program FE	Yes	Yes	Yes	Yes	Yes	Yes
Region/JM Cycle FE	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The coefficients in columns 1-2 and 4-5 are marginal effects of probit regressions. The dependent variable in columns 1 and 3 (2 and 5) is a dichotomous variable for being placed in an academic (tenure track) position. Columns 3 and 6 are estimated using a tobit model, with the dependent variable being the imputed RePEc ranking of the institution of initial job placement, where private sector jobs are given an imputed ranking of 1,000, the highest (worst) ranking. All tobit regressions are censored with an upper limit of $1,000$. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Common first and last names are derived from the 1990 and 2010 U.S. Census, respectively. Robust standard errors are in parentheses.

Table A11: Black/Ethnic Immigrant Names and Callback Rates in Bertrand and Mullainathan (2004) and Oreopoulos (2011)

Bertrand and Mullainathan (2004)	Name Difficulty	Percent Callback
Black		
Ebony	-0.973	9.62
Kenya	-0.973	8.67
Leroy	-0.523	9.38
Tyrone	-0.361	5.33
Jermaine	0.004	9.62
Jamal	0.153	6.56
Tremayne	0.200	4.35
Tamika	0.297	5.47
Darnell	0.675	4.76
Rasheed	0.770	2.99
Latonya	0.826	9.13
Hakim	0.970	5.45
Kareem	1.038	4.69
Aisha	1.148	2.22
Keisha	1.547	3.83
Latoya	1.549	8.41
Tanisha	1.839	5.80
Lakisha	2.161	5.50
Average	0.575	6.21
Correlation:	-0.488 [0.040]	

Oreopoulos (2011)	Name Difficulty	Percent Callback
Indian		
Tara Singh	-0.603	10.29
Maya Kumar	-0.538	8.66
Shreya Sharma	0.348	9.54
Arjun Kumar	0.742	7.82
Samir Sharma	0.985	8.59
Panav Singh	1.264	8.25
Rahul Kaur	1.913	8.14
Priyanka Kaur	2.557	7.61
Average:	0.834	8.61
Correlation:	-0.755 [0.030]	
Pakistani		
Hassan Khan	-0.304	6.30
Fatima Sheikh	0.245	8.11
Sana Khan	0.392	8.82
Ali Saeed	0.705	8.33
Chaudhry Mohammad	1.102	6.12
Asif Sheikh	1.296	3.85
Hina Chaudhry	1.348	7.80
Rabab Saeed	3.142	4.26
Average:	0.991	6.70
Correlation:	-0.588 [0.125]	
Chinese		
Na Li	-0.802	7.65
Min Liu	-0.671	11.34
Lei Li	-0.644	9.32
Tao Wang	-0.557	10.98
Dong Liu	-0.534	7.88
Fang Wang	-0.283	12.57
Yong Zhang	-0.279	8.60
Xiuying Zhang	1.511	7.42
Average:	-0.283	9.47
Correlation:	-0.338 [0.412]	
Indian/Pakistani/Chinese		
Average:	0.514	8.26
Correlation:	-0.594 [0.002]	

Notes: The table contains all Black and ethnic immigrant names taken from publicly available replication data for Bertrand and Mullainathan (2004) and Oreopoulos (2011), respectively. The reported correlations are between name difficulty ratings and callback rates. P-values for correlations are in brackets.

Table A12: Name Fluency and Callback Rates: Experimental Data from Bertrand and Mullainathan (2004)

	All Applicants				Black Applicants	
	(1) Callback	(2) Callback	(3) Callback	(4) Callback	(5) Callback	(6) Callback
Black	-0.032		-0.018	-0.015		
	(0.006)		(0.006)	(0.006)		
Female				0.005		0.012
				(0.006)		(0.006)
College Educated				0.007		0.012
				(0.008)		(0.007)
Number of Jobs on Resume				-0.002		0.002
				(0.003)		(0.003)
Years of Experience				0.008		0.003
				(0.001)		(0.001)
Years of Experience^2				-0.000		-0.000
				(0.000)		(0.000)
Honors				0.054		0.040
				(0.017)		(0.011)
Volunteering Experience				-0.002		0.008
				(0.008)		(0.010)
Military Experience				0.003		-0.013
				(0.015)		(0.008)
Working in School				-0.001		-0.006
				(0.003)		(0.004)
Listing Email				0.011		-0.003
				(0.008)		(0.008)
Computer Skills				-0.024		-0.009
				(0.011)		(0.009)
Special Skills				0.063		0.049
				(0.008)		(0.006)
First Name Length				0.003		0.006
				(0.003)		(0.003)
Algorithm Rating: First Name		-0.017	-0.011	-0.012	-0.011	-0.014
		(0.004)	(0.005)	(0.005)	(0.003)	(0.003)
Observations	4,870	4,870	4,870	4,870	2,435	2,435

Notes: The sample is derived from publicly available replication data for Bertrand and Mullainathan (2004). Columns 1-4 include all job applicants, while columns 5-6 focus on Black applicants. The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

