Disambiguating American Federal Election Campaign Contributions

Guy Mark Lifshitz

Master of Science

School of Computer Science

McGill University

Montreal,Quebec

2014-04-15

A thesis submitted to McGill University in partial ful llment of the requirements of the degree of Master of Science

DEDICATION

This document is dedicated first, to all those trying to increase the transparency of democratic institutions across the globe. Secondly, it is dedicated to my family for supporting me through my studies. Elle est aussi dédier à Lucas Morin et Renaud Barne (vous savez pourquoi vous le méritez). I would like to extend much thanks to Jimmy Gutman for the unwavering moral support. I also wish to thank Daphnne Chacon, Jerem Cyr, and Stephen Brophy, for taking the time to recognize the value of this work. Finally, I must reserve some special gratitude for Mark Bay, for sharing a given name, and my unwavering commitment to research.

ACKNOWLEDGEMENTS

I would like to first acknowledge my supervisor Prof. Derek Ruths, without whom I would not have had the opportunity to fuse my passions for both politics and computer science. Secondly, I would like to acknowledge the hard work of our collaborators at Northeastern University, David Lazer, Sasha Goodman, and Navid Dianati. This work was supported by a grant from the NSF Program in Sociology.

ABSTRACT

A massive database of all financial contributions made by individuals to American political campaigns since the 1970s exists and is publicly available. If not for widespread spelling and naming inconsistencies and, of greater concern, the absence of unique identifiers for individual contributors in this database, it would be an unparalleled resource for academic and professional analysis of political behavior in the United States. We have developed a deduplication system which links together the contributions made by single individuals, effecively creating contributor identifiers. Of interest to both the entity resolution community, and those working on crowdsourcing, our disambiguation system makes use of computer algorithms, as well as human computation via *Amazon Mechanical Turk* and domain experts. Our paper makes three contributions: (1) showing how a disambiguation system can incorporate both computer and human computation to this sub-domain, (2) showing how to reduce a large dataset to reasonable sized subsets for efficient entity matching, and (3) providing a disambiguation of a decade's worth of contributions occurring in Delaware.

ABRÉGÉ

Une base de données groupant toutes les contributions individuelles aux campagnes électorales américaines depuis les années 1970 existe et est accessible au public. Elle pourrait constituer une ressource inestimable pour l'analyse académique comme professionnelle des comportements politiques aux Etats-Unis, mais souffre d'erreurs de dénomination et manque d'un moyen pour identifier individuellement les contributeurs figurant dans la base de données. Nous avons développé un système de déduplication qui relie les contributions effectuées par des particuliers, créant de manière effective des identifiants pour chaque contributeur. Notre système de désambiguïsation, utilisant des algorithmes informatiques, de la calculation via le Turc Mécanique d'Amazon et des experts, est susceptible d'intéresser la communauté des chercheurs travaillant sur la résolution d'entités et le crowdsourcing. Notre document présente trois contributions. Il montre (1) comment un système de désambiguïsation peut intégrer des calculs humains et informatiques à la fois, (2) comment réduire une immense base de données à des sous-ensembles de taille raisonnable pour les relier efficacement, et (3) apporte une désambiguïsation d'une décennie de contributions ayant eu lieu au Delaware.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

The American political system is influenced by many factors. These include, ballot votes on election day, lobbying from special interests, and campaign contributions. Understanding each of these components helps political scientists better study the political system, while also helping ensure fairness in the democratic process of governing. In this thesis we focus specifically on increasing the transparency of political contributions from individual donors in the United States of America (US). We achieve this by resolving the unique identities found in a large, public, database of contributions to American political parties. We refer to this process as "disambiguation".

To further motivate the importance of this work, consider that the Democratic Congressional Campaign Committee recommends members of congress spend four hours a day making calls to potential donors, and often donors will spend thousands of dollars to attend a candidate's fund-raiser with the hopes of gaining brief access to the candidate [13]. Also consider, that in the 2012 elections, 93% of congressional races were won by the candidate who spent the most money, consistent with similar percentages from other recent elections [34]. The average cost to win a seat in the House of Representatives was $652,000, and $2.8 million for the Senate [13]. All these statistics signal that financial considerations are an important element in the political system.

In 1974, in the wake of the Watergate scandal, the US congress passed a campaign finance law increasing the oversight of financial activity in the political system. Amongst its many reforms, the law requires that campaign contributions over a certain threshold be recorded. Today each contribution

over $200 must be declared. The law also placed a cap on the total donations an individual can make in a campaign cycle.

All data recorded since the passing of the campaign finance law is available on the Federal Election Commission's (FEC) website[1] . The dataset contains hundreds of millions of records, containing data about each transaction's donor and receiver. The decision to record all this data was a step in the right direction for transparency. However, despite appearing, at first sight, as a highly attractive dataset for political science research, the data has not been heavily used. This is because it is very difficult to determine which contribution records belong to the same individual contributor. The dataset lacks unique identifiers linking together all the contributions made by a single individual. The only identifying information about the source of a contribution is the contributor's name, occupation, employer, and address. The fields are all self-reported, and rife with spelling-errors and inconsistencies. Complicating matters further, the dataset spans decades. In this time period individuals may change jobs, move, get married, and perform other activities that change the way they report values for these fields. All of these changes will negatively impact our ability to easily compare two records for similarity.

Moreover, the sheer size of the dataset calls for well thought-out, efficient data-processing. Since there are millions of records dating from the last decade alone, it is impossible to do an $O(N^2)$ comparison between each set of records. We must find an approach that does not require a comparison between each possible pair of records, but rather a way of breaking the data down into smaller sets. This thesis describes such a system.

[1] Detailed Files About Candidates, Parties and Other Committees, Federal Election Commission, http://www.fec.gov/finance/disclosure/ftpdet.shtml

We are not the first group to attempt to disambiguate this dataset. Other teams have focused on annotating smaller portions of the dataset by hand, or have used fuzzy string matching between records [27, 6]. These methods are not rigorously described, and as we will show, there are questions as to the quality of their final results. All the teams who have worked with the data so far have been political scientists with limited experience dealing with such large datasets. Their techniques are often not as methodically sound as those found in fields accustomed to working with "big data". To solve the disambiguation problem, domain knowledge from both Computer Science, and Political Science was needed. We collaborated with political scientists at Northeastern University to combine our understandings from both fields, and better tackle the problem together. The work we present in this paper is a more systemic approach than any attempted so far.

We break down our approach into several phases similar to those commonly practiced in the data matching field[10]. We start with a pre-processing phase where we clean up obvious mistakes in the database. We remove unusable records, such as those with missing names. In this phase we also normalize data so it can be easily compared. We then proceed to use computer algorithms to separate the data into increasingly smaller subsets using information about the each record's reported name, employer, occupation, and address. With each split, the number of total pairwise comparisons needed to perform the next split decreases, making efficient runtime possible. Finally, we assign contributor identities to these sets.

Unfortunately, the reporting of fields can vary wildly, and we found it very difficult to find all records matching the same individual using computer algorithms alone. Hence, we employ human computation using crowdsourcing

to identify similar entries that were not easily detected using computers. Finally, records that were not matched by the crowd are analyzed more closely by a trained expert who compared record entries with online sources of information, to find evidence that the records belong to the same, or different, individuals.

To prove that our system works, we performed a complete disambiguation of contributions in Delaware spanning a decade. We found that the vast majority of identities could be determined using computation alone, and those that needed to be processed by the crowd were generally correct. Our trained expert was unable to make substantial improvements on the crowd's findings, showing that computer and crowd computation were able to discover most of the high-confidence identities. Once we had a complete disambiguated dataset, we performed analysis to show clear patterns in the concentration of wealth in the political system.

With our disambiguation of Delaware we have proved that our system works, and the same process can easily be replicated to other states.

CHAPTER 2
Background

2.1 FEC Data

The US Congress created the Federal Election Commission (FEC) to administer and enforce the 1974 Federal Election Campaign Act, the statute that governs financing of federal elections. The FEC's duties are to "disclose campaign finance information, enforce the provisions of the law such as the limits and prohibitions on contributions, and oversee the public funding of Presidential elections" [14].

To complete its mandate, the FEC records, and makes available, data about financial transactions which exceed $200. The resulting dataset does not include a unique identifier, meaning that all records produced by the same individual are unlinked. Linking the records produced by a single person would be a great benefit to political scientists, as well as government watchdogs like *OpenSecrets.org*. Using this data, they could better understand the political system, and find cases of fraud.

Cases of fraud have been found thanks to the FEC data. In January 2014, the National Rifle Association (NRA) was charged with violating electoral law by failing to properly declare contributions. The fraud was not discovered by the FEC, but by Brown student Sam Bell, who had taken personal initiative to cross-reference the NRA's campaign expenditure reports [52]. Disambiguating the FEC dataset will almost certainly reveal many more cases of fraud.

