New Methods for Studying Old Work

by

José Ignacio Velarde Morales

B.S., Massachusetts Institute of Technology (2017)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2020

Author .....
Department of Electrical Engineering and Computer Science
August 14, 2020

Certified by .....
David Autor
Ford Professor of Economics
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

New Methods for Studying Old Work

by

José Ignacio Velarde Morales

Submitted to the Department of Electrical Engineering and Computer Science
on August 14, 2020, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Understanding the task content of new jobs is crucial to understanding labor markets. However, structured, task-level data about jobs in the US is nonexistent for the earlier decades of the 20th century. In this thesis, I create a novel dataset that can be used to study new work in 1940. This involves three main contributions. First, I match individual respondents in the 1940 Census to jobs in the 1940 Census Alphabetical Index of Occupations (CAI) using natural language processing (NLP) techniques. This allows us to identify which respondents were working in new jobs. Using the method I developed, I am able to match 85% of respondents in our sample to jobs in the CAI. The second contribution is to match individual respondents in the 1940 Census to jobs in the 1939 Dictionary of Occupational Titles (DOT). Using the method I developed, I am able to match 82% of respondents in our sample to jobs in the DOT. The third contribution of this work is to provide multiple measures of job complexity, skill requirements, and task composition for jobs in 1940. I create these measures using an NLP system that predicts these attributes based on each job's textual description from the 1939 Dictionary of Occupational Titles. I use later editions of the Dictionary of Occupational Titles to train and evaluate the system. The system is able to predict these measures with an accuracy of over 80%, and its predictions generalize well across years.

Thesis Supervisor: David Autor
Title: Ford Professor of Economics

Acknowledgments

This thesis is the culmination of many months of hard work and there are many people I could not have done it without. I consider myself immensely fortunate to have been blessed with such a wonderful team. The first person I want to thank is David Autor. Thank you for your kindness, enthusiasm, sense of humor, and mentorship over the last couple of years. Your guidance and example pushed me to be a better researcher and person. I also want to thank Anna Salomons, Marin Soljacic, Rumen Dangovski, Charlotte Loh, and Guillem Ramirez for their support on the technical aspects of this project, I could not have done it without all of you.

I also want to thank the people that supported me outside of research. I am grateful to Katrina LaCurts and Anne Hunter for their guidance and support during my master's program. I would also like to thank Christopher Ackerman and Rebecca Jackson for making E52 a warmer and friendlier place. I have very fond memories of my time working in the MIT Econ department and my friendship with the two of you is a huge source of that. I want to thank the people in MIT Casino Rueda, the Tech Catholic Community, and IV Homeless Outreach for helping make Cambridge my home these last two years. Finally, I would like to thank my mom, my little brother, and all of my family for their unconditional love and support during this time I have been away from home, you are my biggest treasure in this world.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Understanding new work is a crucial part of understanding the labor market effects of technologies. By new work, I mean jobs requiring new combinations of tasks or activities. Technological change is often accompanied by fears of widespread job displacement. Despite this, however, long run employment levels have not changed significantly over time. Even though technology makes some jobs obsolete, it often also creates demand for new jobs. A more complete understanding of the effects of a certain technology on the labor market requires that we study the jobs created, the jobs destroyed, and the changes to remaining jobs.

Task based models of labor displacement (see Acemoglu and Autor 2011 and Acemoglu and Restrepo 2018) are an essential part of our understanding of the interaction between laborers and technology. Autor, Levy, and Murnane 2003 use job task data along with a task based model to analyze how computerization altered job skill demands. Similarly, Acemoglu and Restrepo 2017 use a task based model to study the impact of industrial robots on employment and wages. There are several datasets that contain structured information about the tasks and skill requirements of different jobs (e.g. O*NET, Burning Glass, etc.). However, these datasets only cover recent decades. This kind of structured data is unavailable for years prior to 1965. Developing datasets with information on the skill requirements and task composition of jobs will be crucial to increasing our understanding of new work and inequality in these earlier time periods.

This thesis makes three contributions. First, I provide a method for individual respondents in the Census to job titles in the 1940 *Census Alphabetical Index of Occupations* (CAI). Second, I provide a method for matching respondents in the 1940 Census to job titles in the 1939 *Dictionary of Occupational Titles* (DOT). Finally, I create a natural language processing system that predicts job complexity, skill requirements, and broad task composition for jobs in 1940 based on their textual description.

The first contribution of this work is to match individual respondents in the Census to job titles in the 1940 *Census Alphabetical Index of Occupations* (CAI) using natural language processing techniques. I then use the method developed by Lin 2011 to identify which jobs in the Alphabetical Index were new jobs. This will allow researchers to study the characteristics of new workers in 1940. The 1940 Census has the free text each respondent used to describe their occupation. As a result, it contains many misspellings, abbreviations, and other features that make this a challenging task. Using the method I developed, I was able to: find a CAI job title for 85% of people in the sample, find workers in 1,286/1,544 unique new jobs, and identify over 475,000 new workers in 1940.

The second contribution of this work is to match individual respondents in the Census to job titles in the 1939 *Dictionary of Occupational Titles* (DOT) using natural language processing techniques. This was a more challenging problem than matching respondents to jobs in the CAI because the Census and the DOT use different occupational coding schemes and DOT titles tend to have a greater level of detail. Using the method I developed I was able to match 84% of respondents in our sample to jobs in the DOT, including over 380,000 new workers.

The third contribution of this work is to create a natural language processing system that predicts job complexity, skill requirements, and broad task composition for jobs in 1940 based on their textual description. I use the 1965, 1977, and 1991 editions of the *Dictionary of Occupational Titles* (DOT) to train and evaluate this system. The later editions of the DOT contain structured information about job complexity, skill requirements, and task composition of jobs, as well as textual descriptions

of the jobs. These later editions serve as training data. The different editions of the DOT also allow us to test how well the system's predictions generalize across years. The system provides accurate predictions of all attributes, and generalizes well across years. The average accuracy across all attributes was 82% when evaluated on a test set from the same year it was trained on. We use this system to predict job complexity, skill requirements, and task composition of jobs in 1940.

These three contributions can be combined to create a single dataset that has measures of job complexity, skill requirements, and broad task composition for most respondents in the 1940 Census as well as a flag identifying whether or not they were a new worker.

The rest of this thesis is structured as follows: Chapter 2 discusses related work. Chapter 3 describes the data used: The *Dictionary of Occupational Titles*, the *Census Alphabetical Index of Occupations*, and the 1940 Census Complete Count Files. Chapter 4 provides background on the natural language processing techniques used, and describes the methods used for matching and prediction. Chapter 5 discusses the results.

Chapter 2

Related Work

2.1 New Work Literature

Lin 2011 pioneered the approach of comparing successive editions of the *Census Alphabetical Index of Occupations* to identify new job titles. Once they had identified the new titles, they would compute the share of titles within each census occupation that were new titles. They then computed the average characteristics of new workers by weighting the characteristics of workers in each census occupation by each occupation's share of new titles. So for example, the share of new workers that were college educated would be the average share of college educated workers in each census occupation weighted by each occupation's share of new titles. This approach relies on the assumption that the share of new workers in each census occupation is equal to the share of new titles in that occupation. Using this approach, they found that new workers were concentrated in cities with more college educated workers and with more industry variety. Autor and Salomons 2019 use a similar approach to study new work. They find that a disproportionate share of new work is generated in cities. More importantly, they found that new work is heavily polarized amongst skill categories.

2.2 Machine Learning for Economics

Advances in machine learning and natural language processing have increased our ability to make high quality predictions. Athey 2018 documents many of the potential use cases for machine learning in economics. One of the most important use cases they document is using machine learning to create datasets that would require too many resources to create manually. There are already several examples of this in the literature. Gentzkow and Shapiro 2010 use simple natural language processing techniques to study newspaper political bias. Angrist et al. 2017 use natural language processing techniques to classify economics research articles into different subfields (e.g. labor economics, micro economics, etc.) and use the resulting classifications to study the impact of each subfield based on extramural citations. In this work, we use state of the art natural language processing techniques to create a dataset with structured information about job complexity, skill requirements, and task composition.

Chapter 3

Data

The goal of this project is to study the characteristics of new work and new workers in 1940. To do this, we use three data sources, which are described in further detail below. We use the Census Alphabetical Index of Occupations to identify which jobs were new jobs in 1940. We use the 1940 Census Complete Count Files to study the demographics, education level, and earnings of workers in 1940. We use the 1939 Dictionary of Occupational Titles to study the task content of jobs in 1940. Finally, we must link the three data sources to study the demographic characteristics of new workers and task content of new jobs. Matching job titles in the CCC to job titles in the CAI will allow us to identify new workers. Matching job titles in the CCC to job titles in the DOT will provide task content for respondents' jobs.

3.1 1940 Census Complete Count Files (CCC)

The United States conducts a census every ten years to collect data about its population. The Census collects a wide range of information about respondents including occupation, income, education, race, and place of residence. The data are confidential, but becomes publicly available 72 years after Census Day. The data from the 1940 Census became publicly available in April of 2012.

Many research projects that use Census data use the individual level micro data

provided by IPUMS (see Ruggles et al. 2010). This micro data suppresses respondent information such as name and exact address. One drawback of these micro data is that the occupational information in them is very coarse. For example, patent lawyers and divorce lawyers are both grouped into the occupation "lawyer". Each census occupation has a corresponding code. The complete count files have each respondent's written occupation in addition to their broad census occupation. Census occupations were assigned to respondents based on their written response. The written occupations often have typos and abbreviations. As a result, there are over 3 million unique job titles among the 132 million records in the 1940 complete count files. The table below contains some sample written occupations.

Table 3.1: Sample Occupational Titles from the Census Complete Count Files

Occupation String	**Occupation Code**	**Occupation Group**
Guard Fonman	602	Guards, watchmen, doorkeepers
Operator Building	308	Carpenters
Bank Runner	224	Messengers, errand, office boys
Truck Winder	496	Operatives & kindred workers, nec
Nutrinist	V52	Professional workers, nec

3.1.1 Sample Restrictions

We restricted our sample to be people in the labor force aged 16-64. We excluded unpaid family workers and individuals living in institutional group quarters. We also excluded individuals that were unemployed or in the military. We excluded individuals that had a census occupation code or census industry code of "999", both of which correspond to "unclassified". Finally, we only included individuals that had worked for more than one week in the previous year. There are 132,400,000 observations in the 1940 Census Complete Count files. After applying our sample restrictions, we are left with 39,655,167. Table 3.2 lists the sample restrictions we used.

Table 3.2: Sample Restrictions

Variable	**Restriction**
Age	Age between 16-64
Residential status	Not living in group quarters
Worker type	Not an unpaid family worker
Weeks worked	Worked for at least 1 week
Employment status	Employed and not in the military
Occupation Code	Occupation code other than 999
Industry Code	Industry code other than 999

3.1.2 Data We Worked With

To reduce the size of the dataset we worked with, we collapsed the data by unique combinations of occupation string and occupation code. There are 2,322,673 unique combinations of occupation string and occupation code in our data. For each unique combination of occupation string and occupation code, we computed the counts of the characteristics of interest. Using these counts, we are able to compute shares of demographic variables of interest (e.g. sex, educational shares, race, etc.).

3.2 1930/1940 Census Alphabetical Index of Occupations (CAI)

The Alphabetical Index of Occupations (CAI) is an official list of industries and occupations compiled by the US Census Bureau. It is continuously updated as occupations are created or become obsolete. The main purpose of the index is to map specific occupation titles into broader census occupation groups. An occupation title describes a narrow set of jobs that are comprised of a similar set of activities. Census occupation groups also describe occupations, but at a coarser level of detail. The 1940 index includes over 14,000 unique job titles, (e.g. Prosecutor lawyer) and their corresponding census occupation code (e.g. V26 - Lawyers and Judges). There are 225 unique census occupation codes in the 1940 Census. The index also includes information about the industries different occupations are found in. The index was used by Census employees to assign respondents official census occupations and industries.

We compare successive editions of the CAI to identify new jobs in each decade. For example, to determine if a job was a new in the 1940 CAI, we use the 1930 CAI to check if the job existed in 1930. If the job did not exist in 1930, we classify it as a new job. Some examples of new jobs in 1940 are "typewriter repairer", "research physician", and "jewelry polisher". For more details on this approach of identifying new work, see Lin 2011.

Table 3.3: Sample Occupational Titles from the Census Alphabetical Index

Occupation Title	**Census Occupation Code**	**Title Type**
Chemical Engineer	318	Existing
Carpenter	496	Existing
Jewelry Polisher	436	New
Research Physician	318	New
Pattern Developer	362	New

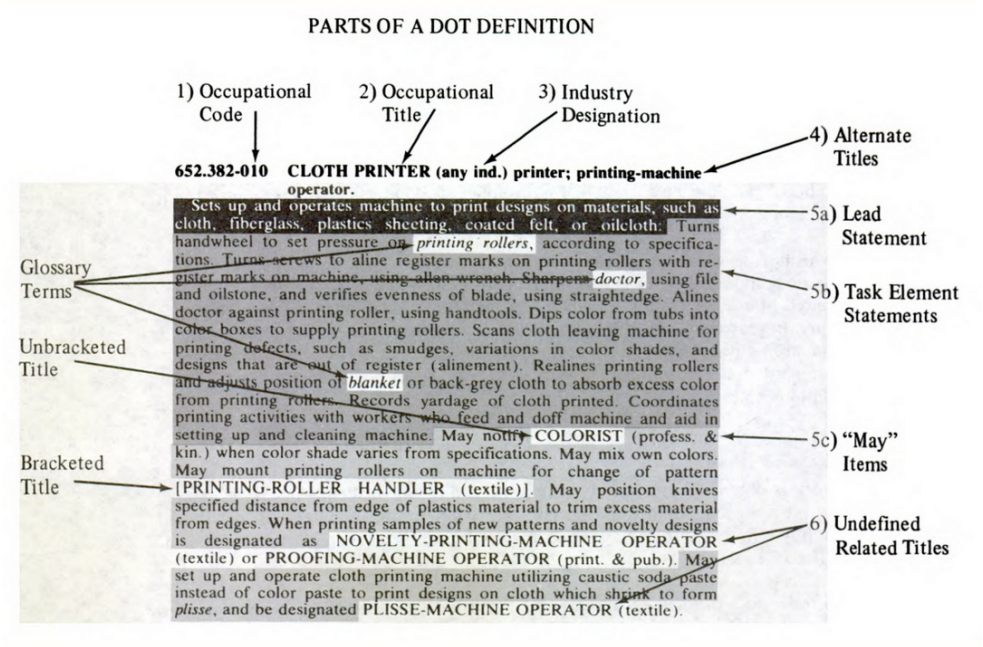
3.3 Dictionary of Occupational Titles

The Dictionary of Occupational Titles (DOT) is a volume that was created by the United States Department of Labor to help match job openings and job seekers. It provides detailed textual descriptions of thousands of jobs. The job descriptions generally describe what work is done, who it is done by, and why it is done. The descriptions also include information regarding the industry and occupation group the job is in. The first edition of the DOT was published in 1939.