Table A13: Name Fluency and Callback Rates: Experimental Data from Oreopoulos (2011)

	All Applicants					Ind/Pak/Chn Applicants	
	(1) Callback	(2) Callback	(3) Callback	(4) Callback	(5) Callback	(6) Callback	(7) Callback
Female				0.018	0.019		0.006
				(0.005)	(0.005)		(0.007)
Top 200 World Ranking University				-0.003	-0.003		0.006
				(0.005)	(0.005)		(0.007)
Listing Extracurricular Activities				-0.002	-0.002		0.011
				(0.005)	(0.005)		(0.006)
Fluent in French & Other Languages				0.019	0.019		0.021
				(0.007)	(0.007)		(0.009)
Master's Degree				0.006	0.006		0.007
				(0.007)	(0.007)		(0.010)
High Quality Work Experience				0.009	0.009		0.014
				(0.005)	(0.005)		(0.007)
Additional Required Credentials				0.041	0.041		0.024
				(0.014)	(0.014)		(0.015)
Listing Canadian References				-0.029	-0.028		-0.022
				(0.015)	(0.015)		(0.015)
Accreditation of Foreign Education				-0.012	-0.012		-0.006
				(0.013)	(0.013)		(0.013)
Permanent Resident				-0.007	-0.007		-0.007
				(0.014)	(0.014)		(0.013)
Indian	-0.046		-0.036	-0.035	-0.033		0.002
	(0.005)		(0.007)	(0.009)	(0.009)		(0.008)
Pakistani	-0.057		-0.049	-0.049	-0.050		-0.015
	(0.007)		(0.008)	(0.009)	(0.009)		(0.012)
Chinese	-0.041		-0.038	-0.035	-0.029		
	(0.005)		(0.006)	(0.009)	(0.011)		
Chinese Canadian	-0.053		-0.053	-0.050	-0.045		
	(0.006)		(0.006)	(0.007)	(0.008)		
Greek	-0.031		-0.018	-0.018	-0.035		
	(0.012)		(0.015)	(0.016)	(0.018)		
British	-0.024		-0.024	-0.023	-0.023		
	(0.008)		(0.008)	(0.008)	(0.008)		
Full Name Length				0.000			-0.000
				(0.002)			(0.002)
Algorithm Rating: Full Name		-0.014	-0.008	-0.007		-0.008	-0.007
		(0.003)	(0.004)	(0.004)		(0.003)	(0.004)
First Name Length					-0.001		
					(0.002)		
Last Name Length					0.003		
					(0.003)		
Algorithm Rating: First Name					-0.006		
					(0.004)		
Algorithm Rating: Last Name					-0.001		
					(0.004)		
Observations	12,910	12,910	12,910	12,910	12,910	7,158	7,158

Notes: The sample is derived from publicly available replication data for Oreopoulos (2011). Columns 1-5 include all job applicants, while columns 6-7 focus on applicants with ethnically Indian, Pakistani, and Chinese names. The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

Table A14: Name Fluency and Callback Rates: Experimental Data from Oreopoulos (2011) – Sample of Ethnic Immigrant Applicants

	Indian (1) Callback	Pakistani (2) Callback	Chinese (3) Callback
Algorithm Rating: Full Name	-0.006	-0.012	-0.031
	(0.006)	(0.009)	(0.019)
Observations	3,312	957	2,848
Control for Name Length	Yes	Yes	Yes
Control for Gender	Yes	Yes	Yes
Control for Resume Characteristics	Yes	Yes	Yes

Notes: The sample is derived from publicly available replication data for Oreopoulos (2011). All specifications in this table focus on job applicants with ethnically Indian, Pakistani, and Chinese names. The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

Table A15: Name Fluency and Callback Rates by Gender: Experimental Data from Bertrand and Mullainathan (2004) and Oreopoulos (2011)

BERTRAND AND MULLAINATHAN (2004)	All Applicants		Black Applicants	
	Male	Female	Male	Female
	(1) Callback	(2) Callback	(3) Callback	(4) Callback
Algorithm Rating: First Name	-0.006	-0.016	-0.019	-0.020
	(0.033)	(0.002)	(0.016)	(0.004)
Observations	1,124	3,746	549	1,886
Control for Name Length	Yes	Yes	Yes	Yes
Control for Race	Yes	Yes	No	No
Control for Resume Characteristics	Yes	Yes	Yes	Yes

OREOPOULOS (2011)	All Applicants		Ind/Pak/Chn	
	Male	Female	Male	Female
	(5) Callback	(6) Callback	(7) Callback	(8) Callback
Algorithm Rating: Full Name	-0.002	-0.014	-0.003	-0.013
	(0.011)	(0.006)	(0.011)	(0.006)
Observations	6,343	6,567	3,543	3,615
Control for Name Length	Yes	Yes	Yes	Yes
Control for Ethnicity	Yes	Yes	Yes	Yes
Control for Resume Characteristics	Yes	Yes	Yes	Yes