2.1.1 Prior Work

Unfortunately, rigorous work on disambiguating the FEC contributor data has been rare, generally focusing on small partitions of the data, or using

poorly-documented methods [27, 6]. So far, the best attempt at disambiguating the data came from political scientist Adam Bonica. In 2013, Bonica released a *Database on Ideology, Money in Politics, and Elections* (DIME), intended as "a general resource for the study of campaign finance and ideology in American politics." The DIME features a disambiguated version of the FEC dataset dating back to 1979. Creating the dataset took "a large-scale effort to compile, clean, and process data on contribution records." Bonica cleaned up and standardized names, address, and occupations and employers and gave unique identifiers to individual donors [6].

Although a large portion of the Bonica dataset seems correct, we noticed inconsistencies. For example, some entities were not merged into the same identity, either due to typos, or for reasons beyond our understanding. Some records were linked despite not sharing enough fields to indicate that they represent the same individual. For more details and specific examples, see Tables 2–1 and 2–2.

Having shown the limitations of previous attempts to disambiguate the FEC contributor dataset, we believe that there is still important work to be done in this field.

2.2 Contributor Information

As mentioned, the FEC collects certain information about contributors. Most of the fields are not beneficial to disambiguation, but some are. The field which we felt were beneficial to constructing identities were name, address, occupation, and employer. We now discuss these fields, and the methods used to collect them, in more detail.

2.2.1 Data Collection

A strong understanding of the FEC's data collection process is beneficial to understanding how to work with its data. With that in mind, we now

Table 2–1: Examples of errors in the DIME: the pairs of records listed here should have been merged.

These were not linked - a misspelling (Richard vs. Ryszard) caused the split.

Last Name	First Name	Occupation	Employer	Mailing Address	ZIP code	State
Zyla	Richard	Co-Owner	J-Z Construction Corp.	217 E 7th Street	11378	NY
Zyla	Ryszard	President	J.Z. Renovation Construction	207 E 7th St.	11378	NY

These John Miller entries were not linked - though occupation and employer match, which suggests a move.

Last Name	First Name	Occupation	Employer	Mailing Address	ZIP code	State
Miller	John	Diagnostic Radiol-ogis	Self-Employed	6235 Park W Dr	77706	TX
Miller	John	Diagnostic Radiol-ogis	Self-Employed	905 Wern Ave	70401	LA

Table 2–2: Examples of errors in the DIME: the pairs of records listed here were incorrectly linked to the same contributor identity.

These two "John Creighton"s were linked. They are possibly the same individual, though no strong indication comes from career or address.

Last Name	First Name	Occupation	Employer	Mailing Address	ZIP code	State
Creighton	John	Retired	Retired	3711 130th Ave NE	98005	WA
Creighton	John	Attorney	Preston Gates & Ellis	907 14th Ave	98122	WA

A Maryland "Mary Bush" got matched with two "Mary O Bush"s, for no apparent reason.

Last Name	First Name	Occupation	Employer	Mailing Address	ZIP code	State
Bush	Mary		Retired	3509 Woodbine St	20815	MD
Bush	Mary O	Real Estate	Self-Employed	PO 1546	33475	FL
Bush	Mary O	Real Estate	Self-Employed	PO 1546	33475	FL

These were linked, but there is no reason to think they should be.

Finnell	Michael H	Investment	Mit Asset Management	51 Burning Tree Rd	6830	CT
Finnell	Michael H		Self-Employed	724 Holladay Rd	91106	CA

describe the journey each record takes from time of collection, to placement on the FEC's website, and finally into the filtered dataset we work with.

In the US, candidates and parties must raise and spend campaign contributions through entities known as campaign committees. These committees are registered with the FEC and are subject to disclosure requirements. Amongst these requirements is a quarterly report listing details on campaign donations exceeding $200. For each contribution, details must include, amongst other information, the contributor's name, address, occupation, and employer. Each committee asks contributors to self-report these fields at the time a when donation is made. Reports made by the contributor may be hand written, or digital. In the case of a hand-written report, the committee is generally required to transcribe the data into a digital format before sending to the FEC, usually using OCR technology, or a data entry clerk. The FEC collects the quarterly reports from all committees and aggregates it into a dataset available for download on their website, along with documentation about each field [14].

Although the contributor data is available on the FEC website, the published dataset does not include contributor addresses. Instead, we used a dataset which did contain addresses, provided by our collaborators in the *Computational Social Science Lab* at Northeastern University. By comparing the published contributor dataset with raw FEC data files, the Northeastern team was able to cross-link contribution information and recover addresses. The end result was a dataset representing contributions made between 2003 and 2013, in select states. The work in this thesis shows a disambiguation for contributions in Delaware, but data from New York State was also used to demonstrate certain portions of our algorithm in a more populous state.

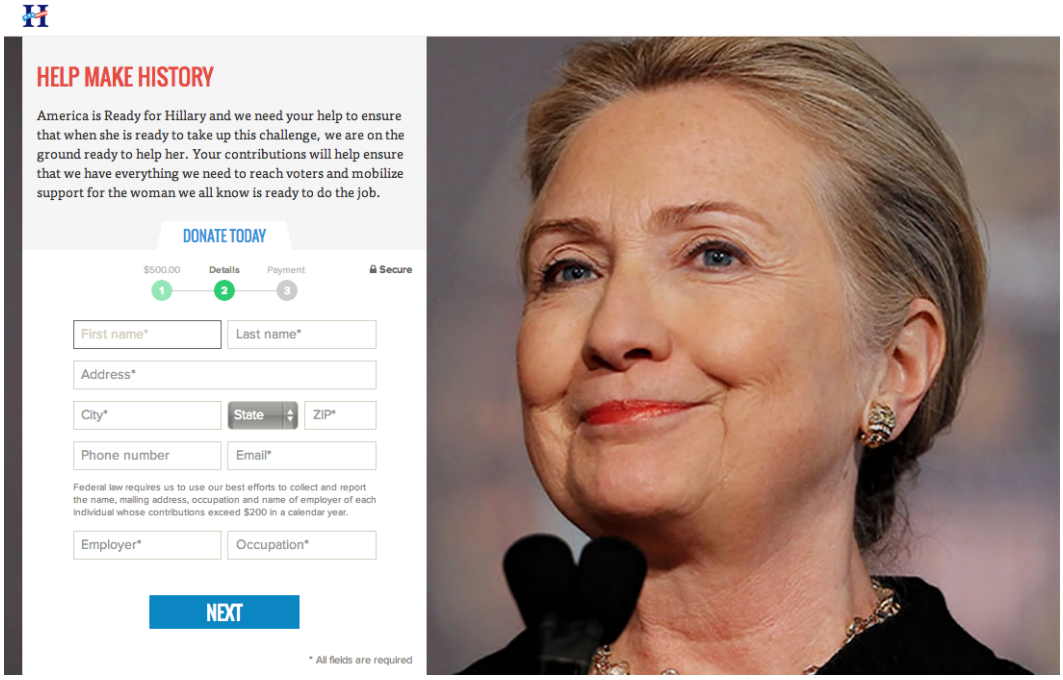

Figure 2–1: An online contribution form, showing the required fields: name, address, city, zip, employer, and occupation. Extra fields such as e-mail are not required by the FEC. Extra fields are collected by the campaign for other purposes, such as collecting demographic information on donors, or sending updates to donors.

Image# 11972379886

SCHEDULE A-P
ITEMIZED RECEIPTS

Use separate schedule(s) for each category of the Detailed Summary Page

FOR LINE NUMBER: (check only one)
16 [ ] 17a [X] 17b [ ] 17c [ ] 17d [ ] 18
19a [ ] 19b [ ] 20a [ ] 20b [ ] 20c [ ] 21

PAGE 34032 OF 72081

Any information copied from such Reports and Statements may not be sold or used by any person for the purpose of soliciting contributions or for commercial purposes, other than using the name and address of any political committee to solicit contributions from such committee.

NAME OF COMMITTEE (In Full)
Obama for America

A. Full Name (Last, First, Middle Initial)
Ms. Geraldine Clinton

Mailing Address 2101 Connecticut Ave NW

City: Washington
State: DC
Zip Code: 20008-1728

FEC ID number of contributing federal political committee: C

Name of Employer: Not Employed
Occupation: Retired

Receipt For: 2008
[X] Primary [ ] General
[ ] Other (specify) ▼

Election Cycle-to-Date ▼ 1200.00

Transaction ID : 4178964

Date of Receipt: 06 / 30 / 2008

Amount of Each Receipt this Period: 200.00

Figure 2–2: A contribution record as saved on the FEC's servers. Note the information not relevant to this thesis, like whether the donation was for a primary, or general election.

2.2.2 Challenges to Disambiguation

Disambiguation of the FEC dataset is not an easy task. Without the presence of a unique identifier in the dataset, we are forced to rely on information from name, address, occupation, and employer fields. These fields each have their own nuanced problems, but overall certain recurring problems are common to all fields.

Data is not checked for accuracy at the time it is filled out by a contributor. This leads to many challenges. Spelling mistakes can occur if a donor misspells any of the fields, and fields are occasionally left blank. Additionally, the way different contributors report the same value for a field may differ, and an individual may even change the way they report a field over time. In the worst case, it is possible that a reported field is completely fictitious. Errors may also occur when transcribing written records into digital formats. OCR software may misread a character, and data entry clerks may enter a field with a typo.