3.3.1 DOT Occupational Codes

Every definition in the DOT includes an occupation code. The occupations in the dictionary were organized into a hierarchical coding structure developed by the United States Employment Service. In the 1939 DOT, all occupations are divided into seven major occupational groups and each major occupational group is divided into smaller groups at different levels of detail. The groups with the most granular level of detail are the three-digit occupation groups. For example, the first major occupational

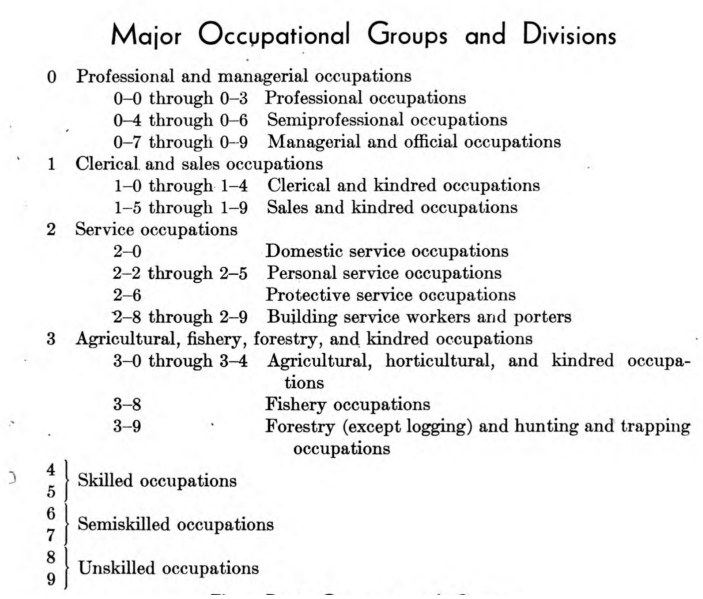
Figure 3-1: Example definition from the 1977 DOT



group is "Professional and Managerial Occupations" and all occupations in this group have occupation codes starting with '0'. Managerial occupations have codes starting with '0-7', '0-8', or '0-9'. The three-digit occupation group of "Hotel and restaurant managers" has an occupation code of '0-71'. There are 662 unique three-digit occupation groups. Figure 3-2 shows the occupational codes associated with each major occupational group.

3.3.2 Structured Information About Jobs

There are 5 editions of the DOT, published in 1939, 1949, 1965, 1977, and 1991. Each edition has about 13,000 job definitions. The 1965, 1977, and 1991 editions of the DOT include structured information about the jobs in addition to the textual description. The structured information is in the form of numerical codes that quantify job complexity, training requirements, and skill requirements. The structured information about aptitudes in 1977 is only available for a subset of 3,608 titles. The data was collected for all titles, but the source codebook that contains the data for all titles is currently lost. Table 3.4 shows the information available in each edition



Major Occupational Groups and Divisions

0 Professional and managerial occupations
- 0–0 through 0–3 Professional occupations
- 0–4 through 0–6 Semiprofessional occupations
- 0–7 through 0–9 Managerial and official occupations

1 Clerical and sales occupations
- 1–0 through 1–4 Clerical and kindred occupations
- 1–5 through 1–9 Sales and kindred occupations

2 Service occupations
- 2–0 Domestic service occupations
- 2–2 through 2–5 Personal service occupations
- 2–6 Protective service occupations
- 2–8 through 2–9 Building service workers and porters

3 Agricultural, fishery, forestry, and kindred occupations
- 3–0 through 3–4 Agricultural, horticultural, and kindred occupations
- 3–8 Fishery occupations
- 3–9 Forestry (except logging) and hunting and trapping occupations

4, 5 } Skilled occupations

6, 7 } Semiskilled occupations

8, 9 } Unskilled occupations

Figure 3-2: Major Occupational Groups and Divisions, DOT

of the DOT. The codes are further described in the following sections.

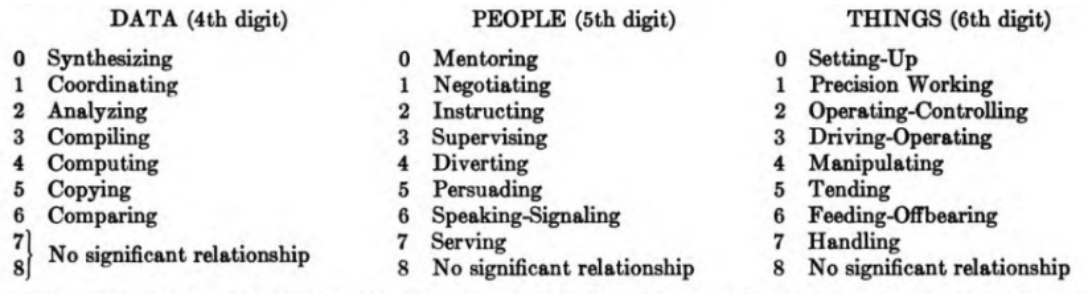
Table 3.4: DOT Data Availability

Edition	**# of Definitions**	**DPT**	**Attributes**
1939	26,997		
1949	21,020		
1965	12,339	✓	✓
1977	11,832	✓	✓*
1991	12,741	✓	✓

Data, People, Things Codes

Each job is assigned a 3-digit code that describes the job's complexity in relation to data, people, and things. Each digit in the code corresponds the job's complexity in one of those three categories. DPT codes use a reverse scale where lower values correspond to higher complexity. For example, the job of Aeronautical Test Engineer has a code of 061. The code tells us the job has high data complexity, medium/low people complexity, and high things complexity. The figure below shows the meanings of the different codes.

Figure 3-3: Explanation of Data, People, Things Code



DATA (4th digit)	PEOPLE (5th digit)	THINGS (6th digit)
0 Synthesizing	0 Mentoring	0 Setting-Up
1 Coordinating	1 Negotiating	1 Precision Working
2 Analyzing	2 Instructing	2 Operating-Controlling
3 Compiling	3 Supervising	3 Driving-Operating
4 Computing	4 Diverting	4 Manipulating
5 Copying	5 Persuading	5 Tending
6 Comparing	6 Speaking-Signaling	6 Feeding-Offbearing
7, 8 No significant relationship	7 Serving	7 Handling
	8 No significant relationship	8 No significant relationship

Job Attributes

Job descriptions in the DOT include descriptions of the mathematical education, aptitudes, and temperaments required to perform each job. Mathematical education refers to formal and informal education that develops basic reasoning skills in math. Aptitudes are specific abilities that an individual should have in order to perform a specific job. Examples of aptitudes are finger dexterity and hand-eye coordination. Temperaments are adaptability requirements made on the worker by specific types of job situations. These can include directing activities or performing repetitive work. The full list of job attributes we predict can be found in Table 3.5.

Table 3.5: DOT Job Attribute Descriptions

Attribute	**Description**	**Scale**
GED Math	Any education developing general math skills	1-6
Finger Dexterity	Ability to use fingers to manipulate small objects	1-5
Eye-hand-foot Coord.	Motor responsiveness to visual stimuli	1-5
DCP	Involves the direction and planning of activities	0/1
STS	Involves the precise attainment of set standards	0/1
SVP	Specific vocational preparation, training time	1-9

Job Attributes in 1965 In 1965, the job attributes were presented in a more aggregate manner. Instead of listing the attributes for each individual occupation, the 1965 DOT lists them for groups of occupations that share the same data, people, things codes. As a result, all occupations with the same DPT code are listed as having the same attributes. Since the attributes are listed for groups of occupations instead of individual occupations, single attributes often have multiple values. Whenever

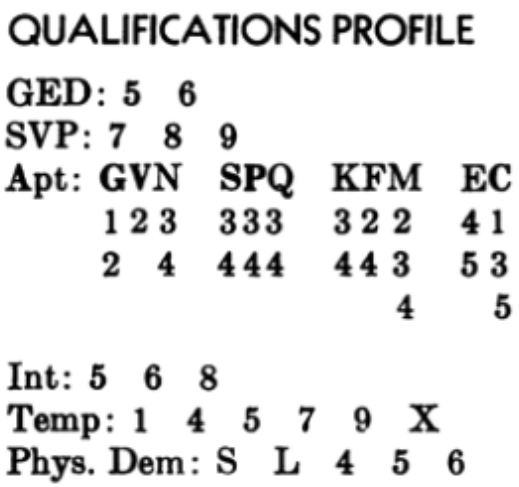


Figure 3-4: Example of Attributes in 1965

an attribute has multiple values, we use the median of the values. For example, occupations with DPT codes of 028 have possible GED values of 5 and 6 and possible SVP values of 7, 8, and 9, so we assign all of those occupations a GED value of 5.5 and an SVP value of 8. This also led to a difference in the number of labels for each attribute between 1965 and 1991. Figure 3-4 contains an example of how the attributes were presented in 1965.

3.4 DOT-Census Occupation Code Crosswalk

A major difficulty in linking titles in the CCC to titles in the DOT is that they have different occupational coding schemes. In general, DOT 3 digit occupation codes are much more detailed than 3 digit Census occupation codes. There are 662 unique DOT 3-digit occupation codes and 225 unique Census occupation codes. Census occupation codes are mostly industry independent, whereas many DOT occupation codes incorporate industry information. For example, no Census occupation code contains any mention of beverage production, however, the DOT occupation codes of "4-03", "6-03", and "8-03" are listed as corresponding to "Occupations in production of beverages". Many DOT occupation codes correspond to occupations in the production of a group of goods. Figure 3-5 contains examples of DOT occupations that include

industry information.

Occupations in Production of Food Products

4–01.	Bakers
4, 6, 8–02.	Occupations in production of bakery products, n. e. c.,
4, 6, 8–03.	Occupations in production of beverages
6, 8–04.	Occupations in canning and preserving of foods
4, 6, 8–05.	Occupations in production of confections
4, 6, 8–06.	Occupations in processing of dairy products
4, 6–07.	Millers, grain, flour, feed, etc.
4, 6, 8–08.	Occupations in production of grain-mill products, n. e. c.
4, 6, 8–09.	Occupations in slaughtering and in preparation of meat products
4, 6, 8–10.	Occupations in production of miscellaneous food products

Figure 3-5: Example of DOT occupations with industry information

SYMBOL		OCCUPATION AND OCCUPATION GROUP
Occ.	Ind.	
		CRAFTSMEN, FOREMEN, AND KINDRED WORKERS—Continued
		Inspectors (n. e. c.[1]), by industry:
318	V2–V8	Mining
318	V9	Construction
318	47	Railroads (includes repair shops)
318	45, 46, 48–54	Transportation, except railroad
318	55–6V	Communication and utilities
318	60–79	Wholesale and retail trade
318	VV–V1, XV–X9,	Miscellaneous industries and services [9]

Figure 3-6: Sample page from the CAI List of Principal Occupations and Industries

We use a section of the Census Alphabetical Index of Industries and Occupations called "List of Principal Occupations and Industries" to create a crosswalk between Census occupation codes and DOT occupation codes. This section contains the most common industry designations for each occupation code. Figure 3-6 contains a sample page from the List of Principal Occupations and Industries. The number of industry designations varies by occupation code. For example, the occupation code "712 - Boarding house and lodging house keepers" is only listed as appearing under industry "87 - Hotels and lodging places". The occupation code "496 - Operatives and kindred workers, nec", on the other hand, is listed as appearing in over 80 industries. Given a 3-digit occupation code from the DOT, we first try to find a Census occupation code

that closely resembles it. For example, the occupation group "Bakers" appears in both the Census and the DOT, so we match the two codes. For DOT occupations that contained industry information, such as "Occupations in the production of beverages", we would match them to all census occupation codes that were listed as appearing in that industry in the List of Principal Occupations and Industries. For example, given the 3-digit DOT occupation code "8-03 Occupations in production of beverages", we assign it to all the occupations in the List of Principal Occupations and Industries appearing under industry "X0 - Beverage industries". In this specific example, the only census occupation codes appearing in that industry are "496 - Operatives and kindred workers, nec" and "988 - Laborers, nec". So, the DOT code "8-03" is matched to Census codes 496 and 988 in our crosswalk.

Chapter 4

Methods

The main technical components of this project are: 1. matching job titles in the Census Complete Count Files (CCC) to job titles in the Census Alphabetical Index of Occupations (CAI) 2. matching job titles in the Census Complete Count Files to job titles in the Dictionary of Occupational Titles (DOT) and 3. predicting job attributes based on the textual descriptions of jobs in the 1939 Dictionary of Occupational Titles.

4.1 Natural Language Processing Background

4.1.1 Stemming and Lemmatization

Stemming and lemmatization are methods for obtaining the root or base form of a word. Stemming generally refers to heuristic algorithms that achieve this by removing letters at the end of words. Lemmatization usually involves predicting a word's part of speech and using a dictionary to find its base form. Since it is based on heuristics, stemming generally increases recall but hurts precision. Stemming is more robust to misspellings, but more likely to give false positives, lemmatization is unlikely to give false positives, but is susceptible to misspelling. Table 4.1 contains examples of the stemmed and lemmatized titles.