Notes: The samples are derived from publicly available replication data for Bertrand and Mullainathan (2004) and Oreopoulos (2011). Columns 1-2 and 5-6 are based on the full sample of each data set, while columns 3-4 and 7-8 focus on the sample of Black job applicants and applicants with ethnically Indian, Pakistani, and Chinese names, respectively. The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

Table A16: Name Fluency and Callback Rates: Experimental Data from Bertrand and Mullainathan (2004) and Oreopoulos (2011) – Sample of Low Quality Resumes

	Bertrand and Mullainathan (2004) Low Quality Resume			Oreopoulos (2011) No Master's			Pooled Data Low Quality Resume/No Master's		
	(1) Callback	(2) Callback	(3) Callback	(4) Callback	(5) Callback	(6) Callback	(7) Callback	(8) Callback	(9) Callback
Black	-0.023	0.001	0.007						
	(0.007)	(0.005)	(0.005)						
Algorithm Rating: First Name		-0.018	-0.021						
		(0.003)	(0.003)						
Indian				-0.047	-0.036	-0.035			
				(0.006)	(0.008)	(0.010)			
Pakistani				-0.058	-0.050	-0.050			
				(0.007)	(0.009)	(0.009)			
Chinese				-0.041	-0.038	-0.034			
				(0.006)	(0.006)	(0.010)			
Chinese Canadian				-0.058	-0.058	-0.055			
				(0.006)	(0.006)	(0.007)			
Greek				-0.020	-0.005	-0.006			
				(0.015)	(0.019)	(0.019)			
British				-0.025	-0.025	-0.024			
				(0.009)	(0.009)	(0.009)			
Algorithm Rating: Full Name					-0.009	-0.008			
					(0.004)	(0.005)			
Black/Immigrant (Ind/Pak/Chn)							-0.029	-0.015	-0.017
							(0.005)	(0.006)	(0.007)
Algorithm Rating: First Name								-0.012	-0.010
								(0.003)	(0.004)
Observations	2,424	2,424	2,424	10,717	10,717	10,717	13,141	13,141	13,141
Control for Name Length	No	No	Yes	No	No	Yes	No	No	Yes
Control for Gender	No	No	Yes	No	No	Yes	No	No	Yes
Control for Resume Characteristics	No	No	Yes	No	No	Yes	No	No	Yes

Notes: The samples are derived from publicly available replication data for Bertrand and Mullainathan (2004) and Oreopoulos (2011). All specifications in this table focus on the subsample of job applicants with low quality resumes, where resume quality is determined based on a subjective measure in Bertrand and Mullainathan (2004) and whether one holds a Master's degree in Bertrand and Mullainathan (2004). The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

Table A17: Name Fluency and Callback Rates: Experimental Data from Nunley et al. (2015)

	(1) Callback	(2) Callback	(3) Callback	(4) Callback
Black	-0.028		-0.024	-0.034
	(0.007)		(0.007)	(0.009)
Algorithm Rating: Full Name		-0.009	-0.005	-0.009
		(0.003)	(0.004)	(0.005)
Observations	9,396	9,396	9,396	9,396
Control for Name Length	No	No	No	Yes
Control for Gender	No	No	No	Yes
Control for Resume Characteristics	No	No	No	Yes

Notes: The sample is derived from replication data for Nunley et al. (2015). The reported coefficients are marginal effects of probit regressions, where the dependent variable is a dichotomous variable for receiving a callback. The algorithm rating for name pronunciation difficulty is based on a weighted average of the letter-based and phoneme-based sub-rating schemes, where the weights are derived from neural network learning. Clustered standard errors at the job advertisement level are in parentheses.

Technical Appendix for "How Do You Say Your Name? Difficult-To-Pronounce Names and Labor Market Outcomes"

Qi Ge* Stephen Wu†

Abstract: In this technical appendix, we provide annotated code for the algorithm used to measure pronunciation difficulty for various words/names. This program was developed by James Kaffenbarger, Griffin Perry, Kenneth Talarico, Gwendolyn Urbanczyk, and Adam Valencia (December 2021).