2.2.3 Fields

The FEC provides many fields describing each transaction. Of those, the fields which are most informative about the contributor's identity are listed here, along with a description of challenges which were often found in the data.

Last Name. Besides missing values and spelling errors, names often featured varying orderings of their constituent parts (first, last, and middle names). Occasionally first and last name fields were clearly in the wrong order.

First Name. The first name field sometimes also contained middle name information. In most cases where two names were present, either the first name or middle name would be presented as an initial. For example, variations of a name might appear as *John E. Hoover*, *J. Edgar Hoover*, or

Table 2–3: Examples of difficult field transitions [1 of 3]

An example of a missing employer field. Also, note the various ways of reporting the same occupation.

Last Name	First Name	Occupation	Employer	Mailing Address
Kidd	Robert	Orthodontics	Self	850 S State St
Kidd	Robert	Dentist		850 S State St
Kidd	Robert	Orthondontist	Self Employed	850 S. State St

An example showing two missing fields.

Last Name	First Name	Occupation	Employer	Mailing Address
Pappas	Nicholas			606 Swallow Hollow Rd
Pappas	Nicholas	CEO	Biotraces Inc	606 Swallow Hollow Road

An individual who either was promoted to CEO, or alternated between titles for the same occupation.

Last Name	First Name	Occupation	Employer	Mailing Address
Purzycki	Joseph	Chief Operating Officer	Barclays Group US Inc	125 S. West Street
Purzycki	Joseph	Managing Director	Barclays Group US Inc	125 S. West Street

Table 2–4: Examples of difficult field transitions [2 of 3]

The use of the middle name initial is inconsistent.

Last Name	First Name	Occupation	Employer	Mailing Address
Purzycki	Steven	President and Chief Executive Officer	Nanticoke Memorial Hospital	801 Middleford Road
Purzycki	Steven A.	President and Chief Executive Officer	Nanticoke Memorial Hospital	801 Middleford Road

Here we see inconsistent use of the nickname, as well as variations on employer, and street type.

Last Name	First Name	Occupation	Employer	Mailing Address
Stoneberger	Jeffrey	CEO	Shure Line	120 Cazier Drive
Stoneberger	Jeff	CEO	Shureline	120 Cazier Court
Stoneberger	Jeffrey	CEO	Self Employed	120 Cazier Drive

Table 2–5: Examples of difficult field transitions [3 of 3]

Employer and occupation are found in the wrong fields on the fourth entry. Additionally, we see inconsistent use of the nickname and one of the fields was left blank.

Last Name	First Name	Occupation	Employer	Mailing Address
Dalonzo	William	President & CEO	Friess Assoc	PO Box 4127
Dalonzo	Bill	President & CEO	Friess Associates	P.O. Box 4127
Dalonzo	Bill		Friess Associates LLC	PO Box 4127
Dalonzo	William	Friess Associates	President & CEO	P.O. Box 4127
Dalonzo	William	President	Friess Association	P.O. Box 4127

The error in the spelling of *Volkswagen* was likely due to an OCR transcription error. By consulting online sources, we also see that the individual has indicated the address of the car dealership in one of the records.

Last Name	First Name	Occupation	Employer	Mailing Address
Carlson	Arthur	President	Dover Volkswagen Inc	1387 N Dupont Hwy
Carlson	Arthur	Owner	Dover Volkswagon	57 Teal Lane

John Hoover. Use of middle initials was inconsistent, even for the same contributor. It was rare for both first and middle names to be written out in full. However, when two names were written out in full, it was sometimes unclear if both name components should be considered as one. For example *Marry Anne* could be a first name *Marry*, middle name *Anne*, or a single first name *Marry-Anne*. Nicknames were also used, and individuals switched between using various versions of their given names. For example *Bill* and *William* may be used interchangeably.

Mailing Address. Overall, address information was well-formatted. Entries in this field usually consisted of a mailing address parseable according to USPS guidelines [47], or a post-office box number. Occasionally this field was left blank.

ZIP code. The USPS has divided the US into geographic boundaries and assigned each a unique ZIP code. ZIP codes can either be five digits long, or nine digits long. A nine digit ZIP code is known as a ZIP+4 code, the first five digits indicate the ZIP code and the trailing four represent a subdivision within that ZIP code. The vast majority of ZIP code information was consistent and well-formatted.

State. This is the contributor's self-declared state of residence. These were usually consistent with the ZIP code's state, indicating a correct value.

City. This is the contributor's declared city of residence. We did not use this field, relying on ZIP code and mailing address as a more refined measure of location instead.

Occupation and Employer. Individuals would often change the way they declared these fields, switching between synonyms for the same position, such as *Attorney* and *Lawyer*. Promotions and retirements also occurred. Occasionally these fields were represented as acronyms.

Other Fields. While other fields such as the party to which the donation was made, or the amount given, may also be informative, we must ignore these fields. Political scientists often will look for patterns in party allegiance (referred to as party ID in the literature), as well as spending amounts or other data based on these fields. If we used those fields to disambiguate the data, we would be introducing inherent bias in any research based on our disambiguated dataset.

2.3 Data Matching

Data Matching refers to the task of linking up matching records from different databases together. Generally the entities being linked represent people, and the datasets provide varying identifying characteristics of these individuals, such as name, date of birth, and address. Data matching techniques can be applied to most fields where records need to be matched, but the particular details of the implementation will be domain-specific [38]. Applications of data matching can be found in diverse fields, such as statistical analysis of census data and health care [18, 23, 22, 51].

Data matching techniques developed concurrently within different communities. Statisticians developed a concept called *probabilistic record linkage*. In 1969, Fellegi proposed a computer assisted methodology for record-linking using comparisons between fields of records and showed that statistical models would hold if the fields being compared were conditionally independent from each other [15]. Meanwhile, computer scientists were developing their own data matching methods, motivated by the search for ways to maintain large databases. Initially focusing primarily on using string matching, in the past decade the increasing prevalence of big data and data mining lead the community to consider new methods. Recent data matching algorithms make

use of a suite of tools, including machine learning [38], graph-theory [16], and natural language processing [10].

In *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Peter Christian identifies several stages common to most data matching problems [10]. The stages that are most relevant to this thesis are *data-preprocessing*, *indexing*, *record comparison*, and *evaluation*, as well as a pre-data-matching phase *information extraction*.

2.3.1 Information extraction

Christian defines the information extraction phase as a stage prior to data matching. This is the stage when data is collected and saved in a structured database.

2.3.2 Data-preprocessing

Real-world datasets contain a wide variety of data which makes it difficult to process. Problems include spelling errors, missing values, or values in the wrong field, amongst many others. These problems are generated in the information extraction phase, either at time of creation or during transcription between various formats such as during OCR scanning. In the data-preprocessing phase the database is cleaned up as much as possible while preserving the integrity of the data. This phase's aim is to prepare the entries for easy comparison.

Inconsistencies can come from many sources. Different people will change the way they report the same field over time, for example switching between *Bill* and *Billy* [51]. Inconsistencies can also occur when comparing two datasets with different representations for the same field, for example dates may have different formats. To deal with this, preprocessing should attempt to normalize and standardize variations of a value to a single common value [51, 23]. Normalizing is the process of bringing data to within the same scale,

for instance making sure that distances are all recorded in the same measuring unit. Standardizing refers to finding some other representation for values. Standardization should result in a categorization of all the values of a field. An example of standardizing might be to replace occupation strings with their government-defined SIC codes [41, 21].

Fields may present data that is unusable, or in the worst-case, a field may be entirely empty. In these cases the field's value could be replaced by an appropriate place holder, possibly null, or the entry could be removed entirely[10].

2.3.3 Indexing

Comparing every pair of entries in databases is unfeasible in large datasets. Recognized as a major concern from the beginning of record-linking research, methods have been found to deal with these large datasets [15, 22, 38]. In a sparse dataset most comparisons between entities will not result in a match. In fact in most cases the entries will be very different with no overlap of their fields. During indexing, also known as blocking, data is split into subsets of the original data. The comparison phase will then compare within these blocks, but not between them. When deduplicating, this will cut down the number of pairwise comparisons from $O(n(n-1)/2)$ to something much smaller. The computational complexity within each set will still be $O(m(m-1)/2)$, where here m is the number of entities to compare within each block. The summed total time spent on each of these smaller blocks will be less than that of the entire dataset.

Selecting how to setup these blocks is very important. The field on which the blocks are created must be of high quality with few blank values. Ideally, the block splitting algorithm will create supersets of the matching algorithm's output, making sure all matching entries will be in the same block. Some

matches will not be found if the block splitting method does not result in super-sets of the matching algorithm's sets. Jaro proposed a multiple pass blocking algorithm that relies on mutually exclusive fields that could pick up these missed matches [22]. For example, we could separate a dataset of individuals by name and ZIP code. In the dataset split by name we could find the set of people who moved. In the dataset split by ZIP code we could find those who misspelled their names. Merging the results of both sets would give us a list of people who moved and misspelled their name. Remaining missed matches can be found using human inspection.

Other methods to reduce the computational complexity of comparisons have been proposed. These include, placing records with similar field values next to each other, and sampling entries based on values in their fields [38].