Title	**Lemmatized**	**Stemmed**
county engineer	county engineer	counti engin
pants trimmer	pant trimmer	pant trimmer
examiner claims	examiner claim	examin claim
drying room operator	dry room operator	dry room oper
veining shrimp	vein shrimp	vein shrimp
building from	build from	build from

Table 4.1: Examples of Stemming and Lemmatization

4.1.2 Bag of Words Representation of Documents

A corpus is a collection of documents, and documents are collections of words. One way to represent a document is a vector is the "bag of words" approach. In this approach, a corpus can be represented as a matrix where the rows represent documents and the columns represent words. The $(i, j)^{th}$ entry of this matrix corresponds to how many times word j appears in document i. This is called the "bag of words" approach because it does not take into account word order in a document, each document is treated as a "bag of words". For example, the corpus consisting of the two sentences: "The dog bit the cat" and "The cat bit the mouse" would be represented by the following term-document matrix where the columns correspond to the words the,dog,bit,cat,mouse.

$$\begin{bmatrix} 2 & 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Note: that words that are too common, such as "the", are called "stop words" and are often excluded from term-document matrices. Words that appear in all documents are also excluded.

4.1.3 TF-IDF Weighing

TF-IDF weighing is a way to give more importance to words that are more meaningful. TF stands for "term frequency". This is how many times a word appears in a document. IDF stands for "inverse document frequency". This corresponds to the number of documents in the corpus the word appears in. The $(i, j)^{th}$ entry in the TF-IDF matrix will be $tf(j, i) \times idf(j)$ where $tf(j, i)$ is how many times word j appears in document i and $idf(j)$ is word j's inverse document frequency. The matrix's rows are then normalized to have a Euclidean norm of 1.

Words that are very common, such as "the", "and", or "because", will have a very large document frequency because they will appear in most documents. As a result, they will have a very small inverse document frequency and their importance will be decreased. The exact formula for inverse document frequency used is:

$$idf(w) = \log \frac{n}{1 + df(w)} + 1$$

where n is the number of documents in the corpus and $df(w)$ is the number of documents word w appears in.

4.1.4 TFIDF Fuzzy Matching

TF-IDF weighing can be used to compute the similarity between two strings. This can be accomplished by treating each string as a document and treating groups of 3 characters in the strings as individual words. For example, for the string "helper" the "words", or groups of 3 characters, are: "hel", "elp", "lpe", "per". Each string can be represented as a vector, and we can compute the similarity of strings by computing the cosine distance of their vector representations. We use this method of fuzzy matching instead of traditional ones (e.g. Levenshtein, Jaro-Winkler, others etc) because this method is much faster. Using traditional algorithms, the fuzzy matching portion

of our matching procedure took over 21 days to run, TFIDF fuzzy matching took minutes to run and yielded similar results.

4.1.5 Word Embeddings

Word embeddings are dense, low dimensional vector representations of words popularized by Mikolov et al. 2013. These representations of words are able to capture word's semantic and syntactic meanings. Each word's position in the vector space captures some of that word's semantic and syntactic meaning. One of the most famous examples of this is as follows. Let v_{word} denote the vector representation of a given word, word. If we take the vector for king, v_{king} and subtract the word vector for man, v_{man}, and add the word vector for woman, v_{woman}, the resulting vector is very close to the word vector for queen, v_{queen}. In other words, $v_{king} - v_{man} + v_{woman} \approx v_{queen}$.

The first word embeddings were obtained by training a feedforward neural network with a single hidden hidden layer, on the task of predicting a word based on the word's context (i.e. it's surrounding words). In the sentence "the quick brown fox jumped over the lazy dog", the network would receive the words "fox" and "over" as input, and should predict "jumped". After the network was trained on this task, the weights from the hidden layer are used as embeddings.

Word embeddings improved the performance of natural language processing systems in a variety of tasks including sentiment analysis, document classification, and machine translation One limitation of word embeddings is that each word has a single vector representation, regradless of the number of definitions it has. For example, the embedding for the word "set" would be the same in the sentence "We should set the thermostat to a lower setting" as in the sentence "There is a large set of possibilities."

4.1.6 BERT/RoBERTa

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a model developed by Devlin et al. 2018 based on the Transformer model developed by Vaswani et al. 2017. One of BERT's key innovations is that it uses a word's entire context in a sentence to determine its vector represenation. The Transformer model enables the vector representation of every word to depend on all the other words in the sentence. This allows words with multiple meanings, such as "set" in the example above, to have different vector representations depending on the context. In addition to this, the BERT model provides a vector representation of the entire sentence. This representation can then be used for different NLP tasks such as document classification and question answering. When it was first published, BERT achieved state of the art results in eleven natural language processing tasks. Liu et al. 2019 were able to beat BERT's performance by optimizing the procedures used to train the model.

4.2 CCC-CAI Matching Procedure

The procedure used to match titles in the CCC to titles in the CAI is summarized below and elaborated on in the following subsections. We only matched titles that had the same census occupation codes. The thresholds used at step are available in Table 4.2.

1. **Preprocessing** remove capitalization, extra whitespace, and punctuation. Get alternative forms of titles: lemmatized, stemmed, fix misspellings.

2. **Exact Match** match observations with identical title, lemmatized title, stemmed title, or SymSpell title

3. **Fuzzy Match** match observations with similarly spelled title or lemmatized title

4. **TF-IDF Match** match observations with high title or lemmatized title TF-IDF similarity.

5. **Word Embedding Match** match observations with similar title word embedding representations.

Table 4.2: CAI Matching Thresholds

Match Type	**Title Form**	**Threshold**
Fuzzy	Original	.75
Fuzzy	Lemmatized	.775
TF-IDF	Original	.73
TF-IDF	Lemmatized	.775
Word Embedding	Original	.95

Preprocessing

We processed the titles by making all characters in the title lowercase, removing non alpha-numeric characters, and removing leading and trailing whitespace from titles. We created alternative forms of each title by applying different transformations. The transformations we apply are lemmatizing, stemming, fixing misspellings, and lemmatizing after fixing mispellings. We use the Python library, SymSpell, to correct misspelled titles. After applying the transformations, each title has five alternative forms: original, lemmatized, stemmed, spell fix, and lemmatized spell fix. SpellFix is applied at the end because it sometimes completely changes the meaning of a word. For example, it turns "pumper mechan" into "pumper meghan". See Table 4.3 for examples of the alternative forms.

1. Make all characters in the job title and census occupation code lowercase.

2. Remove any non alpha-numeric characters in the job title.

3. Remove any leading and trailing whitespace from both the job title and the census occupation code. Remove any extra whitespace between words in the job title. For example, "Patent Lawyer" becomes "Patent Lawyer".

Table 4.3: Title Transformations

Original Title	**Lemmatized**	**Stemmed**	**SpellFix**
ropeing tender	rope tender	rope tender	roping tender
samples clerk	sample clerk	sampl clerk	samples clerk
tickt printer	tickt printer	tickt printer	ticket printer
lginotype operator	lginotype operator	lginotyp oper	linotype operator

Exact Match

After we have created the alternate forms of each title, we match titles in the CCC to titles in the CAI that have identical titles and occupation codes. So, for example, given a title in the CCC, we would first try to find a title in the CAI that exaclty matched it. If that fails, we would then try to find a title in the CAI whose lemmatized version exactly matched the lemmatized version of the CCC title. If that failed, we would then try to find a title in the CAI whose stemmed version matched the stemmed version of the CCC title. We repeat this process with all of the alternative versions of the title mentioned in the Preprocessing section. The search order is: original title, lemmatized title, stemmed title, spell fix title, and lemmatized spell fix title.

Fuzzy Match

We use 3 character TF-IDF as described in section 4.1.4 to compute the string similarity between titles in the CCC and titles in the CAI. The fuzzy matching procedure is as follows: given a title in the CCC, we compute its 3 character TF-IDF similarity to all titles in the CAI with the same occupation code. We then match the CCC title to the CAI title with the highest 3 character TF-IDF similarity, as long as the similarity is above a certain threshold. We then repeat the same procedure using the lemmatized title instead of the original title.

TF-IDF Match

We run the TF-IDF matching procedure using the original title and the lemmatized title in that order.

Intuition TF-IDF calculates the importance of each word relative to the occupation code it appears in. For example, in the job title "hotel clerk" with occupation code "266 - Clerks", the word "hotel" provides more information than the word "clerk" because most titles in that occupation code will contain the word "clerk". The TF-IDF matching procedure matches titles based on the importance of the words that appear in both titles.

Description We treat each title as a document, and all of the titles under a single occupation code as a corpus. This means that there will be a different corpus corresponding to each occupation code. Note that a title's TF-IDF vector representation will depend on the corpus it appears in. This is because a term's frequency will vary across occupation codes. Let $v_{t,c}$ be the TF-IDF vector representation of a title, t appearing in corpus c. Given a title in the CCC, we compute its TF-IDF vector using the titles in the CAI with the same occupation code as a corpus. We also compute the TF-IDF vector of every CAI title in the corpus (i.e. every CAI title with the same occupation code). We then compute the cosine similarity of the CCC title's TF-IDF vector and the TF-IDF vector of every title in the corpus. We match the CCC title to the CAI title whose TF-IDF vector has the highest cosine similarity with the CCC title's TF-IDF vector, provided that the cosine similarity is above a certain threshold.

Mathematical Description Let t be a title in the CCC with occupation code O, and let $v_{t,c}$ be the TF-IDF vector representation of title t appearing in corpus c. Let CAI_O be the set of all titles in the CAI with occupation code O. To match a given title, t, we create v_{t,CAI_O}. We also create v_{g,CAI_O} for all titles, g in CAI_O. We then compute the TF-IDF similarity between titles as follows:

$$\text{sim}(g,t) = \frac{< v_{t,CAI_O}, v_{g,CAI_O} >}{||v_{t,CAI_O}|| \cdot ||v_{g,CAI_O}||}$$

The match for title t is then defined as:

$$\text{match}(t) = \{g \in CAI_O \mid \text{sim}(g,t) \geq \text{sim}(s,t), \forall s \in CAI_O \wedge \text{sim}(g,t) \geq q\}$$

where q is the specified threshold. Ties are broken alphabetically.

Word Embedding Match

We used a BERT model to create word embedding representations of titles in the CCC and titles in the CAI. Given a title, t, in the CCC, we would get its BERT vector representation, v_t. We would also get the BERT vector representations of all the titles in the CAI that had the same occupation code as t. We would then compute the cosine similarity between v_t and v_g for all titles, g, in the CAI that had the same occupation code as t. We would then match t to the title in the CAI whose BERT representation had the highest cosine similarity with t's BERT representation as long as that similarity was above a certain threshold.

Thresholds for Inexact Methods

We chose a threshold for each inexact match type (fuzzy, TF-IDF, word embedding) by manually inspecting a random sample of 100 census titles, potential matches in the CAI, and their corresponding similarity score. For each title and its potential match, we manually evaluated if the match was correct. For the fuzzy matching, we picked the lowest threshold that would give us an accuracy of at least 95%. For TF-IDF and Word Embedding matching, we picked the lowest threshold that would give us an accuracy of at least 70%.

4.3 CCC-DOT Matching Procedure

The procedure used to match titles in the CCC to titles in the DOT is summarized below and elaborated on in the following subsections. We only matched titles that had the same census occupation codes. We assigned DOT titles census occupation

codes based on a crosswalk we manually created, as described in section 3.4. The thresholds used at each step are available in Table 4.4

Procedure

1. **Preprocessing** remove capitalization, extra whitespace, and punctuation. Get alternative forms of titles: lemmatized, stemmed, SpellFix, lemmatized SpellFix, full title, lemmatized full title, stemmed full title.

2. **Exact Match** match observations with identical title, matched CAI title, lemmatized title, stemmed title, or SymSpell title. Match observations with identical full title, lemmatized full title, or stemmed full title.

3. **Fuzzy Match** match observations with similarly spelled lemmatized title, CAI title, or full title.

4. **TF-IDF Match** match observations with high lemmatized title or lemmatized full title TF-IDF similarity.

5. **Manual Match** manually match the top 300 occupations with the most respondents.

Table 4.4: DOT Matching Thresholds

Match Type	**Title Form**	**Threshold**
Fuzzy	Lemmatized	.7
Fuzzy	Matched CAI Title	.7
Fuzzy	Original Full Title	.75
TF-IDF	Lemmatized	.7
TF-IDF	Lemmatized Full Title	.8

Preprocessing

We process the titles by making all characters in the title lowercase, removing non alpha-numeric characters, and removing leading and trailing whitespace from titles.

We create alternative forms of each title by applying different transformations and adding industry information. The transformations we apply are lemmatizing, stemming, fixing misspellings, and lemmatizing after fixing mispellings. We use the Python library, SymSpell, to correct misspelled titles. We use each respondent's industry code to create a "full occupational title" for each respondent.

Another alternative title form we use is the CAI title the CCC title was matched to in the CCC-CAI title matching. After applying the transformations, adding the industry information, and including the matched CAI title, each title had nine alternative forms: original, matched CAI title, lemmatized, stemmed, spell fix, lemmatized spell fix, full title, lemmatized full title, and stemmed full title. SpellFix is applied after lemmatizing and stemming because it sometimes completely changes the meaning of a word. For example, it turns "pumper mechan" into "pumper meghan". See Figure 4-1 for examples of the alternative forms.