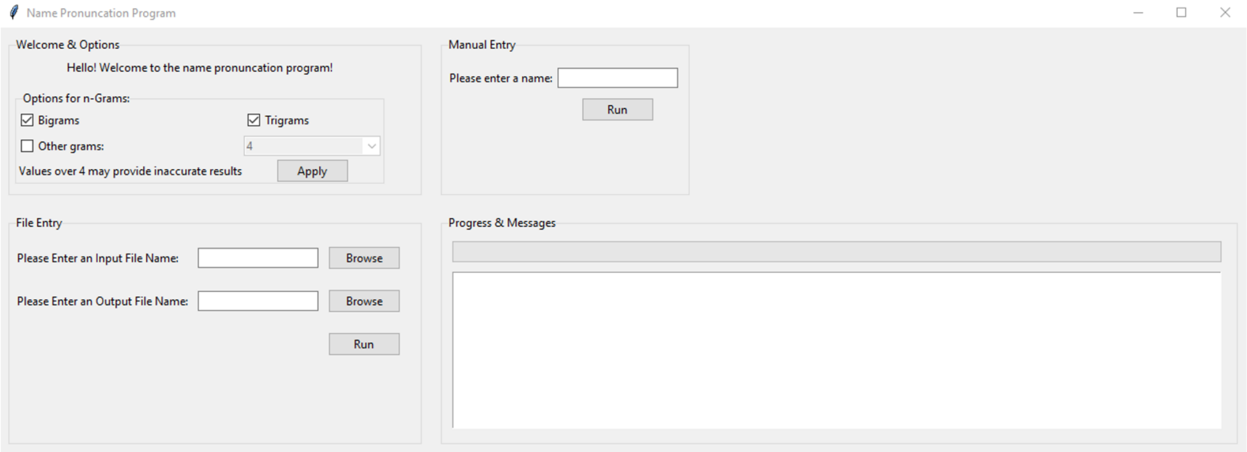
---

*Department of Economics, Vassar College, Poughkeepsie, NY 12604, *qige@vassar.edu*.

†Department of Economics, Hamilton College, Clinton, NY 13323, *swu@hamilton.edu*.

Pronunciation Algorithm

To execute and load the interface that allows you to run the algorithm to measure word complexity, download the folder and then execute/open the file titled run.bat. The interface will look like:



Name Pronuncation Program

Welcome & Options

Hello! Welcome to the name pronuncation program!

Options for n-Grams:

☑ Bigrams ☑ Trigrams

☐ Other grams: 4

Values over 4 may provide inaccurate results

Here is the python code for the main program:

```
from nameui import *
from to_ipa import to_ipa
import csv
from NNModel import convertToModelFormat, get_parent_languge,
↪  get_combined_output
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import math
from random import choice
from ngrams import NgramManager, Ngrams
import os
import time


class MainModel:
    """ Class for the superclass that controls all of the main
    ↪  functionality and
        contains all of the other models as instance variables. """

```

```
    def __init__(self,
     ↪  path_to_csv="ipa_dicts/english-general_american.csv"):
        """ Initializes models and the corpus of words. """
        with open(path_to_csv, encoding="utf8") as f:
            self.corpus = [w[1:-1] for row in csv.reader(f) \
                           for w in row[1].split(', ')]

        self.ipa_model = to_ipa(self)
        # SAE is "Standard American English"
        self.SAE_model = tf.keras.models.load_model('IsAmericanEnglishv4.0')
        self.root_model = tf.keras.models.load_model('RootLanguageModel')
        self.combine_model = tf.keras.models.load_model('Combine Scores
         ↪  Model')

        self.ngrams = NgramManager(self, 2, 3)

        # Needed to communicate/share data across threads
        self._gui = None
        self.prog_val = None
        self.to_gui_message = ""
        self.is_warning = False
        self.result = None
        self.lock = threading.Lock()

    def processInput(self, words):
        """ Method to be called every time the user submits new words. """

        # <names> is a list of every name the user inputted
        names = list(map(lambda x: x.lower().strip(), list(words[0])))
        self.addProgress(10)

        progressDivisor = len(names)
        if progressDivisor == 0:
            progressDivisor = len(names)

        # <ipa_names> is a list of the same length containing IPA
         ↪  transcriptions of each name
        #   i.e., ipa_names[i] is an IPA transcription of names[i]
        ipa_names = []
        progressVal = 0
        for name in names:
            ipa_names.append(self.ipa_model.to_ipa(name)[1:-1])

            progressVal += (15 / progressDivisor)
            if progressVal > 1:
```

```python
                self.addProgress(int(progressVal))
                progressVal = 0

        self.sendToMessageLog("IPA conversion complete", False)

        gram_letters  = []
        progressVal = 0
        for name in names:
            gram_letters.append(round(100 -
             ↪  self.ngrams.generateLetterProbs(name), 2))

            progressVal += (10 / progressDivisor)
            if progressVal > 1:
                self.addProgress(int(progressVal))
                progressVal = 0

        gram_phonemes = []
        progressVal = 0
        for name in ipa_names:
            gram_phonemes.append(round(100 -
             ↪  self.ngrams.generatePhonemeProbs(name), 2))

            progressVal += (10 / progressDivisor)
            if progressVal > 1:
                self.addProgress(int(progressVal))
                progressVal = 0

        self.sendToMessageLog("N-gram calculations complete", False)

        # get neural net scores
        # Tnks seems to take a while?
        phonemeNN = convertToModelFormat(self.SAE_model,
                                         pd.read_csv('Eng_2Chars.csv'),
                                         self)
        rootLanguageNN = convertToModelFormat(self.root_model,
                                         pd.read_csv('singleChars.csv'),
                                         self)

        nn_scores = phonemeNN.convert(names)
        root_NN_scores = rootLanguageNN.convert(ipa_names)
        root_Parents = get_parent_languge(root_NN_scores)

        self.sendToMessageLog("Neural Network calculations complete", False)