2.3.4 Comparison

Once data is segmented into blocks, data-matching will do a pairwise comparison between entries. Each field will need to be compared, and the type of comparison used depends on the field type. Here we outline the types of comparisons related to this thesis.

String Comparison

In 1966, Levenshtein proposed a string comparison method that counts the number of character changes that differentiate two words [29]. Since then, Levenshtein distance has become a popular tool for string comparison in natural language processing [40, 42]. The Levenshtein distance measures the minimum number of character insertions, deletions, and reversals needed to transform one string into another. This number is referred to as the *edit distance*.

Modifications of the Levenshtein algorithm can consider common properties of typos or OCR mistakes by assigning different costs to the transformations based on the pairs of letters involved [32].

Geographical-Distance Comparison

Locations can be represented in several ways, for example by their center alone, or by a bounding-box or bounding-sphere. Assuming that data has been standardized to a geo-location, with a latitude and longitude, a similarity between locations could be measured by the spherical distance between entries.

More complex data

Data can often present itself in complex forms. Each complex data type will require its own domain-specific interpretation. When dealing with complex data, understanding the relations between the constituent parts is important. In the case of FEC data, the mailing address field is a complex data type. Addresses are usually well formatted, consisting of components like street number, and street name. These components generally appear in the same order, and each may or may not be optional. Comparing addresses requires comparing the individual components, while keeping in mind the meaning of each component. Simply doing a string comparison would miss the important nuances of each component's meaning. For example, *123 West Main Street Apartment #4* and *123 Main St* would have a large Levenshtein distance, while representing the same entities.

2.3.5 Evaluation

It is important to assess the quality of the matches produced by the matching system. On smaller datasets clerical review is possible [22], identifying and fixing any mistakes. With larger datasets a more systematic approach is needed to understand the quality of results.

Ideally resulting matches will be compared against ground truth data, also known as gold-standard data. Due to the real-world nature of the data, gold-standard data is rarely available. A ground truth dataset can be created by trained domain-experts manually looking for matches in a subset of the data. Their techniques must be sophisticated enough to represent all difficulties of matches, and should attempt to represent the same diversity of record characteristics as found in the full dataset. Manual data-matching may not be 100% accurate, introducing potential for errors in the gold-standard data. In some domains, access to a commonly used set of high-reliability test data may be possible. It is also possible to generate synthetic test data, but ensuring that the synthetic data has the same characteristics as the dataset is difficult.

Once a form of ground-truth is found, it can be compared to the results obtained by the data matching system. True and false positives, and true and false negatives are used to calculate quality metrics.

2.3.6 Prior Work

Jaro applied record-linking techniques to find matching individuals between the 1985 Census and an independent post-enumeration study. Jaro used many of the fields available in the census data including name, sex, date of birth, race, and address. Their automated matching system correctly matched 96.89% of records [22]. Churches et al. looked at ways to compare field values using hidden Markov models to handle complex data structures like address. The quality of their mechanism's results depended on the type of data that it was processing. It performed slightly better than rule-based approaches when considering complex data-types like address, but worse on simple data like names [11].

Considering these results, we were encouraged that disambiguation of the FEC data using strict rule-based approaches would be possible with the given fields.

2.4 Crowdsourcing

The term crowdsourcing has an expansive meaning, describing many projects that take advantage of the cumulative efforts of many people to perform a single task. Crowdsourcing can be categorized into domains with their own terminologies, like crowdfunding and crowdvoting. In this thesis, we use crowdsourcing to solve computationally difficult tasks.

Crowdsourcing for the purpose of solving computational problems relies on two properties. First, humans are better at certain tasks than computers. Notable examples include object recognition in images [54], or the sequence alignment problem in computational biology [26]. Second, the "wisdom of the crowd" has been shown to out-perform experts in certain situations [31]. Surowiecki describes crowd wisdom as the aggregate response found by averaging individual, isolated judgments [44]. Oinas-Kukkonen places emphasis on the experimental setup required to maximize the wisdom of the crowd. Amongst their recommendations is ensuring the right information is available to the right people, and the presence of a diverse, independent, and decentralized subject pool [33]. Crowd wisdom may break down if the independence requirement is not met, as people change their answers in reaction to social influences from the rest of the group[30, 37, 33].

A large portion of the research community has enthusiastically picked up crowsourcing as a part of their recruitment strategy. Researchers have long relied on lab-sourced participants to perform experiments and repetitive tasks. This would come at high costs, both in time invested finding and training

workers, and the financial cost when having to compensate workers at a competitive wage. These restrictions made it difficult to scale up an experiment. Online crowdsourcing can greatly simplify participant recruitment.

Examples of successful research using crowdsourcing include reCaptchas, which while verifying that an Internet user is not a robot, helps transcribe books [49], Duolingo, which helps translate the Internet while users learn a new language [17], and ESP, a game which helps add meta-data to images [48].

A big question is: why do humans contribute to crowdsourced tasks? Much research has focused on understanding the incentives which bring people to contribute to crowdsourcing initiatives. Kaufmann et al. provide a summary of many of these incentives [25]. They separate incentives into two major categories, intrinsic and extrinsic motivators. Intrinsic motivators are those which arise from within the individual alone, for example finding a task enjoyable, or feeling like their efforts are contributing to the progression of science. Extrinsic motivators can be categorized into immediate payoffs like financial compensation, delayed payoffs like learning new skills, or social motivations like making new friends during a task.

Today it is common practice in research to compensate study participants for their time. Several websites have facilitated the application of financial incentives to online crowdsourcing. These websites must provide an easy to use interface, and must be trusted by both those setting up the tasks, and those performing them. *Amazon Mechanical Turk* is a popular example often used in the research community. Other websites include, *crowdSPRING*, *oDesk*, and *ClowdCrowd*.

It is clear from all these examples, that humans are willing to donate resources, whether they be time, money, or something else. It is also clear

from the abundance of research making use of crowdsourcing, that it is a highly useful tool which is applicable across a wide variety of domains. For the reasons stated above, we decided to use crowdsourcing as an important tool in the comparison of records that were difficult to parse using computer algorithms alone.

2.4.1 Amazon Mechanical Turk

Amazon Mechanical Turk (AMT) is an online crowdsourcing platform which creates the possibility for an "on-demand, scalable, workforce" [2]. AMT provides a framework where "workers" (those completing the tasks) and "requesters" (those setting up tasks for completion), can find each other. Requesters provide monetary rewards to their workers upon successful completion of tasks. Tasks tend to be simple, taking a few minutes to complete.

Tasks are placed on the website by requesters seeking to complete a task. Tasks are referred to by the term Human Intelligence Tasks (HITs). Workers can search through all HITs submitted by requesters and select which ones are of interest to them. Workers may view a preview of a HIT before accepting it. Once a worker finds a task they wish to complete, they must follow through on the task to completion in order for their results to be registered and to be eligible for compensation. Once workers complete a HIT, the requesters are given the option to review the answers and approve or reject the work done. The worker is only paid if the requester approves their work, and a percentage of the payout is paid by the requester to AMT in fees.

AMT's incentive structure is largely based on financial compensation. In a survey, it was found that financial reward per time worked was a top concern for many workers [9]. The incentive to complete high-quality work is encouraged by the requester's ability to reject tasks which are not considered correct. Additionally, AMT tracks a user's approval rate, and will give out a

Master level certification once a user has successfully completed enough HITs. Requesters can filter their workers based on these measures to ensure that access to workers who meet a high quality. Workers who attain a *Master* level will expect a higher pay-out to compensate for the higher quality work that they produce [3].

The motivation to maintain a high approval rate ensures that workers produce a quality output. At the same time, the quality of requesters is also important to maintaining an effective crowdsourcing system. Workers will keep track of requesters that they have had good or bad experiences with and may even discuss the reliability of requesters on forums like TurkNation [9]. Workers will return frequently to requesters that have a good reputation. A requester can maintain a good reputation amongst its workers by paying the workers a good wage per HIT, accepting worker output quickly, and responding quickly to e-mails.

Although the AMT infrastructure can support a wide array of task types, certain tasks, like image labeling, repeatedly come up. Some workers may prefer performing one type of task over others, based on their skills and personal tastes. For example, a worker who is skilled at typing fast may enjoy audio transcription HITs since they will be able to achieve a high effective hourly wage. Once a worker finds a task that they enjoy, they may keep seeking out tasks which resemble it. Workers who have repeated a task many times may be primed to respond in a certain way, potentially a concern in some studies[9].

AMT does not allow users to repeat the same task, and researchers have found that the incidence of individuals attempting to go around this restriction is low [9]. However, it is possible that users may discuss a task on online

forums. This could impact the quality of studies which require naïve participants [9], and break the independence requirement for effective crowdsourcing as described by Oinas-Kukkonen and others [30, 37, 33].

Having a large crowd available to perform work is very important when attempting to do crowdsourcing. One of the things that makes AMT very attractive is its popularity. AMT claims to provide over 500,000 workers from countries all across the globe. This makes finding workers easy for requesters, giving them access to a 24/7 workforce. These workers have hundreds of thousands of tasks constantly available to choose from [2].