Original Title	**Lemmatized**	**Stemmed**	**SpellFix**
ropeing tender	rope tender	rope tender	roping tender
samples clerk	sample clerk	sampl clerk	samples clerk
tickt printer	tickt printer	tickt printer	ticket printer
lginotype operator	lginotype operator	lginotyp oper	linotype operator

(a) Title Transformations

Original Title	**Industry**	**Full Title**
trucker	wholesale trade	trucker wholesale trade
labor	postal service	labor postal service
proprietor	food stores	proprietor food stores
worker	construction	worker construction
asst manager	misc retail stores	asst manager misc retail stores

(b) Full Titles

Figure 4-1: Alternative Title Forms

Exact Match

After we have created the alternate forms of each title, we match titles in the CCC to titles in the DOT that have identical titles and occupation codes. So, for example, given a title in the CCC, we would first try to find a title in the CAI that exaclty matched it. If that fails, we would then try to find a title in the CAI whose lemmatized version exactly matched the lemmatized version of the CCC title. If that failed, we would then try to find a title in the DOT whose stemmed version matched the stemmed version of the CCC title. We repeat this process with all of the alternative versions of the title mentioned in the Preprocessing section. The search order is: original title, matched CAI title, lemmatized title, stemmed title, spell fix title, lemmatized spell fix title, full title, lemmatized full title, and stemmed full title.

Fuzzy Match

We use 3 character TF-IDF as described in section 4.1.4 to compute the string similarity between titles in the CCC and titles in the CAI. The fuzzy matching procedure is as follows: given a title in the CCC, we compute its 3 character TF-IDF similarity to all titles in the DOT with the same occupation code. We then match the CCC title to the CAI title with the highest 3 character TF-IDF similarity, as long as the similarity is above a certain threshold. We then repeat the same procedure using several alternative versions of the title. The search order is: lemmatized title, matched CAI title, and full title.

TF-IDF Match

We run the TF-IDF matching procedure using the lemmatized title and lemmatized full title in that order.

Intuition TF-IDF calculates the importance of each word relative to the occupation code it appears in. For example, in the job title "hotel clerk" with occupation code "266 - Clerks", the word "hotel" provides more information than the word "clerk"

because most titles in that occupation code will contain the word "clerk". The TF-IDF matching procedure matches titles based on the importance of the words that appear in both titles.

Description We treat each title as a document, and all of the titles under a single occupation code as a corpus. This means that there will be a different corpus corresponding to each occupation code. Note that a title's TF-IDF vector representation will depend on the corpus it appears in. This is because a term's frequency will vary across occupation codes. Let $v_{t,c}$ be the TF-IDF vector representation of a title, t appearing in corpus c. Given a title in the CCC, we compute its TF-IDF vector using the titles in the DOT with the same occupation code as a corpus. We also compute the TF-IDF vector of every DOT title in the corpus (i.e. every CAI title with the same occupation code). We then compute the cosine similarity of the CCC title's TF-IDF vector and the TF-IDF vector of every title in the corpus. We match the CCC title to the DOT title whose TF-IDF vector has the highest cosine similarity with the CCC title's TF-IDF vector, provided that the cosine similarity is above a certain threshold.

Mathematical Description Let t be a title in the CCC with occupation code O, and let $v_{t,c}$ be the TF-IDF vector representation of title t appearing in corpus c. Let DOT_O be the set of all titles in the DOT with occupation code O. To match a given title, t, we create v_{t,DOT_O}. We also create v_{g,DOT_O} for all titles, g in DOT_O. We then compute the TF-IDF similarity between titles as follows:

$$\text{sim}(g,t) = \frac{< v_{t,DOT_O}, v_{g,DOT_O} >}{||v_{t,DOT_O}|| \cdot ||v_{g,DOT_O}||}$$

The match for title t is then defined as:

$$\text{match}(t) = \{g \in DOT_O \mid \text{sim}(g,t) \geq \text{sim}(s,t), \forall s \in DOT_O \wedge \text{sim}(g,t) \geq q\}$$

where q is the specified threshold. Ties are broken alphabetically.

Thresholds for Inexact Methods

We chose a threshold for each inexact match type (fuzzy and TF-IDF) by manually inspecting a random sample of 100 census titles, potential matches in the DOT, and their corresponding similarity score. For each title and its potential match, we manually evaluated if the match was correct. We generally chose the thresholds such that the accuracy was at least 90% for fuzzy matching and at least 80% for TF-IDF.

4.4 Job Characteristics Prediction

We used the textual descriptions of jobs in 1940 to predict their job complexity as well as their routineness. The 1965, 1977, and 1991 editions of the DOT serve as our labeled data because every job title includes a definition as well as labels indicating job complexity and job routineness.

4.4.1 Data

4.4.1.1 Data, People, Things Complexity

As mentioned in the data chapter, each job in the later editions of the DOT is assigned a 3-digit code that describes the job's complexity in relation to data, people, and things. Each digit in the code corresponds the job's complexity in one of those three categories. The figure below shows the meanings of the different codes.

Figure 4-2: Explanation of Data, People, Things Code

DATA (4th digit)	PEOPLE (5th digit)	THINGS (6th digit)
0 Synthesizing	0 Mentoring	0 Setting-Up
1 Coordinating	1 Negotiating	1 Precision Working
2 Analyzing	2 Instructing	2 Operating-Controlling
3 Compiling	3 Supervising	3 Driving-Operating
4 Computing	4 Diverting	4 Manipulating
5 Copying	5 Persuading	5 Tending
6 Comparing	6 Speaking-Signaling	6 Feeding-Offbearing
7, 8 No significant relationship	7 Serving	7 Handling
	8 No significant relationship	8 No significant relationship

For the task of predicting these codes, we used data from the 1965, 1977, and 1991 Dictionary of Occupational Titles. The 1965 data has 12,339 observations, the 1977 data has 11,968 observations and the 1991 data has 12,741 We split each dataset into a training, validation, and test set using a 60/20/20 split. We trained one model on data from each year and tested it on the data from the two other years in order to evaluate model generalize-ability with older and newer data.

4.4.1.2 Aptitudes/Temperaments

Job descriptions in the DOT include descriptions of the mathematical education, aptitudes, and temperaments required to perform each job. Mathematical education refers to formal and informal education that develops quantitative reasoning. Aptitudes are specific abilities that an individual should have in order to perform a specific job. Examples of aptitudes are finger dexterity and hand-eye coordination. Temperaments are adaptability requirements made on the worker by specific types of job situations. These can include directing activities or performing repetitive work. The full list of job attributes we predict can be found in the table below.

Attribute	**Description**	**Scale**
GED Math	Any education developing quantitative skills	1-6
Finger Dexterity	Ability to use fingers to manipulate small objects	1-5
Eye-hand-foot Coord.	Motor responsiveness to visual stimuli	1-5
DCP	Involves the direction and planning of activities	0/1
STS	Involves the precise attainment of set standards	0/1

For this task, we used data from the 1965, 1977, 1991 DOT. However, since the 1977 data only had 3,608 labeled observations, we only used it to evaluate other models. The 1991 data has 12,741 observations. We split each dataset into training, validation, and test sets using a 60/20/20 split. We trained one model on the 1965 data and tested it on the 1991 data and vice versa in order to evaluate model generalize-ability with older and newer data.

4.4.2 Model

For the prediction task, we used a pre-trained RoBERTa Transformer (see Liu et al. 2019) attached to a single layer neural network. We used Wolf et al. 2019's implementation of the RoBERTa Transformer. The neural network attached to the transformer has 768 nodes in the hidden layer and we use cross entropy loss. We trained the network for 10 epochs with early stopping. The performance of neural networks tends to be unstable during training, to ameliorate this, we evaluated the model's performance on the validation set after every training epoch, and only saved the model if it outperformed all the previous models. If the model's performance did not improve after 3 consecutive training epochs, training automatically ended and the best performing model was kept.

Chapter 5

Results

5.1 CCC-CAI Matching

Using the matching procedure we developed, we were able to match the jobs of 84.7% of respondents in our sample to jobs in the Census Alphabetical Index of Occupations. Additionally, we were able to find people working in 1,286 our of 1,544 unique new jobs and a total of 475,245 people working in new jobs. The procedure we developed is able to match misspelled titles, incomplete titles, and synonyms of titles.

5.1.1 Match Rate Progression

Exact matches of the original titles or their alternative forms accounted for 68.5% of respondents in our sample. Fuzzy matching yielded another 6.2% of respondents. The remaining 9.6% of respondents were matched using TF-IDF. Table 5.1 shows how the number of matched workers changes at each step of the procedure. Table 5.2 shows how the number of matched titles changes at each step of the procedure. Table 5.3 shows the number and fraction of new, existing, and overall CAI titles matched.

Table 5.1: Progression of workers matched to a CAI title

Match Type	**Absolute**		**Cumulative**	
	# of Workers	**# of New Workers**	**# of Workers**	**Share of Workers**
Exact	2,6135,755	121,851	26,135,755	0.659
CAI Industry Info Exact	4,746	0	26,140,501	0.659
Lemmatized Title Exact	86,487	4,565	26,226,988	0.661
Stemmed Title Exact	532,235	20,460	26,759,223	0.675
SpellFix Title Exact	411,449	4,200	27,170,672	0.685
SpellFix Lemma Exact	8,528	65	27,179,200	0.685
Original Title Fuzzy	2,398,602	131,363	29,577,802	0.746
Lemmatized Title Fuzzy	76,867	2,144	29,654,669	0.748
Original Title TF-IDF	3,645,705	154,352	33,300,374	0.840
Lemmatized Title TF-IDF	176,784	6,799	33,477,158	0.844
Word Embedding	145,663	29,446	33,622,821	0.848

Table 5.2: Progression of CCC titles matched to CAI titles

Match Type	**Absolute**		**Cumulative**	
	# of Titles	**# of New Titles**	**# of Titles**	**Share of CCC Titles**
Exact	10,211	809	10,211	0.004
CAI Industry Info Exact	251	0	10,462	0.005
Lemmatized Title Exact	2,037	88	12,499	0.005
Stemmed Title Exact	6,521	277	19,020	0.008
SpellFix Title Exact	66,741	1,714	85,761	0.037
Lemmatized SpellFix Title Exact	1,613	38	87,374	0.038
Original Title Fuzzy	320,003	14,163	407,377	0.175
Lemmatized Title Fuzzy	12,988	561	420,365	0.181
Original Title TF-IDF	665,637	34,988	1,086,002	0.468
Lemmatized Title TF-IDF	45,626	1,736	1,131,628	0.487
Word Embedding	10,060	804	1,141,688	0.492

Title Type	**Count Matched**	**Fraction Matched**
Existing	11,988	0.936
New	1,304	0.845
Overall	13,292	0.926

Table 5.3: Fraction of CAI Titles Matched

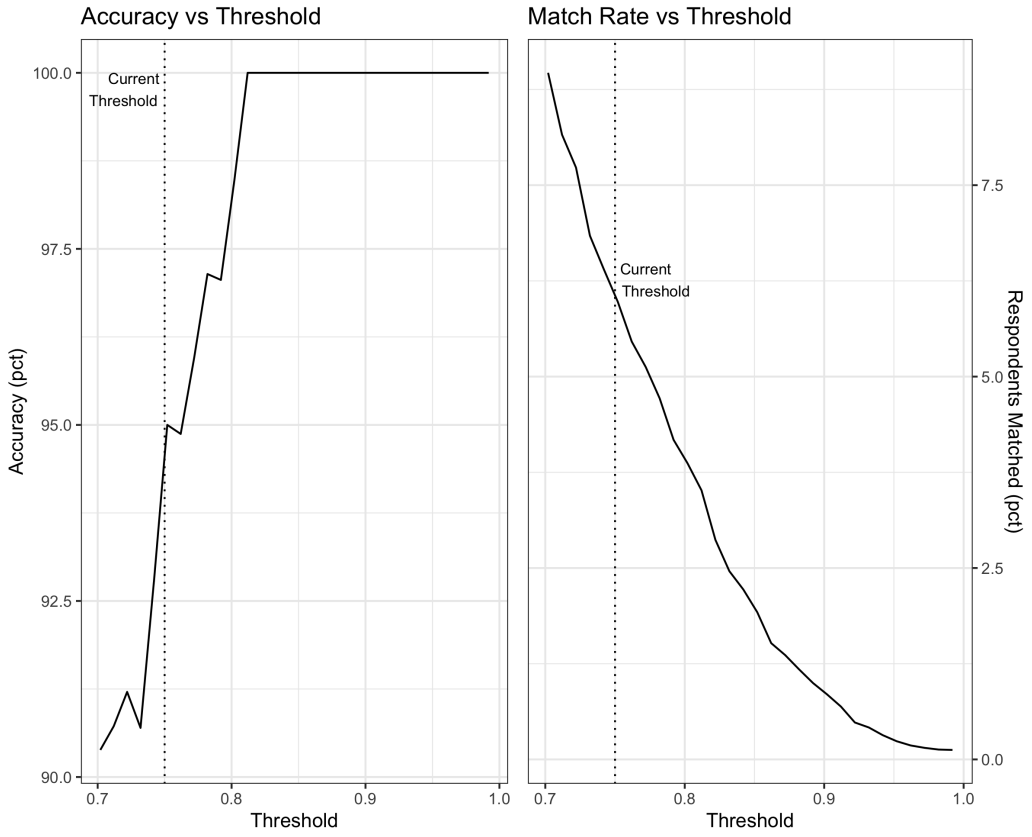
5.1.2 Accuracy and Match Rate of Approximate Methods

To evaluate the accuracy of the fuzzy matching, TFIDF matching, and word embedding matching, we randomly sampled 100 titles matched by each method and audited them manually. We used these manual audits to determine the accuracy of each method at different thresholds. For the fuzzy matching, we picked the lowest threshold that would give us an accuracy of at least 95%. For TF-IDF and Word Embedding matching, we picked the lowest threshold that would give us an accuracy