```

```

        combinedNGrams = [round((gram_letters[i] + gram_phonemes[i]) / 2, 2)
                        for i in range(len(gram_letters))]

        final_scores = get_combined_output(self.combine_model,
        ↪   combinedNGrams, gram_letters, gram_phonemes, nn_scores)
        final_scores = [round(x, 2) for x in final_scores]
        self.sendToMessageLog("Final score calculations complete", False)

        # Threading Stuff - need to acquire the lock (just to make sure)
        # then write the dataframe to the result attribute before
        ↪   releasing
        # the lock and firing the end thread virtual event
        self.lock.acquire()
        self.result =  pd.concat([words[0],
                                  pd.DataFrame(final_scores),
                                  pd.DataFrame(gram_letters),
                                  pd.DataFrame(gram_phonemes),
                                  pd.DataFrame(nn_scores),
                                  pd.DataFrame(root_Parents)],
                                  axis=1, ignore_index=True)
        self.lock.release()
        self.addProgress(5)
        self._gui.generateEvent("<<ThreadEnded>>")

    def setGUI(self, gui_win):
        """
        Method used to set the object's gui attribute.
        @params - self
                - gui_win: the Root_Win object to set _gui to
        @returns - None
        """
        self._gui = gui_win


    def setNGrams(self, nlist):
        """
        Method used to set the object's NGram's manager object so the user
        can select which n they want to run with Ngrams. (This method
↪   cannot
        be run by the GUI while in a multithreaded state, that would
        probably create issues)
        @params - self
                - nlist: a list of ints to pass to the NGrams manager
↪   constructor
```

```python
        @returns - None
        """
        self.ngrams = NgramManager(self, *nlist)

    def addProgress(self, value):
        """
        Method used to add progress to the progress bar. Sets prog_val to value
        and then fires the virtual event to add progress
        @params - self
                - value: the value to add to the progress bar
        @returns - None
        """
        self.lock.acquire()
        self.prog_val = value
        self.lock.release()
        self._gui.generateEvent("<<AddProgress>>")

    def sendToMessageLog(self, output, warning=True):
        """
        Method used to output a message to the message log. Sets is_warning to
        warning, to_gui_message to output, and fires the
        <<SendMessage>> virtual event
        @params - self
                - output: The message to be outputted to the log
                - warning: If true, the message is treated as a warning.
                           Otherwise, it is treated as an 'info' message.
        @returns - None
        """
        self.lock.acquire()
        self.is_warning = warning
        self.to_gui_message = output
        self.lock.release()
        self._gui.generateEvent("<<SendMessage>>")


    def test_gui(self, words):
        """
        Method used to test the gui without running the entire program.
        To use, on the line root = RootWin(model), add a true parameter
        to the RootWin constructor.
        """
        self._gui.generateEvent("<<ThreadEnded>>")

```

```
def main():
    """
    Main sets up the MainModel object and the GUI, then calls the GUI's
    mainloop.
    Since the GUI Needs to know about the model and the model about the GUI,
    we create the model first, then the GUI with the model, then set the model's
    gui to be the GUI we just created, before calling the mainloop.
    """
    try:
        model = MainModel()
        root = RootWin(model)
        model.setGUI(root)
    except Exception as e:
        output = "An error occured while setting up the program:\n"
        output += "".join(traceback.format_exception(type(e), e, e.__traceback__))
        print(output, file=sys.stderr)
        sys.exit(1)

    root.mainLoop()




if __name__ == '__main__':
    main()



# def testoutput():
#     with open("ipa_dicts/english-general_american.csv", encoding="utf8") as f:
#         reader = csv.reader(f)
#         corpus = [w[1:-1] for row in reader for w in row[1].split(', ')]
#     names = [choice(corpus) for _ in range(200)]
#     ipa_names = [ipa_model.ipa(name)[1:-1] for name in names]
#     ngrams_scores = [ngrams_phoneme_algorithm(name) for name in ipa_names]
#     nn_scores = getoutput(ipa_names, model)
#     final_scores = [round(((nn_scores[i] + ngrams_scores[i]) / 2) * 100, 2) for i in range(len(ngrams_scores))]
#     final_scores = [round(100 - x, 2) for x in final_scores]
#     with open("test-out.csv", 'w', encoding="utf8") as f:
```

```
#         writer = csv.writer(f)
#         writer.writerows([[names[i], final_scores[i]] for i in range(len(names))])
```