AMT has been shown to be an effective tool in a wide range of studies covering topics like decision-making, linguistics, and survey taking [12]. Recently, Crump et al. reviewed the applicability of AMT to psychological experimental research [12]. Psychological experiments call for very strict adherence to protocols, requiring complex instructions, millisecond accuracy, and sustained attention over long periods of time. It is very encouraging that Crump et at. found AMT results to look "like a publication-quality replication study" and "[recommended] that reviewers and editors should consider accepting behavioral experiments done on AMT as a valid methodology" [12].

Research has long had to deal with several problems that are alleviated by AMT. First, finding participants who represent the general population can take weeks or even months. AMT makes recruitment very fast, some researchers have gone as far as to call the speed of recruitment "revolutionizing" [12]. Although the demographics of these participants is not quite the same as the general population, Saunders et al. found that the population of workers on AMT "resembled the population of the United States in several key demographic factors" and concluded that the demographic distribution was

acceptable for analysis of the general population [39]. Similar results have been found by others [35, 36].

AMT could also help improve transparency of research. Replication of studies is rarely carried out. By using AMT, studies could be easily replicated, as long as researchers make their scripts publicly available [12].

2.4.2 Prior Work

AMT has been shown to be useful for entity resolution tasks. Wang et al. created a human-machine entity resolution system which allowed classification of products based on the product name [50]. Their system took advantage of computer similarity measures to find easy matches, and left harder cases to AMT workers to process. They found that workers were able to complete matchings of many records [50]. Gokhale et al. proposed a "hands-off" approach to crowdsourcing entity matching problems containing many fields. Their system uses AMT responses to train a machine learning algorithm. Workers responses were selected using 2+1 majority vote, which considers the response of two workers if they agree, or uses a third response and finds the majority vote. They found that machine learning algorithms could learn correct coding behaviour from AMT worker results [19].

All the evidence surrounding the effectiveness of AMT as a crowdsourcing tool reassured us that it could be a useful tool for our work.

CHAPTER 3
Methods

The process we implemented is inspired by the standard data matching work-flow described by Christen [10]. We start with a data-preprocessing phase which cleans the data. Afterwards, we carry out a process similar to blocking. Our blocks are actually created over several iterations, each iteration using different matching criteria to create sub-blocks. We use match comparison algorithms to decide how to split each block. Eventually the blocks will contain records that can be given unique user IDs. Some sets will not be easily processed by computer alone, needing human curation. In these cases, we first send sets of contributions to a crowd operating on AMT. The crowd will indicate which sets represent the same individual. For the entries which the crowd is uncertain about, we give the pairs to an expert for matching. Our process guarantees high certainty matches by maintaining a very strict matching criteria. The matches that are missed by our system represent difficult cases, dealing with these cases will require further work not carried out in this thesis.

We performed a disambiguation on Delaware because it is the sixth least populous state in the US. This meant that hand-curated evaluation of data was more realistic. We also make references to certain key statistics from one of the most populous states, New York, to show that Delaware's data profile is not an outlier.

3.1 What Constitutes A Match?

This research is predicated on the assumption that it is possible to match records based on the fields which the FEC provides. To convince the reader

that this is possible, we now describe how these fields can be combined together to give indications of matches, or non-matches in the data. To remind the reader, the fields we have to work with are: name, address, occupation, and employer.

Fundamentally, when looking at two records we have to ask two questions. First, do the two records seem similar enough to indicate that the same person created them, and second, is it possible that someone else produced the same record. With an eye to these questions, this section presents considerations needed for an ideal matching system. We also briefly present some of the limitations that made many of these matches impossible to find using either traditional computation, or crowdsourcing. More details about our system are provided later.

3.1.1 Question 1: Field Transitions

We start by discussing the possible changes that fields might undergo over decades. Assuming a lack of typos and other clerical concerns, we identify two root causes that could explain changes to fields. First, a person may make real-world changes in their life, such as move or change employers, which cause the entity represented by each field to change. Secondly, a person may change the way they report the same real-world entity, for example, varying between common pseudonyms for a corporation.

Combining matching information for all the fields will be necessary to deciding if two records match. Deciding which fields need to match, and which ones are allowed to change, is not a well defined problem and requires intuition gained from real-world exposure to the common use of these fields. Can an individual go from being a *Lawyer* in a law firm to a *CEO* of a mining company? The answers to these questions are not well defined. Hence, our automated comparison system only considered the least ambiguous of

transitions. We detected matches only when records shared the same name, and either the same addresses or same employer and occupation.

To explain the nuances, we provide a description of the changes that may occur on each field. We provide suggestions on how a perfect matching algorithm might capture all of these transitions, and then discuss limitations which we encountered. Many of the limitations expressed here might be overcome by a crowd of human workers, or an expert coder who has access to online third-party data.

First Name. An individual may alternate between various nicknames or spellings of their name. The preference for a certain variation on their name may change with time or may be context specific, for instance, using a nickname in familiar environments and their given name at formal events. Conventions exist concerning the matching between nicknames and their formal versions. Using these well-known conventions, a matching algorithm could easily find all the variations of a name. Detecting spelling variations on names is a related problem. Again, many spelling variations are well defined, for example, "John" and "Jon", and can be processed in the same way as nicknames. We used an online dataset of nickname equivalence classes to find the possible matchings on first name variations.

Last Name. A person's last name will usually stay the same. The most common cause for changing a last name would be marriage. It is convention for the bride to take on the groom's last name, so this transition will mostly affect females. Some women decide to carry their maiden names as a component of their last name after marriage. A notable example is *Hillary Diane Rodham Clinton*. If the maiden name remains part of a person's last name, a matching algorithm could potentially uncover marriage transitions using string parsing alone, otherwise, consulting third-party sources like marriage records might be

necessary. We did not attempt to detect either case, focusing only on matching when spelling was the exact same.

Address. A person may change the way they report the same address. Usually, this will result in small changes, like forgetting to indicate an apartment number, or interchanging words like "street" and "avenue". Mailing addresses have many components making it possible to recover mistakes by considering the other components. We implemented a comparison system which allows some flexibility by allowing missing values. A more difficult transition to detect is when a person moves. In this case, addresses will be completely different and finding the transition between the addresses will be nearly impossible without a third-party dataset. Such a dataset exists, the *National Change of Address* database, maintained by the USPS. Unfortunately, research uses are not allowed with this dataset. We did not consider moves when comparing addresses.

ZIP code. Usually the ZIP code of a person's residence will only change when the person moves outside the ZIP code. As discussed with addresses, move information was not available. Another cause for ZIP code changes is when the USPS reassigns or splits-up specific ZIP codes. Theoretically a comparison algorithm could maintain a list of these transition, but building this is difficult. Instead, our system considered that the geo-location of ZIP codes would remain the same after a split. Hence, distance was used as a metric for ZIP code similarity.

Employer. People change employers on a frequent basis. On average, 2.6% of the US population change employers each month [5], and the time spent working with an employer can vary greatly. Transitions between employers will usually remain in the same domain, making transitions detectable if company domains are known. Unfortunately, datasets listing the domain of

companies were noisy and difficult to work with, so we did not consider these. Another consideration is if a company merges with an individual's employer; future records will list the new parent company as their employer. Finding changes in ownership and titles of companies is difficult, but can sometimes be done by a human using web searches. Finally, a person may change the way they report their employer, sometimes using acronyms or variations on naming conventions, for example "Microsoft" and "Microsoft Inc".

Occupation. Generally individuals will work in the same domain throughout most of their lives, so we expect occupation transitions to occur less frequently than employer transitions [20]. Any change in position may coincide with a change in the occupation field's value. Certain transitions are domain specific. For example, an *Attorney* may become a *Partner* at a law firm. Additionally, people may change the way they report their occupation with time, as certain descriptors come into fashion. Unfortunately, all this domain knowledge is hard to capture using computer-based computation. Our automated comparison only considered exact string matches, leaving the more difficult comparisons to human computation.

3.1.2 Question 2: Doppelgängers

The second question requires understanding the nature and distribution of possible field values in both the general population, and the sub-domain of political contributors. Effectively we ask, what is the chance that two people would generate records with enough overlapping fields to make us mistakenly think that both records match the same person.

There are two considerations at play. First, we consider the frequency of each field's value, and second, the nature of each of those values in relation to the values found in the record's other fields. We now explain this in more

detail, starting with the simple case, and working our way up to more complex situations involving multiple fields.

First, consider a comparison on two values of a single field. In this simple case, the certainty that two records match would be solely related to the distribution of values for the field in the population.

Consider for example, that the last name *Smith* is the most common last name in the US, representing 1.2% of the population [53]. If we were only given last names as a field, we could not be certain that two records with entry *Smith* represented the same person. Our confidence that the two entries were the same would relate inversely to this frequency in the US population. We could only truly be certain that two records were the same if only one person in the country had that last name.

Now add a second field, the first name. *James* is the most common first name representing 3.3% of the US population [7]. Although the number of people in the US named *James* is large, and the number named *Smith* is also large, the number of people named *James Smith* will be orders of magnitude smaller than either field alone. Finally, adding the remaining fields of address, occupation, and employer, will give us even more certainty. Eventually, the combination of fields will be such that the likelihood of two individuals sharing the entire set of fields would be negligibly small.