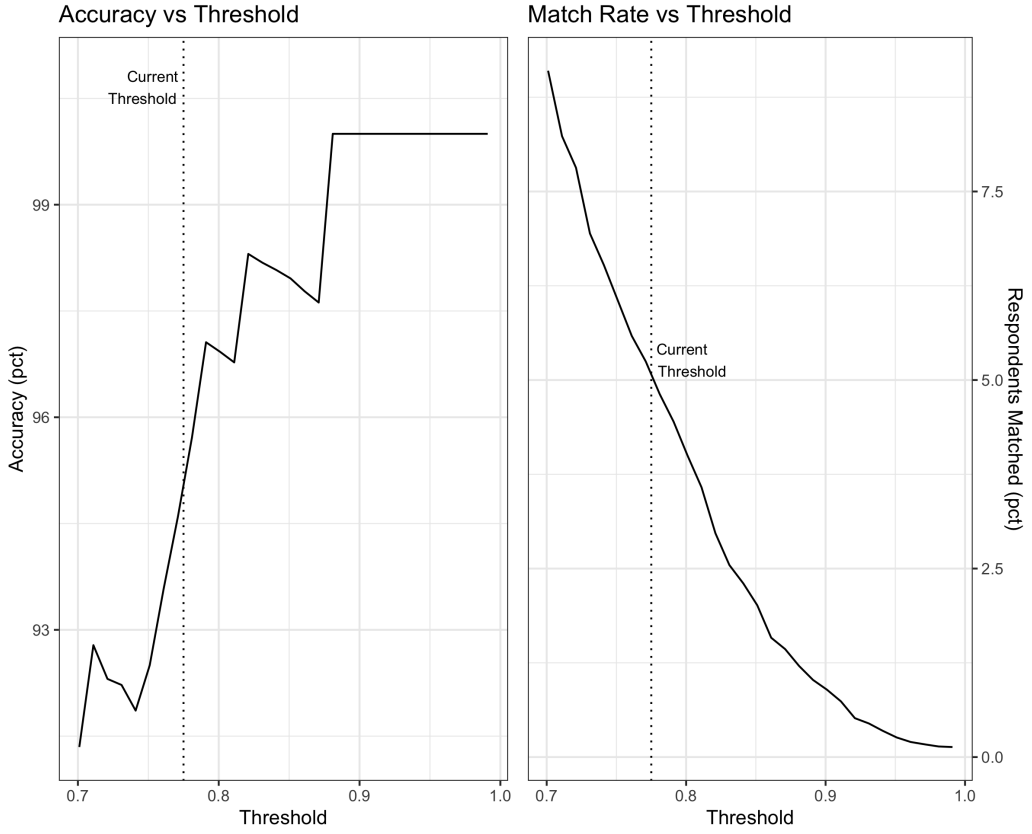
of at least 70%. Table 5.4 shows the accuracy of each method at the threshold we used.

Table 5.4: Accuracy of Approximate Methods

Match Type	**Accuracy**	**Share of Workers Matched**
Original Title Fuzzy	0.950	0.060
Lemmatized Title Fuzzy	0.959	0.002
Original Title TFIDF	0.706	0.092
Lematized Title TFIDF	0.717	0.004
Word Embedding	0.700	0.004

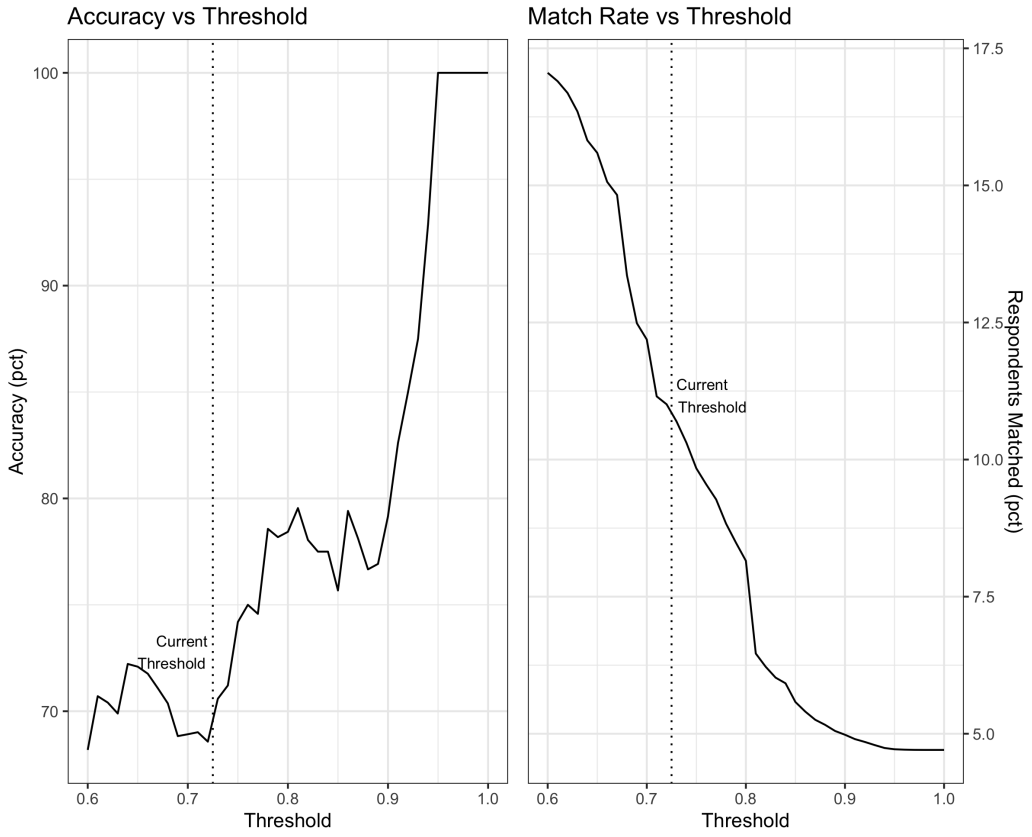


(a) Original Title Fuzzy Matching Plots

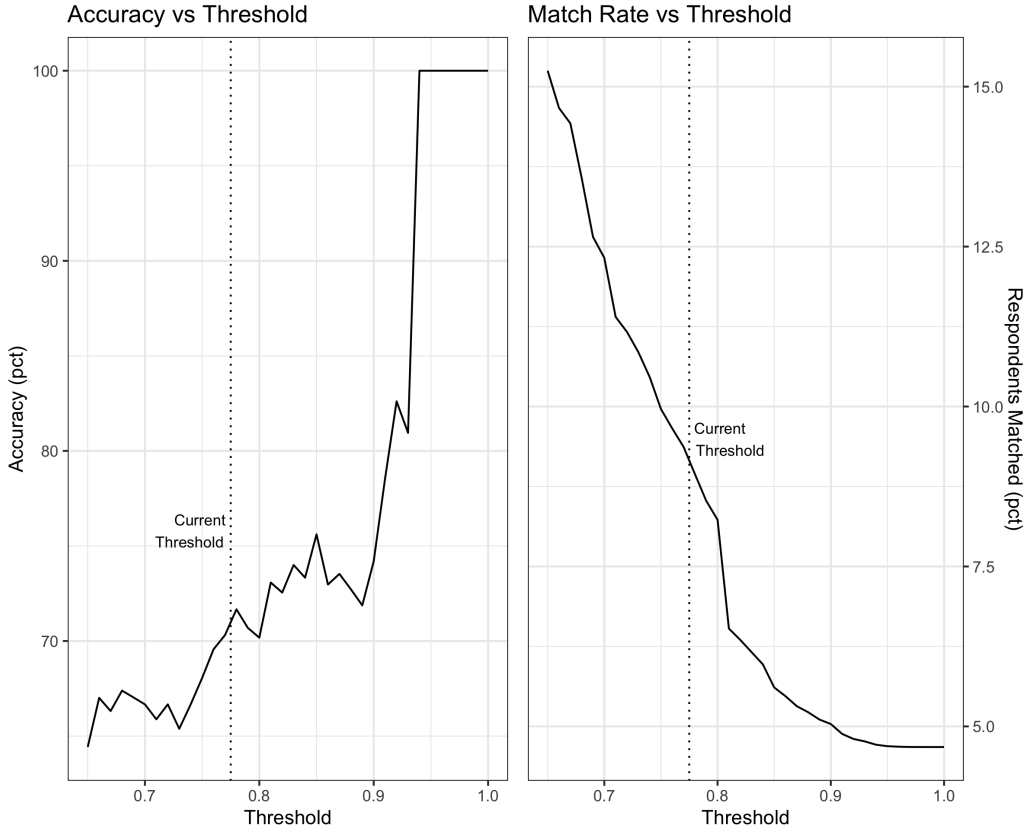


(b) Lemmatized Title Fuzzy Matching Plots

Figure 5-1: Fuzzy Matching Accuracy and Match Rate vs Threshold



(a) Original Title TFIDF Matching Plots



(b) Lemmatized Title TFIDF Matching Plots

Figure 5-2: TFIDF Matching Accuracy and Match Rate vs Threshold

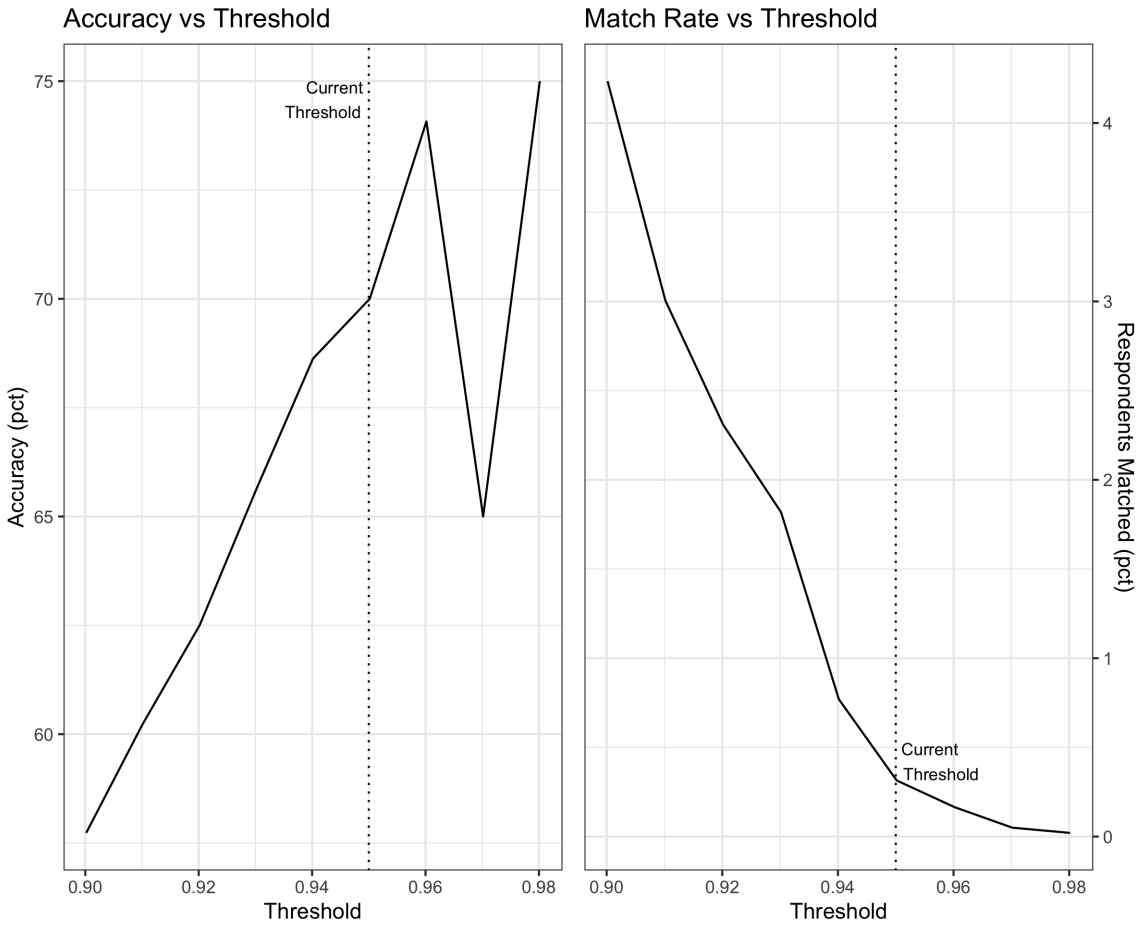


Figure 5-3: Word Embedding Matching Accuracy and Match Rate vs Threshold

5.1.3 Example Matches

In this section, we provide examples of the matches made by each of the different methods we use.

5.1.3.1 Exact Matches

Exact matches were the first part of the matching procedure. Given a title in the CCC, we match it to a title in the CAI that has an identical title or an identical alternative form. For example, we match two titles if they are identical after stemming them. The alternative forms of titles are useful for matching titles that are slightly mispelled or plural. Table 5.5 contains examples of exact matches.

Table 5.5: Sample Exact Matches

CCC Title	CAI Match	Match Type
tobacco raiser	tobacco raiser	Original Title
mariner	mariner	Original Title
boots bender	boot bender	Lemmatized Title
contractors	contractor	Lemmatized Title
p w a labor	p w a laborer	Stemmed Title
teachers assistent	teachers assistant	Stemmed Title
stenogenapher	stenographer	SpellFix Title
toletype operator	teletype operator	SpellFix Title
salesgen	salesman	Lemmatized SpellFix Title
teiloring	tailor	Lemmatized SpellFix Title

5.1.3.2 Industry Information Match

This variant of exact matching is helpful for matching titles in the CCC that were more detailed than their counterparts in the CAI. For example, the CCC title "attendant store" is matched to the CAI title "attendant" that appears in the industry "store".

Table 5.6: Sample Industry Information Matches

CCC Title	CAI Title	CAI Industry Information
attendant store	attendant	store
proprietor lodginghouse	proprietor	lodginghouse
owner restaurant	owner	restaurant
laborer school	laborer	school

5.1.3.3 Fuzzy Matches

This method is helpful for matching misspelled titles and titles with minor wording variations. We first runn the fuzzy matching algorithm using the original CCC title and then runn it using the lemmatized version of the CCC title. In a manual audit

of the matches, we found that this method's accuracy was 90%. Table 5.7 contains matches made using fuzzy matching on the original title as well as the lemmatized title.