Here is python code that helps derive difficulty scores for letter n-grams and phoneme n-grams:

```
import csv
class NgramManager:
    def __init__(self, mainModel, *sizes):
        self.grams = [Ngrams(size) for size in sorted(sizes)]
        self.mainModel = mainModel

    def generateLetterProbs(self, words):
        probs = []
        #to deal with if name is multiple words
        words = words.split()
        for word in words:
            for gram in self.grams:
                if len(word) == 1:
                    #if the input is a single letter, "pronuncability" = 100
                    probs.append(100)
                if gram.length > len(word):
                    break
                probs.append(gram.generateLetterProbOccurence(word))
        if probs == []:
            self.mainModel.sendToMessageLog(f"Input: {word} too small for the current set nGrams, ignoring")
            return 0
        return sum(probs) / len(probs)

    def generatePhonemeProbs(self, words):
        probs = []
        #to deal with if name is multiple words
        words = words.split()
        for word in words:
            for gram in self.grams:
                if len(word) == 1:
                    #if the input is a single phoneme, "pronuncability" = 100
                    probs.append(100)
                if gram.length > len(word):
                    break
                probs.append(gram.generatePhonemeProbOccurence(word))
```

```python
        if probs == []:
            self.mainModel.sendToMessageLog(f"Input: {word} too small for the current set nGrams, ignoring")
            return 0
        return sum(probs) / len(probs)

class Ngrams:
    def __init__(self, length, corpus="ipa_dicts/english-general_american.csv", occurence_table="unigram_freq.csv"):
        self.length = length
        self.corpus = corpus
        self.occurence_table = occurence_table
        self.letter_dictionary = {}
        self.phoneme_dictionary = {}
        self.letter_occurence_dictionary = {}
        self.phoneme_occurence_dictionary = {}
        self._generateNgramDictionaries()
        #self._generateOtherOccurrenceDictionaries()
        self._generateOccurrenceDictionaries()

    def _generateOtherOccurrenceDictionaries(self):
        """ Opens and creates dictionaries that map each gram in the occurence dictionary to
        how often it occurs, (most is 1, least is 0)"""
        #print("starting to generate dictionaries")
        with open(self.occurence_table, encoding="utf8") as f:
            for row in csv.reader(f):
                #row[0] is the word, row[1] is the phoneme, row[2] is the occurence value
                letter_grams = self.generateNgrams(row[0])
                phoneme_grams = self.generateNgrams(row[1])
                for gram in letter_grams:
                    if self.letter_occurence_dictionary.get(gram) is None:
                        self.letter_occurence_dictionary.update({gram: row[2]})
                    else:
                        num = self.letter_occurence_dictionary.get(gram)
                        self.letter_occurence_dictionary.update({gram: num + row[2]})
                #print("finished letter dictionaries")
                for gram in phoneme_grams:
                    if self.phoneme_occurence_dictionary.get(gram) is None:
                        self.phoneme_occurence_dictionary.update({gram: row[2]})
```

```
                    else:
                        num = self.phoneme_occurence_dictionary.get(gram)
                        self.phoneme_occurence_dictionary.update({gram: num
                         ↪  + row[2]})

        #now we have the dictionaries with the total occurences. sort
         ↪  them from highest to lowest
        # and then scale them
        #print("generated non-scaled dictionaries")
        letter_sorted = sorted(self.letter_occurence_dictionary,
         ↪  key=self.letter_occurence_dictionary.get)
        for i in range(len(self.letter_occurence_dictionary)):
            self.letter_occurence_dictionary.update({letter_sorted[i]: ((i
             ↪  + 1) / len(self.letter_occurence_dictionary))})

        phoneme_sorted = sorted(self.phoneme_occurence_dictionary,
         ↪  key=self.phoneme_occurence_dictionary.get)
        for i in range(len(self.phoneme_occurence_dictionary)):
            self.phoneme_occurence_dictionary.update({phoneme_sorted[i]:
             ↪  ((i + 1) / len(self.phoneme_occurence_dictionary))})

        return

    def _generateOccurrenceDictionaries(self):
        """ Opens and creates dictionaries that map each word/phoneme to
         ↪  how often it occurs
            (most is 1, least is 0)"""
        count = 0
        with open(self.occurence_table, encoding="utf8") as f:
            for row in csv.reader(f):
                #hard coded the lengths of the occurence dictionaries,
                 ↪  will need to change later
                #if user wants to provide their own
                self.letter_occurence_dictionary[row[0]] = ((333333 -
                 ↪  count) / 333333)

                if self.phoneme_occurence_dictionary.get(row[1]) != None:
                    #this is done because there are a lot of words that
                     ↪  are pronounced
                    #the same, but spelled differently
                    count += 1
                    continue
                self.phoneme_occurence_dictionary[row[1]] = ((333333 -
                 ↪  count) / 333333)
                count += 1
```

```

    def generateNgrams(self, str):
        """ Given a string and an n, return a list of all grams of that
        ↪  length"""
        answer = []
        for i in range(0, len(str) - self.length + 1):
            end = i + self.length
            answer.append(str[i:end])
        return answer

    def _generateNgramDictionaries(self):
        """ Generates the dictionaries for both letters and phonemes,
        ↪  keeping track of
            the total occurences"""
        with open(self.corpus, encoding="utf8") as f:
                letter_corpus = [w[1:-1] for row in csv.reader(f) \
                for w in row[0].split(', ')]
        with open(self.corpus, encoding="utf8") as f:
                phoneme_corpus = [w[1:-1] for row in csv.reader(f) \
                for w in row[1].split(', ')]

        for str in letter_corpus:
            letter_grams = self.generateNgrams(str)
            for gram in letter_grams:
                if self.letter_dictionary.get(gram) is None:
                    self.letter_dictionary.update({gram: 1})
                else:
                    num = self.letter_dictionary.get(gram)
                    self.letter_dictionary.update({gram: num + 1})

        for str in phoneme_corpus:
            phoneme_grams = self.generateNgrams(str)
            for gram in phoneme_grams:
                if self.phoneme_dictionary.get(gram) is None:
                    self.phoneme_dictionary.update({gram: 1})
                else:
                    num = self.phoneme_dictionary.get(gram)
                    self.phoneme_dictionary.update({gram: num + 1})

        return

    def generateDictionaryLetterProb(self, word):
        """ Given a word, scale data with 100 == most occurences in the
        ↪  dictionary,
            not to be confused with the occurence csv"""
```

```python
        grams = self.generateNgrams(word)
        max_occurences = max(self.letter_dictionary.values()) / 100
        average_gram_prob = 0
        for gram in grams:
            if self.letter_dictionary.get(gram) == None:
                #if the gram is not in the dictionary, treat it as zero to avoid
                #dividing NoneType
                continue
            average_gram_prob += self.letter_dictionary.get(gram) / max_occurences

        if average_gram_prob != 0:
            average_gram_prob = average_gram_prob / len(grams)
        return average_gram_prob

    def generateDictionaryPhonemeProb(self, word):
        """ Given a phoneme, scale data with 100 == most occurences in the dictionary,
            not to be confused with the occurence csv"""
        grams = self.generateNgrams(word)
        max_occurences = max(self.phoneme_dictionary.values()) / 100
        average_gram_prob = 0
        for gram in grams:
            if self.phoneme_dictionary.get(gram) == None:
                #if the gram is not in the dictionary, treat it as zero to avoid
                #dividing NoneType
                continue
            average_gram_prob += self.phoneme_dictionary.get(gram) / max_occurences

        if average_gram_prob != 0:
            average_gram_prob = average_gram_prob / len(grams)
        return average_gram_prob

    def generateLetterProbOccurence(self, word):
        """ Given a word, call generateDictionaryLetterProb, and then scale it up
            using the letter occurence table"""
        prob = self.generateDictionaryLetterProb(word)
        if self.letter_occurence_dictionary.get(word) == None:
            #word is not in the occurence dictionary, so no scaling is done
            return prob
```

```
        scaler = float(self.letter_occurence_dictionary[word])
        prob += (100 - prob) * scaler
        return prob

    def generatePhonemeProbOccurence(self, phoneme):
        """ Given a phoneme, call generateDictionaryPhonemeProb, and then
        ↪  scale it up
            using the phoneme occurence table"""
        prob = self.generateDictionaryPhonemeProb(phoneme)
        if self.phoneme_occurence_dictionary.get(phoneme) == None:
            #phoneme is not in the occurence dictionary, so no scaling is
            ↪  done
            return prob
        scaler = float(self.phoneme_occurence_dictionary[phoneme])
        prob += (100 - prob) * scaler
        return prob
```