We only have a few fields, so it is possible to reach a confidence ceiling. If a contributor has a common name, employer, occupation, and address, we cannot guarantee that they are not two different people. Even rare combinations of fields won't be 100% certain. Our automated disambiguation system did not distinguish between the popularity of different fields. However, it is important to keep in mind that the idea of ambiguity is understood in general

society. The human workers may not consider occupations like *Homemaker* as informative enough to form a match.

Also consider that the demographics of the FEC data are different from the general population. Although a person with a unique name may be represented in the Census, that one person could donate many times and produce many records in the FEC data. To expand further on this, certain fields may be more prone to having different distributions in the FEC data than in the general population. For example, people with high-earning jobs are more likely to donate money to campaigns. Hence, occupations like *Lawyer* will be overrepresented in the FEC data when compared to the general population. For each field, choosing whether to consider the US population or the FEC population will depend on our assumptions about whether donating money to campaigns is mutually dependent or independent to the field's value. We assume that name is independent of the FEC domain, and that occupation, employer, and address are dependent.

More complex cases require a deeper understanding of the true nature of the data, often needing consideration of the relationship to other fields. We now explore some of the fields, and the considerations required when making matches. We remind the reader that the computer computation portion of our system only considers the simplest cases, the ideas discussed here represent considerations an ideal algorithm would need to make. They mainly affect the human computation element of our system.

Address. Many street names are common to different cities, so it is possible that two street addresses would represent different locations. Combining the mailing address with the ZIP code ensures that we are dealing with a unique address, since each address in a ZIP code is unique. Now that we have located a unique address, several different situations might occur which

affect the quality of the match. In most cases, people report their residential addresses, meaning the number of people sharing that address will be limited to a single family. However, it is possible that an address is shared by orders of magnitude more people. For example, in New York City it is common for hundreds of people to share the same building address. It is traditional for people to include an apartment number when reporting addresses, but many people did not include this in the FEC data. In this case, we are required to compare on street address, a field shared by all residents of that building, meaning that there is increased risk of finding two different people with the same name at that reported address. Another common case which could lead to addresses being shared by many people would be when individuals listed their employer's address. In NYC, the most common reported address was *235 E 42nd Street*, Pfizer World Headquarters, shared by 2,536 contribution records. Records at this address cover a diverse collection of names, indicating that many different people declared their employer's address when contributing.

Occupation and Employer. With occupation and employer, a similar idea holds. The most common occupation in New York was *Attorney* representing 11.1% of contributions, far overrepresented compared to the 0.4% of attorneys found in the general population [4]. Common occupations had to be considered along with their employer, but even then, a popular occupation combined with a popular employer would be ambiguous. Despite being popular, some occupations were actually quite informative. We must not only consider frequency of two fields, but also their relationship to each other in the real-world. For example *President* was the fifth most common occupation in Delaware, but we must consider that a single company only has one president. Hence, when comparing the *President* occupation with a matching employer, we can be very sure that the two records are a match.

Table 3–1: The top ten reported occupation and employer strings in Delaware.

Rank	Occupation	Freq.	Employer	Freq.
1	Retired	4,040	Retired	2,591
2	Attorney	2,859	[blank]	1,535
3	[blank]	953	Self-Employed	1,139
4	Homemaker	914	Self	1,021
5	President	732	Astrazeneca Pharmaceuticals LP	777
6	Physician	723	N/A	719
7	Owner	525	Self Employed	650
8	Information Requested	445	Not Employed	584
9	Lawyer	355	None	513
10	Consultant	353	Homemaker	474

3.2 Statistics and Data Pre-processing

As mentioned in Chapter 2, the data we have to work with is very messy. To illustrate this, we present some statistics on the types of errors found in each field. To prove that numbers found in Delaware are not anomalies, we present statistics from New York as well. We also present the procedures used to clean-up the dataset before further processing.

State. We found that 2.1% of the records in the Delaware dataset actually had ZIP codes located outside of the state. In NY we found a similar percentage, at 2.5%. By cross-referencing the declared street addresses and ZIP codes with online sources, we were able to see that the declared street addresses would nearly always match the declared ZIP code. The fact that both the address and ZIP codes matched gave us very strong evidence that the entries represented individuals outside the state. Although we only looked at

Table 3–2: The top ten reported occupation and employer strings in New York.

Rank	Occupation	Freq.	Employer	Freq.
1	Attorney	36,232	[blank]	20,212
2	Retired	25,449	Self-Employed	15,947
3	[blank]	13,840	Self Employed	15317
4	Partner	8,392	Self	13,633
5	Homemaker	7,201	Retired	11,917
6	Lawyer	6,877	Not Employed	9,987
7	Executive	6,610	N/A	7,659
8	Information Requested	6,554	None	5,380
9	President	6,283	Goldman Sachs	3,565
10	Managing Director	5,996	Pfizer Inc	2,944

a subset of all cases where this occurred, the consistency of the pattern gave us comfort that it could be extrapolated. Hence, we assumed that all cases where the ZIP codes were not in the state were actually referencing a contributor who lived outside of the state, and should not be included in our analysis. We removed these records from our disambiguation of the state. This assumption was further supported by the fact that the average Levenshtein distance between these outlier ZIP codes and the nearest code in Delaware, according to edit distance, was 2.9. Such a large edit distance makes it unlikely that the inconsistency is due to a typo, and points to something else going wrong in the dataset.

ZIP. Beyond indicating an incorrect state, ZIP codes themselves required cleaning. The USPS defines how ZIP codes can be formatted. ZIP codes can be represented either as five-digit codes, or nine-digit codes known

as ZIP+4 codes. ZIP+4 codes are subdivisions of ZIP codes, meaning that the last four digits can be truncated to give the correct five-digit ZIP code. We normalized ZIP+4 codes into their five-digit versions by slicing off the last four digits. Although we would get more certainty about a person's location from their ZIP+4 code, not all records reported nine-digit codes. By ensuring that all ZIP codes were at the same level of precision, we simplified the comparison phase. A handful of ZIP codes were simply poorly formatted, with neither five, nor nine digits. ZIP codes smaller than *10000* are meant to contain leading zeros, meaning that any entries with fewer than five digits were poorly formatted and were removed. Entries with greater than five digits were truncated to their leading five digits. Manual inspection indicated that this was the appropriate action in most cases since the leading five digits usually represented ZIP codes consistent with the street address.

Name. Some records had either missing first or last names. This occurred in 2.5% of records in Delaware and 3.0% of records in New York State. Although this leaves some information that could be used for disambiguation, such as mailing address and ZIP, disambiguating these records would require a very different process than the one we explore in this thesis. We removed entries that had missing names.

Address. Addresses were generally well formatted and could be processed by an address parsing library. Some street addresses were missing. Although street address is useful information, we can still find matches without it by looking for identical name, occupation, and employer. Hence, we do not filter out fields with missing addresses.

Occupation and Employer. As with addresses, we do not remove entries with missing occupations or employers. We assume that the remaining fields should be informative enough to find matches. Values which, while

not empty, still conveyed a lack of information, were often found in these fields. Examples include "INFO REQUESTED", and "N/A". These should be considered equivalent to empty fields. Although we did not perform cleaning on these fields, we did generate histograms showing the frequency of the various employers and occupations. We later used this histogram to gain a better understanding of the likelihood that more than one person might share the same field values.

General. Use of punctuation marks was inconsistent. We found that dashes were consistently exchanged with a space character and periods were replaced by nothing. To simplify string comparison, these switches were made. For example "forty-second" was changed to "forty second".

The presence of missing fields, particularly name, raises concerns about the quality of the data collection process. It seems that poorly formatted fields were not mutually independent; records with missing fields often contained more than one missing field, or errors in those other fields. This meant that when filtering poorly formatted entries, we had the beneficial side effect of removing other poorly formatted fields. One of those effects was the elimination of most of the entries with no address. In Delaware, the number of records with a missing street address went down from 462 to 19.

3.3 Set Splitting

Having cleaned our data, we now attempt to find records which match an individual. Our goal is to find matches for individuals who share the same name, and same address or job. To do this we could simply do a pairwise comparison over the entire dataset. However, this would result in an unreasonably large number of comparisons in a large dataset. It is especially unrealistic if we increase the complexity of the comparison algorithm.