Table 5.7: Sample Fuzzy Matches

CCC Title	**CAI Match**	**Title Form**	**Fuzzy Score**
ser station helper	station helper	Original	0.923
gas tester attendant	attendant	Original	0.843
construction mechanic	construction man	Original	0.898
rodman for wpa	rodman	Lemmatized	0.803
rublishing	publisher	Lemmatized	0.855
nursing publis health	nurse public health	Lemmatized	0.830

5.1.3.4 TF-IDF

This method is mostly helpful for matching incomplete titles. The main advantage of this method is that it computes the relative information provided by each word for each occupation code separately. For example, the word "clerk" conveys little information in the occupation group that contains only clerks, because all titles in that occupation have the word "clerk" in them. However, the same word provides a lot of information in the occupation group "Managers, proprietors, and owners" because very few titles in that occupation group contain the word "clerk".

Table 5.8: Sample TF-IDF Matches

CCC Title	**CAI Match**	**Title Form**	**TFIDF Score**
cafetener worker	worker	Original	1.000
flat work in laundry	laundry work	Original	0.770
helper torwster	helper	Original	1.000
code verifier	coding clerk	Lemmatized	0.939
dressing sales atkins	dress cutter	Lemmatized	0.872
billiet widind labor samer	laboring man	Lemmatized	0.905

5.1.3.5 Embedding

This method is useful for matching synonyms. For example, it matches "violin teacher" to "violin instructor". At the threshold we set, this method had an accuracy of 80%. However, we were only able to match 0.3% of respondents.

Table 5.9: Sample Embedding Matches

CCC Title	**CAI Match**	**Match Score**
violin instructor	violin teacher	0.961
water worker man	water service man	0.956
iron power	iron drawer	0.954
property merchant	property master	0.971
board charger	board filler	0.960

5.2 CCC-DOT Matching

Using the matching procedure we developed, we were able to match the jobs of 84% of respondents in our sample to jobs in the 1939 Dictionary of Occupational Titles. Additionally, we were able to find people working in 1,150 out of 1,544 unique new jobs and a total of 379,697 out of people working in those new jobs. This is 80% of all new workers identified in the CCC. The procedure we developed is able to match misspelled titles, incomplete titles, and synonyms of titles.

5.2.1 Challenges

The CCC-DOT matching task is significantly more challenging than the CCC-CAI matching task. One aspect that makes the match difficult is the different level of detail included in each dataset. Titles in the DOT tend to be more detailed than titles in the CCC or CAI. For example, the CAI has 133 unique titles containing the word "clerk", whereas the DOT has 679 titles containing the word "clerk". This may be due to the differing purposes of the CAI and the DOT. The CAI was created to be used by Census

officials to match respondents' job titles to Census occupation codes. The DOT, on the other hand, was created to help people discover jobs. Another aspect that makes the matching difficult is that the CCC and the DOT use different occupational coding schemes. To overcome this, we constructed a manual crosswalk, which can be found in the appendix. The Census occupation codes for the most part are independent of industry information, whereas many DOT codes are industry dependent. For example, the DOT occupation code "8-02" corresponds to "Occupations in production of beverages". The last complication is that 12% of titles in the DOT do not have occupation codes. Our procedure would not match a CCC title to any of these titles because we impose the restriction that occupation codes be equal.

5.2.2 Match Rate Progression

Exact matches of the original titles or their alternative forms accounted for 56.1% of respondents in our sample. Fuzzy matching yielded another 11.3% of respondents. 11% of respondents were matched using TF-IDF. The remaining 5.5% of respondents were matched manually. Table 5.10 shows how the number of workers matched evolves at each step of the procedure. Table 5.11 shows how the number of CCC titles matched evolves at each step of the procedure.

Table 5.10: Progression of workers matched to a DOT title

Match Type	**Absolute**		**Cumulative**	
	# of Workers	**# of New Workers**	**# of Workers**	**Share of Workers**
Exact	19,627,998	76,412	19,627,998	0.495
CAI Title Exact	2,475,969	86,619	22,103,967	0.557
Lemmatized Title Exact	13,241	1,170	22,117,208	0.558
Stemmed Title Exact	55,204	3,258	22,172,412	0.559
SpellFix Title Exact	48,774	392	22,221,186	0.560
SpellFix Lemma Exact	3,042	25	22,224,228	0.560
Full Title Exact	6	0	22,224,234	0.560
Full Title Lemma Exact	20	0	22,224,254	0.560
Stemmed Full Title Exact	12,340	0	22,236,594	0.561
Lemmatized Title Fuzzy	3,118,644	70,420	25,355,238	0.639
CAI Title Title Fuzzy	654,453	67,284	26,009,691	0.656
FullTitle Fuzzy	728,002	3,473	26,737,693	0.674
Title Lemma TF-IDF	3,954,919	44,864	30,692,612	0.774
Full Title Lemma TF-IDF	415,692	1,458	31,108,304	0.784
Manual	2,154,089	24,322	33,262,393	0.839

Table 5.11: Progression of CCC titles matched to DOT titles

Match Type	Absolute		Cumulative	
	# of Titles	# of New Titles	# of Titles	Share of CCC Titles
Exact	11,408	603	11,408	0.004
CAI Title Exact	455,490	10,444	466,898	0.171
Lemmatized Title Exact	582	27	467,480	0.172
Stemmed Title Exact	1,623	58	469,103	0.172
SpellFix Title Exact	11,809	229	480,912	0.177
SpellFix Lemma Exact	435	23	481,347	0.177
Full Title Exact	1	0	481,348	0.177
Full Title Lemma Exact	2	0	481,350	0.177
Stemmed Full Title Exact	16	0	481,366	0.177
Lemmatized Title Fuzzy	244,075	8,416	725,441	0.266
CAI Title Title Fuzzy	185,537	29,179	910,978	0.335
FullTitle Fuzzy	28,566	189	939,544	0.345
Title Lemma TF-IDF	354,725	8,231	1,294,269	0.475
Full Title Lemma TF-IDF	73,923	552	1,368,192	0.502
Manual	45	1	1,368,237	0.502

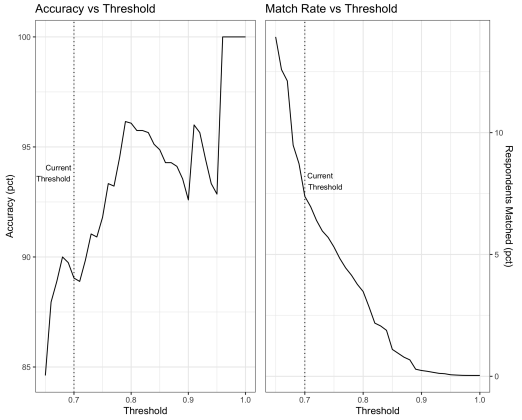
5.2.3 Accuracy and Match Rate of Approximate Methods

To evaluate the accuracy of the fuzzy matching and TFIDF matching, we randomly sampled 100 titles matched by each method and audited them manually. We generally chose the thresholds such that the accuracy was at least 90% for fuzzy matching and at least 80% for TF-IDF. Table 5.12 shows the accuracy of each method at the threshold we used. Fuzzy matching was generally the most accurate, except when used with the CAI Title. We believe this is because the CAI matching procedure is imperfect, which introduces extra noise. Fuzzy matching with the CCC title and with industry information had an accuracy of over 89%. Fuzzy matching with the CAI Title had

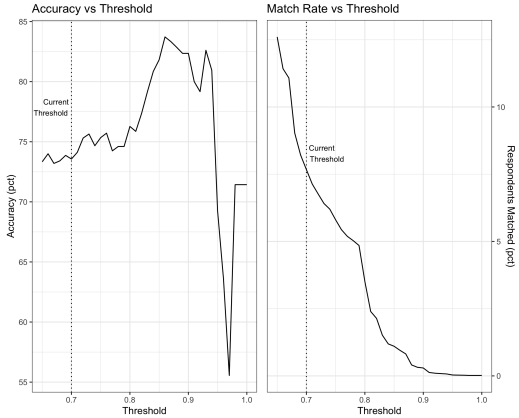
an accuracy of 74%. TFIDF matching using the lemmatized CCC title and industry information had accuracies of over 80%.

Table 5.12: Accuracy of Approximate Methods

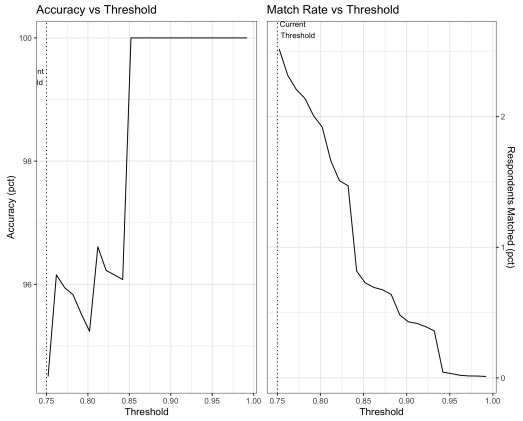
Match Type	**Accuracy**	**Share of Workers Matched**
Lemmatized Title Fuzzy	0.892	0.078
CAI Title Fuzzy	0.736	0.025
Full Title Fuzzy	0.945	0.019
Lematized Title TFIDF	0.876	0.095
Lemmatized Full Title TFIDF	0.806	0.016



(a) Lemmatized Title Fuzzy Matching Plots

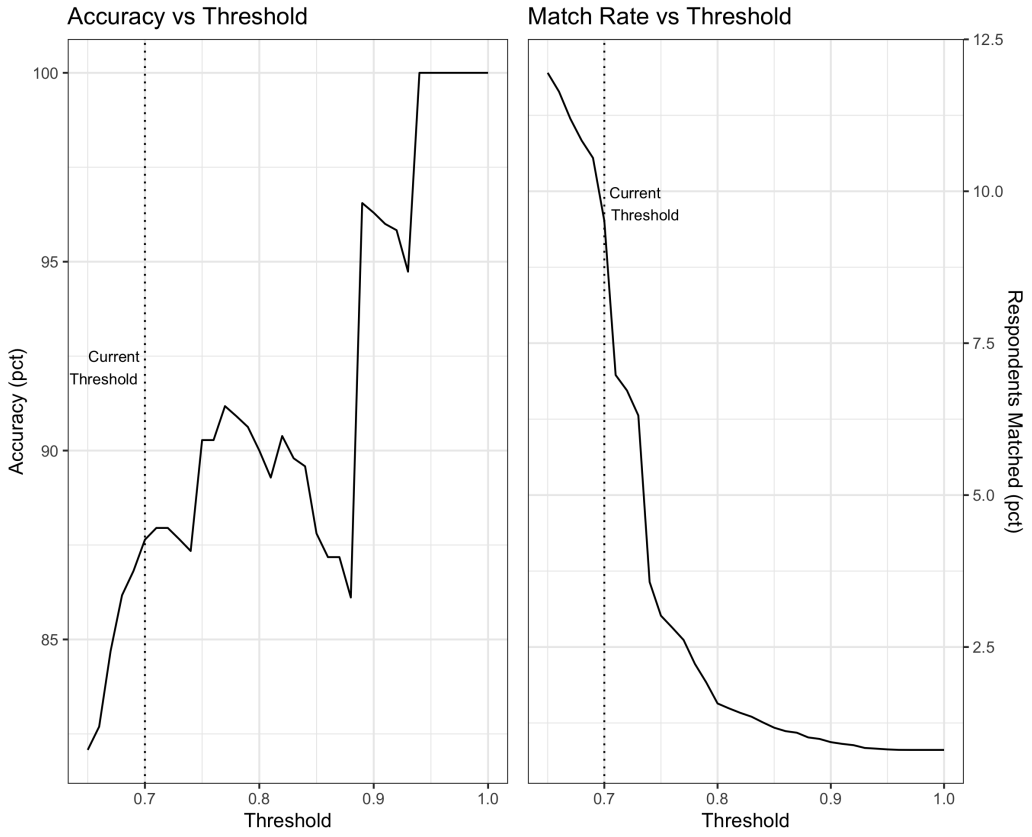


(b) CAI Title Fuzzy Matching Plots

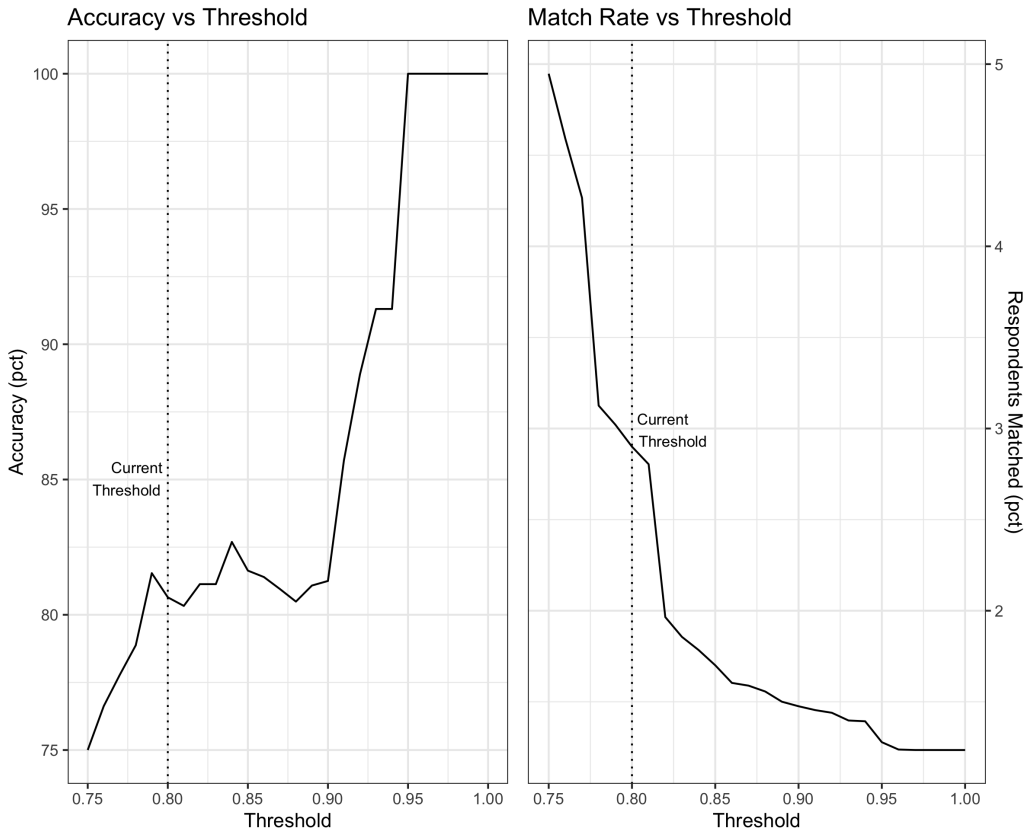


(c) Full Title Fuzzy Matching Plots

Figure 5-4: Fuzzy Matching Accuracy and Match Rate vs Threshold



(a) Lemmatized Title TFIDF Matching Plots



(b) Lemmatized Full Title TFIDF Matching Plots

Figure 5-5: TFIDF Matching Accuracy and Match Rate vs Threshold

5.2.4 Example Matches

In this section, we provide examples of the matches made by each of the different methods we use.