The following code provides examples of calculation for a sample of words:

```
# def generateLetterProbOccurence(self, word):
#         """ Given a word, call generateDictionaryLetterProb, and then
↪  scale it up
#             using the letter occurence table"""
        # prob = self.generateDictionaryLetterProb(word)
        # average_scaler = 0
        # for gram in self.generateNgrams(word):
        #     if self.letter_occurence_dictionary.get(gram) == None:
        #         continue
        #     average_scaler +=
        ↪  float(self.letter_occurence_dictionary[gram])
        # if average_scaler != 0:
        #     average_scaler = average_scaler /
        ↪  len(self.generateNgrams(word))
        # prob += (100 - prob) * average_scaler
        # return prob

# def generate_prob(self, word):
#     """ Given a word, compute the average gram prob """
#     grams = self.generateNgrams(word)
#     average_gram_prob = 0
#     for gram in grams:
#         average_gram_prob += self.dictionary.get(gram) / self.population

#     if average_gram_prob != 0:
```

```
#         average_gram_prob = average_gram_prob / len(grams)
#     return average_gram_prob

# data = ["hello", "world", "Ihope", "thisworks"]
# bi_gram = ngrams(data, 2)

# print(bi_gram.dictionary)
    # def ngrams_word_algorithm(word):
    #     """ Given a word, compute the tri_grams and get the average
     ↪  tri-gram value of the word
    #         from the corpus """
    #     word_trigrams = self.generateNgrams(word, 3)
    #     average_trigram_prob = 0
    #     for gram in word_trigrams:
    #         average_trigram_prob += tri_grams.get(gram) /
     ↪  bi_grams.get(gram[:-1])

    #     # To make sure that the word isn't composed completely of
     ↪  tri-grams not found
    #     # in the corpus
    #     if average_trigram_prob != 0:
    #         average_trigram_prob = average_trigram_prob /
     ↪  len(word_trigrams)

    #     return average_trigram_prob

    # def ngrams_phoneme_algorithm(phoneme):
    #     """ Given a phoneme, compute the z-score from the average of
     ↪  the bi-gram calculations
    #         and convert to a float between 0-1 """
    #     word_bigrams = generateNgrams(phoneme, 2)

    #     average_bigram_prob = 0
    #     for gram in word_bigrams:
    #         # If the corpus doesn't have this bi-gram, continue on to
     ↪  the next bi-gram.
    #         # Might need to change the weight of this later but for now
     ↪  it seems fine
    #         if bi_grams.get(gram) == None:
    #             continue

    #         average_bigram_prob += bi_grams.get(gram) /
     ↪  un_grams.get(gram[0])
    #         #average_bigram_prob += bi_grams.get(gram) / bi_gram_pop