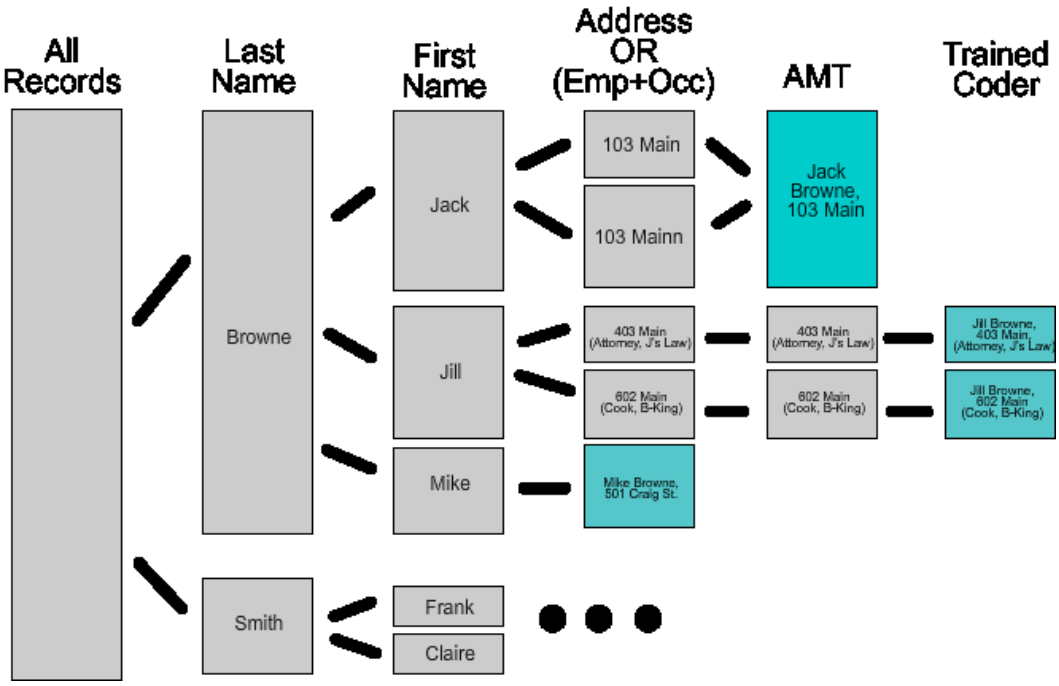

Figure 3–1: The general work-flow for our system. We show the breakdown of sets along the various matchings. IDs are created when a set has been shown to have no remaining ambiguity. Sets with assigned IDs are shown in teal.

To solve this problem, we designed a system which resembles the data-matching concept known as blocking [10]. We split the data into progressively smaller sets, reducing the size of the comparison space on each round of splits. We took advantage of the smaller comparison spaces by increasing the complexity of comparisons as sets became smaller.

The set splitting phase performs pairwise comparisons between particular fields of entries, determining which ones are the same. It places records which match into the same set. Similar entries are placed into more refined sets with each iteration. At some point, sets are precise enough that we can determine with high certainty that the records within them are those that belong to an individual. Sets which are not precise enough to be assigned using a computer, are given to a crowd of human workers to sort out, and finally those that the crowd are unable to match are given to a trained expert. The final result is a

database with a unique contributor-ID given to all records matching the same individual, or a marker indicating that sets of records might belong to the same person, but we are unable to be certain.

We now describe this process in more detail. To simplify comprehension, we define the sets generated at each phase as S1, S2 and S3 sets. The process of how each type of set is found is now described in more detail.

3.3.1 S1 Sets

We start by splitting the entire dataset up according to last name. Except in the cases of marriages and spelling errors, an individual's last name should not change. Our system only considers consistently-spelled last names, and doesn't allow for last name changes like those which occur after a marriage. This means that all records associated to one individual should have the same last name. Last name is also the simplest data type to compare in our system; assuming no spelling mistakes, records will only match if the last names are spelt exactly the same. To build these sets, we first assigned each distinct last name an ID value. We then marked each record with its appropriate last name ID value. This is a simple linear operation. In fact no processing is needed if we use the last name field as the S1 ID key.

3.3.2 S2 Sets

The second split occurs within each of the S1 sets, and is based on first name. We assume that records produced by the same individual will share the same first name, or its nickname variations. Within each S1 set, we do a pairwise comparison of all the record entries, looking for matching names, or their nickname equivalents. We then connect matching records to each other, resulting in connected components, where each name is connected either directly by one edge, or through a longer path, to other records.

The first name comparison considers exact string matches, as well as matches against an equivalence class of names with their common nicknames and various spellings. This equivalence class was built from a public domain list of common nicknames[1] . If a name has nickname equivalents, the comparison algorithm will look at each variation of the name found in the equivalence class, and attempt to find other records with names matching that string. For example, "William", "Bill", and "Billy" are all within the same equivalence class, and records with these names will get grouped together. If a first name contains more than one component, we needed to distinguish the person's commonly used name from their other name. Since in almost all cases the less-used name is represented by an initial, we consider that the longest name component should be used for comparison. Due to the structure of our connected components, our simple system cannot handle differing name initials, "Jon R." and "Jon B." will both match and link to "Jon", and then be linked together through their mutual connection to "Jon". This never occurred in Delaware, and only happened in 34 out of the over 80,000 S2 sets in New York, showing that most sets do not fall into this category. The few sets that do suffer from this problem could be disambiguated by hand.

Our system is now taking advantage of the reduced computational complexity obtained by operating only within smaller sets. The most common last name in the US only represents 1.2% of the population [53], meaning that we can expect S1 sets to generally contain no more than approximately 1.2% of the FEC dataset, the size of which will depend on the population of the state. Since the S1 set are much smaller than the entire FEC dataset, we can do a pairwise comparison of order $O(N^2)$, and expect reasonable computation

[1] Common Nickname CSV, https://github.com/onyxrev/common_nickname_csv

time. The first-name comparison algorithm used in this thesis is simple, but the savings would quickly add up if we performed more complicated name comparisons, like checking for typos.

At the end of the S2 split phase, we have a collection of S2 sets containing all records with the same first and last name.

3.3.3 S3 Sets

The final split will occur within each of the S2 sets. Now the splitting algorithm splits such that records are placed together if their addresses match, or both employer and occupation match. Since we are splitting inside of S2 sets, the sets produced by the S3 split will contain records that match on name and address, or name, employer and occupation. Thanks to the use of connected components, a set will also contain cases where a person moved but kept the same job, or changed jobs while maintaining the same address. Cases where someone changed both their address, and their occupation or employer will be missed and placed in separate S3 sets. For the purpose of disambiguating the FEC data, we assume that the likelihood of sharing the same name and address is very low. We also assume that sharing the same name, employer, and occupation is very low. These assumptions make it possible to claim that S3 sets will contain only records which are common to the same individual.

Address

To check for sets of entries with matching address, we modified an existing address parsing python library. We removed unnecessary parsing, like checks for ZIP, state, and other fields that were not present in our mailing address fields. We expanded on the different types of apartment number prefixes to handle the diverse types of indicators found in the FEC data, such as *suite*, *ste*, and *pmb*. We also implemented a simple algorithm to compare P.O. Box

addresses using regular expressions. P.O. Box numbers were only matched if the number components were the same.

Our mailing address parser handles addresses with this format:

[Street#][Street Direction][Street Name][Street Type][Apartment#]

Street number is the series of numbers which preceed a street name. Street directions generally are written as cardinal and inter-cardinal directions, such as *north*, *south*, and *northeast*, or their initialed representations. Street type is a generic modifier to the street name, such as *street* or *avenue*, or their abbreviated versions. Apartment numbers were demarcated by an apartment prefix followed by an alphanumeric value, for example *Apt. 7B*. Street number and street name were required fields and had to match. Other fields were optional, and only would signal a non-match if they conflicted. This captures the real-world fact that individuals will often not indicate their street direction, apartment number, or street-type descriptors.

To ensure that the address represented the same location, we made sure that the ZIP codes matched. To handle the reassignment of ZIP codes, we considered ZIP codes to match if they were within 10 kilometers of each other.

Employer and Occupation

For job matching we required exact string matches on both fields. If either field was empty, no match was possible. In the case of employer, we also used an equivalence class of known employer matches that we had generated in earlier phases of this project. The equivalence class consisted of spelling variations and known job transitions. The data originated from work done on Ohio, with many entries representing employers located uniquely in Ohio, so its benefits were limited in Delaware. The dataset could easily be expanded to

cover any state if we can attain high quality data about employer equivalences. Equivalence classes could also be applied to the occupation field if data were available.

Finding naming variations on these fields is difficult for a computer, but easy for humans. For example, *"Daley Erisman & Vanogtrop/Attorney"*, *"Daley Erisman & Van Ogtrop"*, *"The Erisman Law Firm LLC"*, and *"Erisman & Vanogtrop Law Office"*, were all variations found in the FEC dataset which all represent the same entity, but finding a way to compare these using computer algorithms is difficult. Our automated matching algorithm only considers exact string matches, leaving the more complex cases to human computation.

3.4 ID Formation

We are now finally able to start assigning unique ID values to sets of records. After set splitting, we have S3 sets whose records almost certainly all belong to the same individual since they share the same name and address, or the same name, occupation, and employer. Nevertheless, we are not certain that there are no other records that belong to that same person. This may occur if there are spelling mistakes in a name, in which case the sets containing this individual's records may be in different S1 or S2 sets; we ignore this case and leave it for future work. In this thesis we deal with solving the case where the matching algorithm failed to find matches on both job and address. In this case both of the individual's S3 sets will appear in the same S2 set.