5.2.4.1 Exact Matches

Exact matches are the first part of the matching procedure. Given a title in the CCC, we match it to a title in the DOT that has an identical title or an identical alternative form. For example, we match two titles if they are identical after stemming them. The alternative forms of titles are useful for matching titles that were slightly mispelled or plural. Table 5.13 contains examples of exact matches.

Table 5.13: Sample Exact Matches

CCC Title	**CAI Title**	**DOT Match**	**Match Type**
spring inspector	spring maker	spring inspector	Original Title
motor assembler	motor assembler	motor assembler	Original Title
pina inspector	inspector	inspector	CAI Title
foundry president	president	president	CAI Title
rolling mill man		roll mill man	Lemmatized Title
asbestos workers helper	p w a worker	asbestos worker helper	Lemmatized Title
woods cutter	cutters helper	wood cutter	StemmedTitle
compose		composer	StemmedTitle
waltcher		watcher	SpellFix Title
dnawing tender	tender	drawing tender	SpellFix Title
mill toremen		mill foreman	SpellFix Lemma
salokers helper	helper	smoker helper	SpellFix Lemma

5.2.4.2 Exact Full Title Matches

As mentioned in the methods chapter, we also use the industry information in the CCC-DOT matching to address the different level in detail of the titles. Given a title

in the CCC, we create a "full title" by appending the industry information available in the CCC. For example, if a respondent has the title "salesman" and the industry code "082", corresponding to "advertising", their full title is "salesman advertising". We also create alternative forms of the full title and use them to match to the DOT. Table 5.14 contains examples of exact full title matches.

Table 5.14: Sample Exact Full Title Matches

CCC Title	**Industry**	**DOT Match**	**Match Type**
interviewer	insurance	interviewer insurance	Full Title
salesmen	real estate	salesman real estate	Lemmatized Full Title
salesmans	real estate	salesman real estate	Stemmed Full Title
management	agriculture	manager agricultural	Stemmed Full Title

5.2.4.3 Fuzzy Matches

This method is helpful for matching misspelled titles and titles with minor wording variations. We first run the fuzzy matching algorithm using the original CCC title, then run it using the lemmatized version of the CCC title, and lastly run it using the full title. Table 5.15 contains matches made using fuzzy matching on the original title as well as the lemmatized title.

Table 5.15: Sample Fuzzy Matches

CCC Title	DOT Match	Title Form	Score
street light maintenance	electrician maintenance	Lemmatized Title	0.837
tippist	typist	Lemmatized Title	0.718
traffic mgr	traffic man	Lemmatized Title	0.953
mine operting	laborer	CAI Title	0.753
hand wearing	hander	CAI Title	0.716
supatement manager	manager department	CAI Title	0.859
cioil cig	construction engineer	Full Title	0.754
asst to director of war service	organizer	Full Title	0.845
laborer opeartor	telephone operator	Full Title	0.846

5.2.4.4 TF-IDF

This method is mostly helpful for matching incomplete titles. The main advantage of this method is that it computes the relative information provided by each word for each occupation code separately. For example, the word "clerk" conveys little information in the occupation group that contains only clerks, because all titles in that occupation have the word "clerk" in them. However, the same word provides a lot of information in the occupation group "Managers, proprietors, and owners" because very few titles in that occupation group contain the word "clerk".

Table 5.16: Sample TF-IDF Matches

CCC Title	DOT Match	Title Form	Score
city desk	desk clerk	Lemmatized Title	0.941
laboratory experimental	experimental man	Lemmatized Title	0.705
band musician	musician instrumental	Lemmatized Title	0.707
budget manager	manager retail store	Lemmatized Full Title	1.000
prop tasern	domestic	Lemmatized Full Title	1.000
practical chemist	domestic	Lemmatized Full Title	1.000

5.3 Job Attributes Prediction

We trained models to predict job attributes using data from the 1965 DOT and 1991 DOT. We evaluated these models using a test set from the same year they were trained on and using the other year's data. Our models were able to accurately predict all attributes for data from the year they were trained on. They were also able to accurately predict most attributes for data from different years. The number of possible labels changed between 1965 and 1991. As a result, for some attributes, our model's predictions have low accuracy but strong correlation with the true value. This is true for all attributes except for eye-hand-foot coordination and finger dexterity.

We examined these two attributes more closely by looking at title definitions that were nearly identical in 1965 and 1991. For example, the definition for "Music Teacher" is almost identical in 1965 and 1991, however, the title's finger dexterity and eye-hand-foot coordination scores differ significantly in the two years. This leads us to believe that the method for assigning finger dexterity and eye-hand-foot coordination scores changed between 1965 and 1991.

5.3.1 Training Year

We decided to use the model trained on the 1965 DOT for three reasons. First, it is the closest to 1939 out of all of the labeled data. Second, the structure and definitions of the 1965 DOT are more similar to the 1939 DOT than the 1991 DOT. And lastly, the 1965 model's predictions of the 1939 DOT's attributes had more significant variation than the 1991 model's predictions. For an example of this, see Figure 5-6.

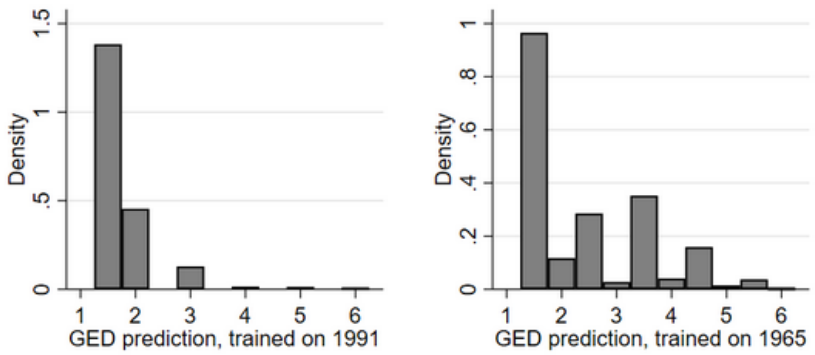


Figure 5-6: 1991 Model vs 1965 Model GED Predictions on 1939 Data

5.3.2 Quantitative Evaluation

We use accuracy, correlation, rank correlation and mean absolute error to evaluate our model's predictions. The model had high accuracy when evaluated on the test set from the same year it was trained on. The labels for some categories change across years, which hurts accuracy. For example, GED has 6 labels in 1991 and 9 in 1965. In our tables, we group together attributes based on whether or not their labels changed across years. Our model was able to accurately predict all attributes whose values did not change across years. For attributes whose values did change across years, our model's predictions were strong correlated with the true values, except for finger dexterity and eye-hand-foot coordination. Table 5-7 contains the model's performance on the 1965 test set, Table 5-8 contains the model's performance on the 1977 data, and Table 5-9 contains the model's performance on the 1991 data.

5.3.2.1 Within Year Performance

Figure 5-7: Performance on 1965 Test Set

Attribute	Accuracy	Correlation	Rank Correlation	MAE	Obs.
Data	0.841	0.890	0.879	0.353	2468
People	0.945	0.899	0.922	0.158	2468
Things	0.775	0.759	0.744	0.687	2468
DCP	0.959	0.838	0.838	0.041	2468
STS	0.809	0.641	0.641	0.191	2468

(a) Attributes with stable values across years

Attribute	Accuracy	Correlation	Rank Correlation	MAE	Obs.
GED	0.731	0.900	0.883	0.246	2468
SVP	0.674	0.857	0.852	0.552	2468
EHFCoord	0.798	0.547	0.603	0.180	2468
FingerDext	0.855	0.645	0.633	0.141	2468

(b) Attributes with changing values across years

5.3.2.2 Across Year Performance

Figure 5-8: Performance on 1977 Test Set

Attribute	Accuracy	Correlation	Rank Correlation	MAE	Obs
Data	0.783	0.872	0.855	0.471	11832
People	0.815	0.850	0.777	0.420	11832
Things	0.800	0.740	0.727	0.686	11832
DCP	0.920	0.771	0.771	0.080	3608
STS	0.758	0.539	0.539	0.242	3608

(a) Attributes with stable values across years

Attribute	Accuracy	Correlation	Rank Correlation	MAE	Obs
GED	0.053	0.780	0.788	1.043	3608
SVP	0.233	0.833	0.837	0.920	3608
EHFCoord	0.394	0.333	0.273	0.486	3608
FingerDext	0.177	0.446	0.434	0.515	3608

(b) Attributes with changing values across years

Figure 5-9: Performance on 1991 Test Set

Attribute	Accuracy	Correlation	Rank Correlation	MAE	Obs
Data	0.780	0.870	0.860	0.476	12741
People	0.804	0.840	0.773	0.449	12741
Things	0.797	0.739	0.724	0.695	12741
DCP	0.951	0.833	0.833	0.049	12741
STS	0.741	0.483	0.483	0.259	12741

(a) Attributes with stable values across years

Attribute	Accuracy	Correlation	Rank Correlation	MAE	Obs
GED	0.178	0.762	0.786	0.851	12741
SVP	0.249	0.840	0.837	0.933	12741
EHFCoord	0.294	0.230	0.152	0.505	12741
FingerDext	0.121	0.373	0.347	0.533	12741

(b) Attributes with changing values across years

5.3.3 Qualitative Evaluation

We also used histograms to compare the distributions of the predicted attributes with the actual attributes. Overall, there is significant overlap between the distribution of the model's predictions and the true distribution of attribute values. This is especially true for attributes whose labels didn't change across years. We also show the