```

```
    #     # To make sure that the word isn't composed completely of bi-grams not found
    #     # in the corpus
    #     if average_bigram_prob != 0:
    #         average_bigram_prob = average_bigram_prob / len(word_bigrams)

    #     z_score = (average_bigram_prob - average_corpus_prob) / standard_deviation

    #     answer = .5 * (math.erf(z_score / 2 ** .5) + 1) # https://stackoverflow.com/questions/2782284/function-to-convert-a-z-score-into-a-percentage

    #     return answer #average_bigram_prob
    #average_corpus_prob = len(bi_grams) / bi_gram_pop
# average_corpus_prob = 0
# for gram in bi_grams:
#     average_corpus_prob += bi_grams.get(gram) / un_grams.get(gram[0])
# average_corpus_prob = average_corpus_prob / bi_gram_pop

# standard_deviation = 0
# for gram in bi_grams:
#     standard_deviation += (bi_grams.get(gram) / un_grams.get(gram[0]) - average_corpus_prob) * (bi_grams.get(gram) / un_grams.get(gram[0]) - average_corpus_prob)
#     #standard_deviation += ((bi_grams.get(gram) / bi_gram_pop) - average_corpus_prob) * ((bi_grams.get(gram) / bi_gram_pop) - average_corpus_prob)
# standard_deviation = standard_deviation / (bi_gram_pop - 1)
# standard_deviation = math.sqrt(standard_deviation)
```

The following code takes the letter-based difficulty scores and the phoneme-based difficulty scores and uses a neural network model to calculate a final word difficulty score that is scaled to be between 0-100:

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import numpy as np
import time
```

```python
import os
from to_ipa import to_ipa

class convertToModelFormat():
    def __init__(self, model, columns, mainModel):
        self.model = model
        self.columns = columns
        self.columns.columns = ["Char(s)"]
        self.mainModel = mainModel



    def convert(self, inputlist):
        """Takes in inputs, and uses the columns given by preselected csv to run on the matching model
        """
        output = []
        progressDivisor = len(inputlist)
        if progressDivisor == 0:
            progressDivisor = len(inputlist)

        progressVal = 0
        temparr = []

        for ipaword in inputlist:

            temp = []
            for i in self.columns['Char(s)']:

                if i in ipaword:
                    temp.append(1)
                else:
                    temp.append(0)

            temparr.append(temp)
            progressVal += 25 / progressDivisor
            if progressVal > 1:
                self.mainModel.addProgress(int(progressVal))
                progressVal = 0


        answer = pd.DataFrame(temparr)
        answer.columns = self.columns['Char(s)'].values

        # Most of the runtime, presumably. Unpack?
```

```
        prediction = self.model.predict(temparr)


        roundedpred = []
        for i in prediction:
            temp = []
            for j in i:
                temp.append(j.round())
            roundedpred.append(temp)

        #output.append(roundedpred)

        return roundedpred

def get_parent_languge(arr):
    outputs = []
    for i in arr:
        if i[0] == 1:
            outputs.append("Germanic")
        elif i[1] == 1:
            outputs.append("Romance")
        elif i[2] == 1:
            outputs.append("Sino-Tebetan")
        else:
            outputs.append("Japonic")
    return outputs

def get_combined_output(model, final_scores, gram_letters, gram_phonemes, nn_scores):
    """Takes in the model, and the outputs from all other aspects of the program, and combines them into one score"""
    #The STDDEV and mean of the training data, used for scaling the outputs
    STDDEV = 0.136461
    MEAN = 1.251892
    inputDF = pd.DataFrame()
    temp = []
    for i in nn_scores:
        temp.append(i[0])
    inputDF["FinScores"] = final_scores
    inputDF["LetterNGramScores"] = gram_letters
    inputDF["PhonemeNGramScores"] = gram_letters
    inputDF["NNscores"] = temp
    prediction = model.predict(inputDF)
    holder = []
```

```
    for i in prediction:
        for j in i:
            #Ensures score is never over 100 or below 0
            if ((((j-MEAN)/STDDEV)*33) +50)> 100:
                holder+=[100.0]
            elif ((((j-MEAN)/STDDEV)*33) + 50)< 0:
                holder+=[0.0]
            else:
                #Scaled by 33 to make results spread wider across all
                ↪  values between 0-100, not centered around 50
                holder+=[(((j-MEAN)/STDDEV)*33) + 50]

    return holder
```