5.3.3.1 1965 Histograms

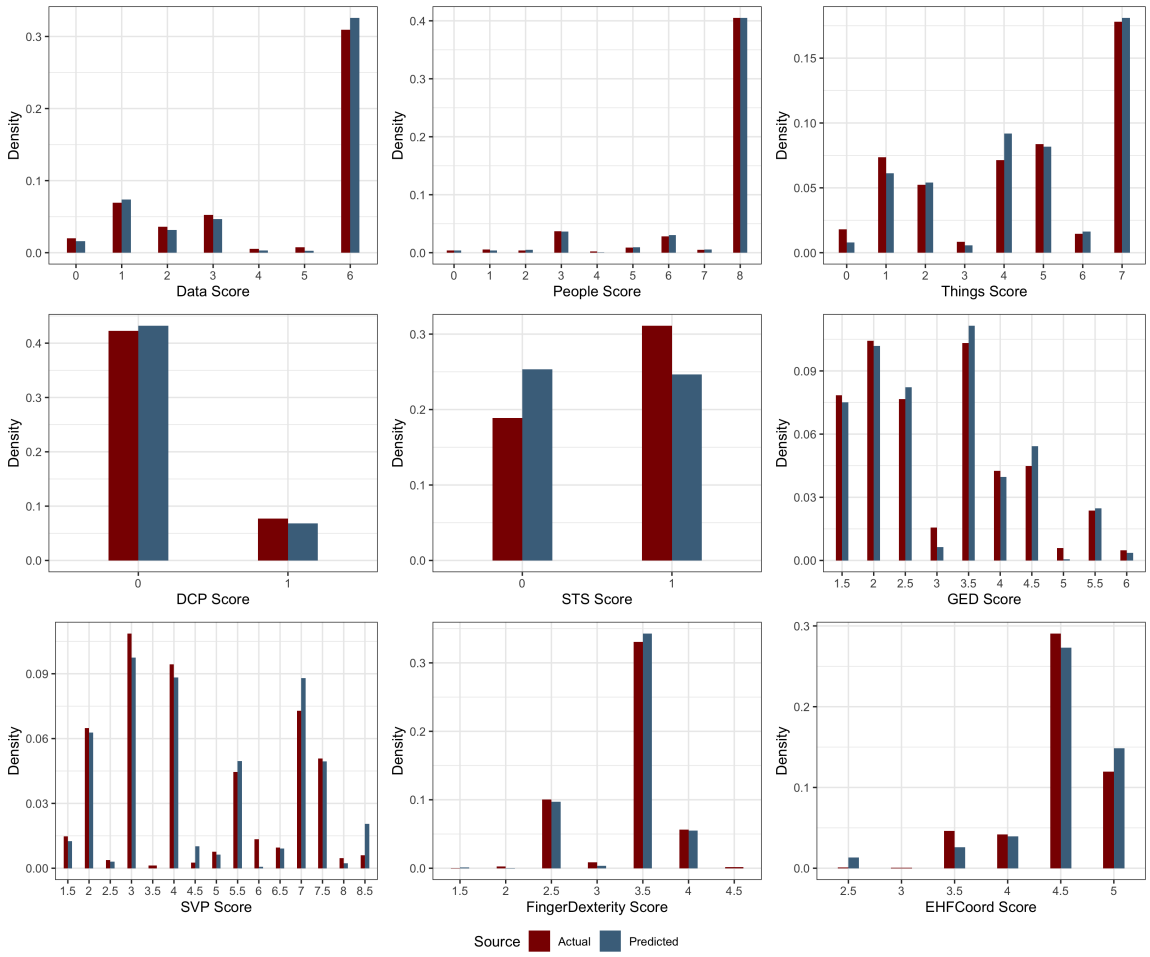


Figure 5-10: 1965 Predicted vs Actual Values on Test Set

5.3.3.2 1977 Histograms

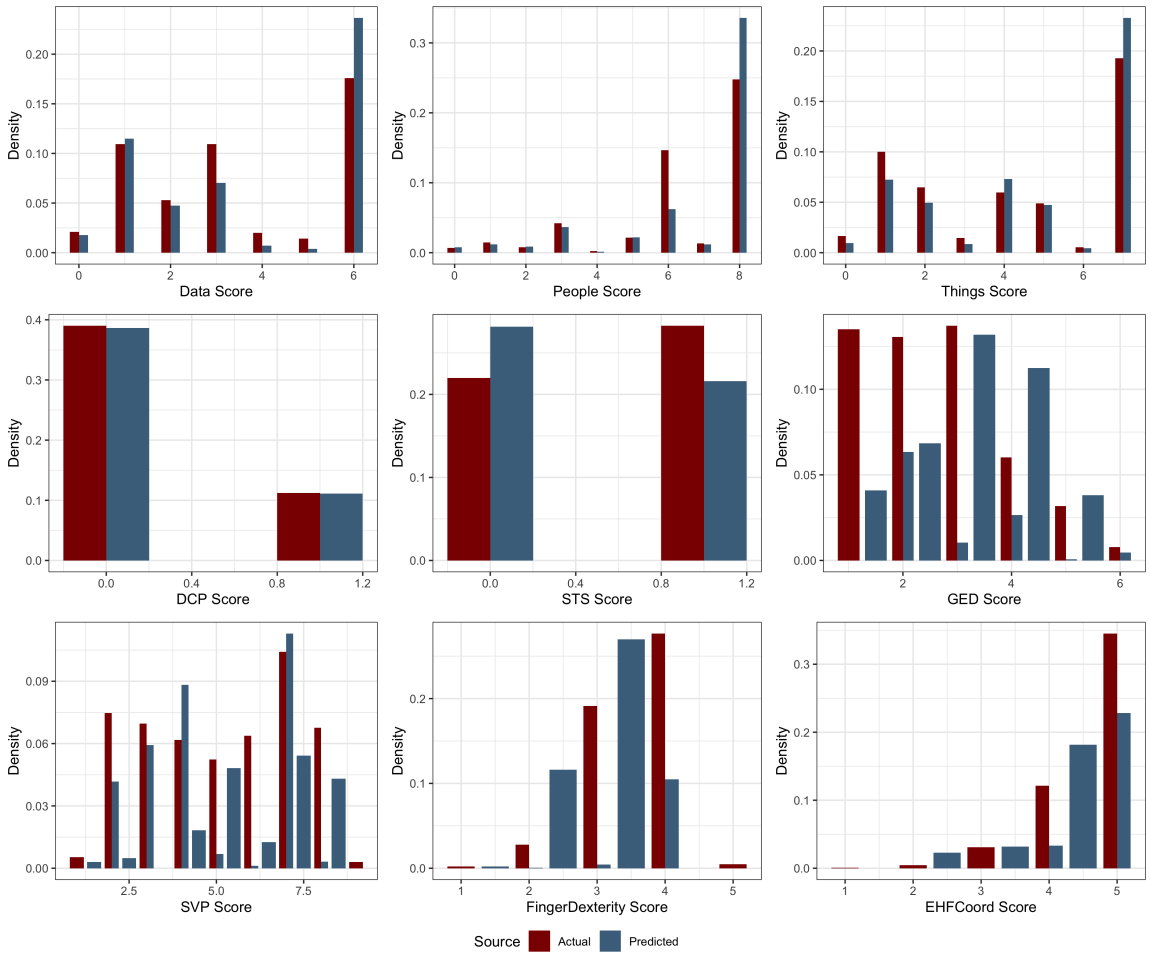


Figure 5-11: Predicted vs Actual Values on 1977 Data

5.3.3.3 1991 Histograms

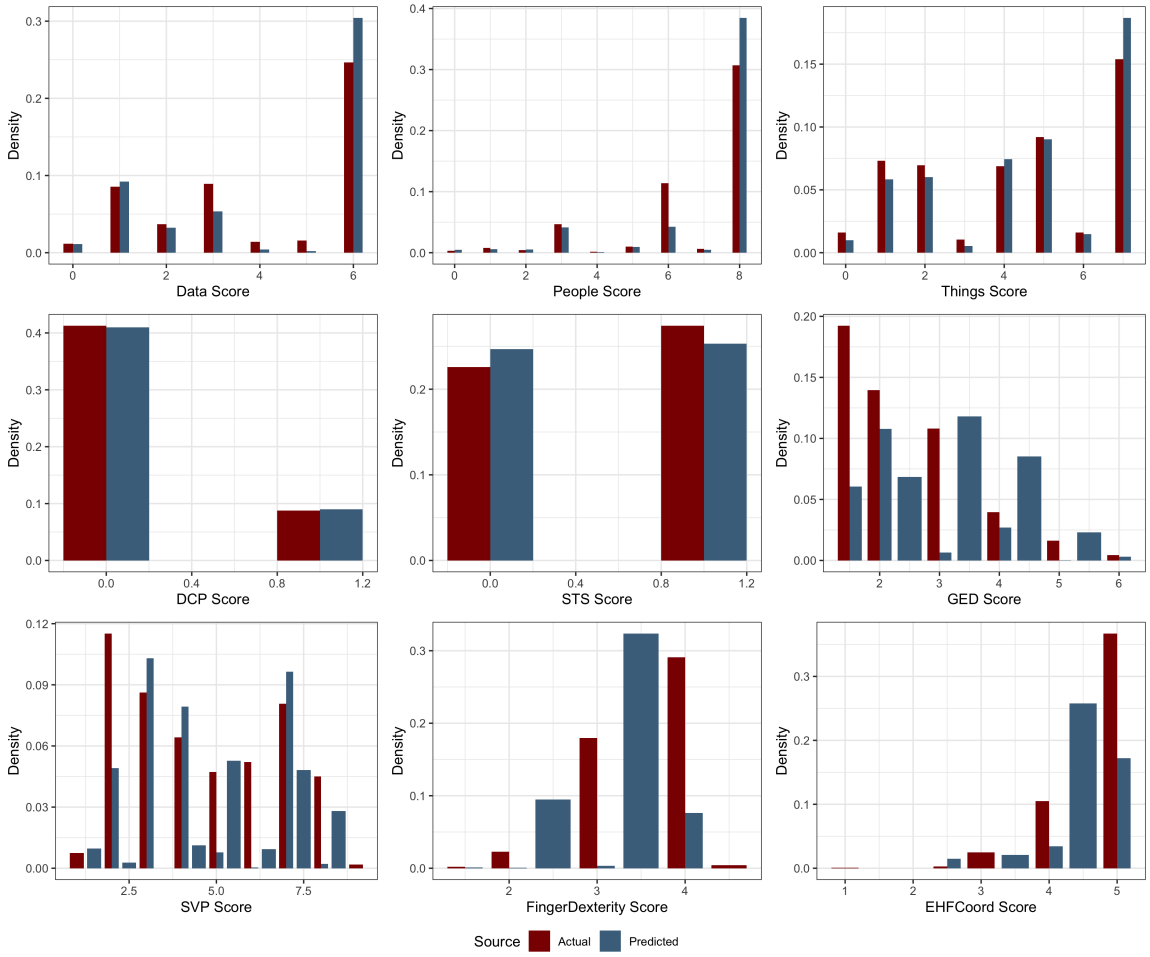


Figure 5-12: Predicted vs Actual Values on 1991 Data

5.3.3.4 1939 Histograms

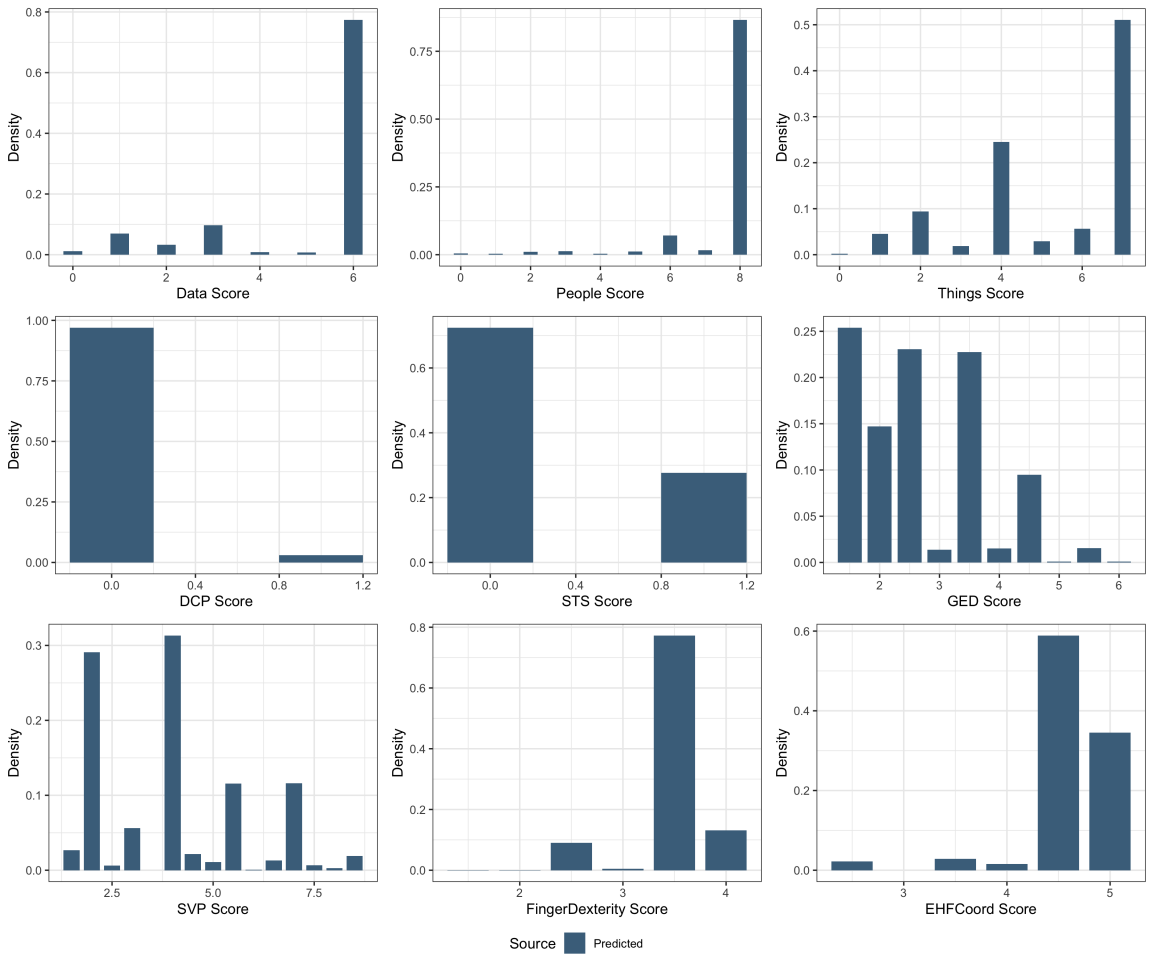


Figure 5-13: Histogram of Predicted Values for 1939 Data

Chapter 6

Conclusion

This work shows how to construct a novel dataset that can be used to study the characteristics of new workers and new work in 1940. I used natural language processing techniques to match job titles in the 1940 Census Complete Count files and job titles in the 1940 Census Alphabetical Index of Occupations. I used similar techniques to match job titles in the complete count files to job titles in the 1939 Dictionary of Occupational titles to obtain textual descriptions for jobs. Finally, I created a natural language processing system that uses the textual descriptions of jobs to predict task content and job complexity. I used the different editions of the Dictionary of Occupational Titles to train and evaluation this system. The three contributions of this work can be combined to create a dataset that contains measures of job complexity and task content for over 80% of workers in the 1940 Census and identifies new workers in that year. The methods used in this work are generalizable and can hopefully be used to create other datasets that are useful for social science research.

Bibliography

[AA11] Daron Acemoglu and David Autor. "Skills, tasks and technologies: Implications for employment and earnings". In: *Handbook of labor economics*. Vol. 4. Elsevier, 2011, pp. 1043–1171.

[ALM03] David Autor, Frank Levy, and Richard J Murnane. "The skill content of recent technological change: An empirical exploration". In: *The Quarterly journal of economics* 118.4 (2003), pp. 1279–1333.

[Ang+17] Joshua Angrist et al. *Inside job or deep impact? Using extramural citations to assess economic scholarship*. Tech. rep. National Bureau of Economic Research, 2017.

[AR17] Daron Acemoglu and Pascual Restrepo. "Robots and Jobs: Evidence from US Labor Markets." In: (2017).

[AR18] Daron Acemoglu and Pascual Restrepo. "The race between man and machine: Implications of technology for growth, factor shares, and employment". In: *American Economic Review* 108.6 (2018), pp. 1488–1542.

[AS19] David Autor and Anna Salomons. "New Frontiers: The Evolving Content and Geography of New Work in the 20th Century". In: (2019).

[Ath18] Susan Athey. "The impact of machine learning on economics". In: *The economics of artificial intelligence: An agenda*. University of Chicago Press, 2018, pp. 507–547.

[Dev+18] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[GS10] Matthew Gentzkow and Jesse M Shapiro. "What drives media slant? Evidence from US daily newspapers". In: *Econometrica* 78.1 (2010), pp. 35–71.

[Lin11] Jeffrey Lin. "Technological adaptation, cities, and new work". In: *Review of Economics and Statistics* 93.2 (2011), pp. 554–574.

[Liu+19] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[Mik+13] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[Rug+10] Steven Ruggles et al. "Integrated public use microdata series: Version 5.0 [Machine-readable database]". In: *Minneapolis: University of Minnesota* 42 (2010).

[Vas+17] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[Wol+19] Thomas Wolf et al. "HuggingFace's Transformers: State-of-the-art Natural Language Processing". In: *ArXiv* abs/1910.03771 (2019).