Assuming no spelling mistakes in names, several cases exist when assigning unique individual IDs:

Simplest Case. In the simplest case, if an S2 set contains only one S3 subset, we can be highly-confident that the set's records belong to the same individual. This comes from the reasonable assumption that there are no other matching individuals with the same name, and all the records are linked by

common addresses or jobs. In this case we assign all the records the same unique contributor ID number.

Harder Cases. The harder cases are when an S2 set contains multiple S3 subsets. These are situations where the matching algorithm was not able to find matching jobs or addresses between contributions in both S3 subsets. A common cause might be spelling variations in the fields, which cause the comparison algorithm to not link similar records. Our computer algorithms have failed us in these cases, and finding better ones would be difficult, hence we now rely on human computation. We send S2 sets with more than one S3 subset to AMT, where the crowd is asked to determine which S3 sets should be merged to form a single set. If the majority of the crowd is unsure whether the two S3 sets were generated by the same person the set is given to a trained expert for a second round of inspection. Details on the crowdsourcing aspects of this system are given in the next section of this chapter. In the case that the crowd or the expert believe the sets match, we merge the sets, and assign a unique contributor ID to all the records enclosed. Beyond deciding if S3 sets are the same, the expert can also decide if the sets represent different people, or that there is not enough information to be certain either way. In the case that the expert thinks they are different, we give each S3 set its own unique contributor ID, and assign each of their enclosing contributions the corresponding ID. In the case that the expert feels the data is too ambiguous, we leave the sets unmerged and mark the records with "unsure-link" IDs.

Note that our AMT task only handles the case that an S2 set contains two S3 sets. S2 sets with more than two children will require a different task setup, and need to be considered in future work. Only a small fraction of contributions in Delaware fall into this category, five percent, as seen in Table 4–3.

3.5 AMT

As described in the previous section, S2 sets containing two S3 sets were processed by human computation. We now describe the details of the crowd-sourced portion of this task.

We designed a task that presented a worker with two tables, each table listing the contributions found in one of the S3 pair sets. For each pair of tables, the worker was asked to select between two radio buttons, one indicating confidence the sets matched, and the other indicating reasonable doubt. The same set of tables were presented to nine different workers, and majority vote was used to decide if the sets represented the same individual. When majority vote was unable to find a match between the pair, the pair was marked for post-processing by a trained coder as described in the *Post-Processing* section of this thesis.

As shown by many research teams, the quality of instructions can affect the quality of output [12]. In order to ensure that the general population would interpret the task correctly, we tested the task on a handful of lab-sourced participants. We asked participants to verbally describe their thought-process during the task, and used their input to iteratively improve the instructions.

The instructions start by describing the fields to the worker. We then ask the worker to perform a series of example matches. On a correct selection, a helpful text description informs the worker why the entries should, or should not, be matched. The worker is allowed to make mistakes during this part until their answers are consistent with the expected answer. A summary of the instructions is provided at the top of each work-page to remind workers of key considerations. We did not reveal the purpose of the study, hoping to reduce worker bias.

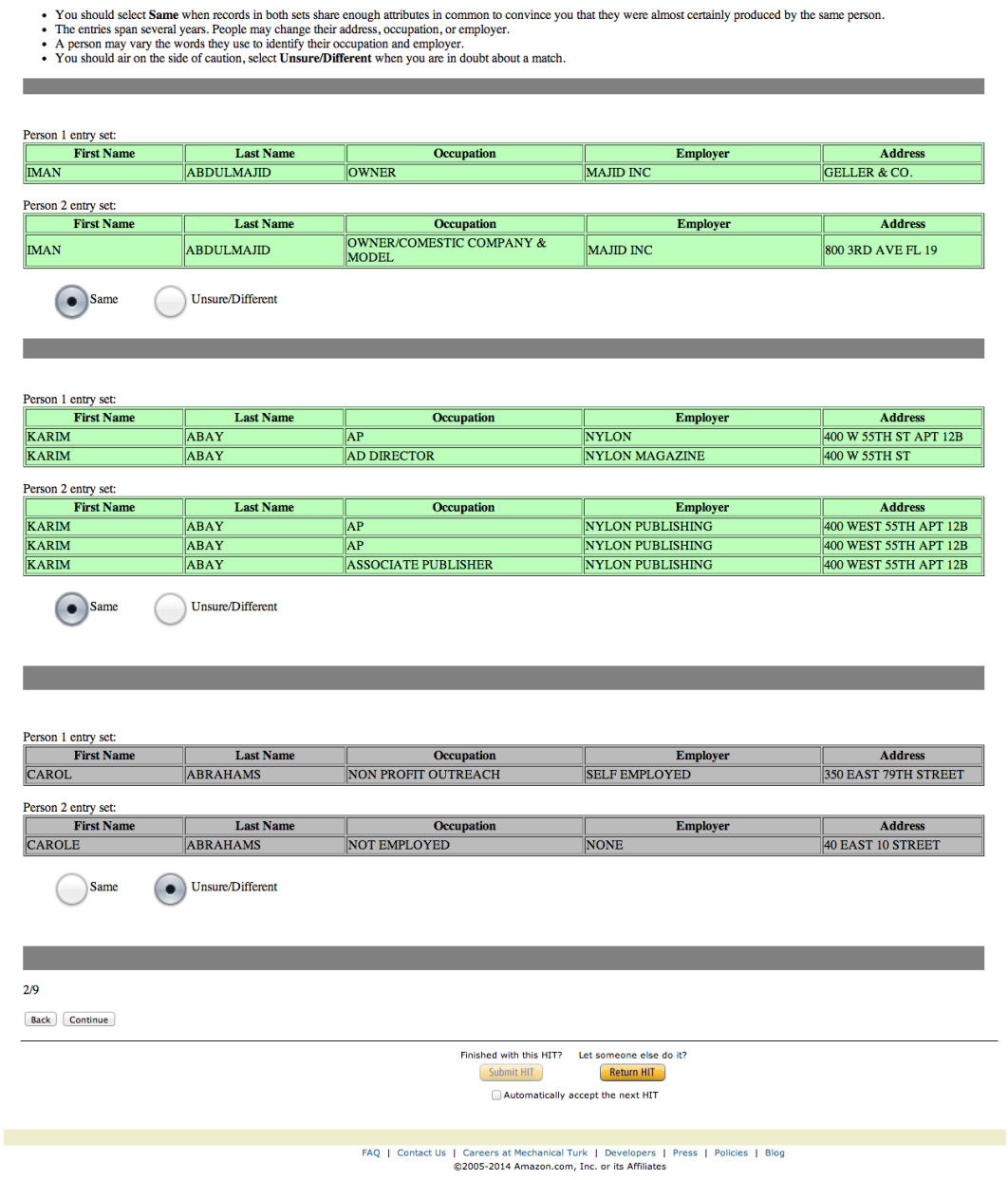

Figure 3–2: A sample of the matchings workers are expected to perform on Amazon Mechanical Turk.

We attempted to make the user interface as simple to use as possible. To simplify comparison between tables, we hid entries with the same exact same field values as existing rows in the table. This kept tables smaller and easier to compare. Radio button sizes were scaled up to make them easier to click, and clicking on them would change the colour of the corresponding tables to indicate that a choice had been made. Each page showed ten questions, giving the worker a sense of progress and not overwhelming them with a long list of matches to perform. Workers were required to answer all questions on a page before proceeding. Workers could go back to a previous page to consult the instructions, or change previous answers.

AMT does not allow a user to repeat the same HIT, and every set of records was only found in one HIT. This means that there is no worry of the same user answering the same questions twice. However, AMT does allow users to perform different HITs for the same batch of HITs. Hopefully users will improve with experience as they start seeing the same patterns of comparisons emerge. We expect these workers to get faster at the task as they gain experience, meaning that as workers improve, they are more likely to return for future HITs. This does introduce concern that some workers will be better trained for the task than first-time workers.

We expect first-time workers to spend longer on the instruction step, increasing their completion time. For a worker with previous experience performing the task, completing a HIT of this task should take between 10 and 15 minutes. Earlier trials had shown that the average completion time varied drastically, but averaged 20 minutes. We only recruited *Master* level workers to ensure a high-quality work force. We aimed for an average effective hourly wage of around $4.00 an hour, a bit low for a *Master* level worker. A worker

who could complete the task quickly could earn about $6.00 an hour, creating an incentive to return to the task and improve.

The decision to limit ourselves to *Master* level workers was motivated by existing research. Studies have found that the accuracy-level of non-master workers was only 91.6% that of *Master* workers[24]. *Master* workers have also been shown to be more likely to return to the same task. Since we expect individuals to improve as they return to our task, this property is desirable[43]. The known negative effects of increased run-time and higher financial costs were not significant in this paper since Delaware was a small state.

3.6 Post-Processing: Expert Curation

Workers on AMT do not have domain knowledge of the FEC dataset. This will influence the matches they make. For instance, they will not understand that *Attorney* is a heavily overrepresented data field. For this reason, we added a second step which uses a trained expert to check the crowd's output. Assuming a reasonable understanding of the task by the our workers, we define the following types of errors which we expect to see often:

Error type 1. The first error type occurs if the crowd is overly generous on mergers with low-information fields. Examples include, finding the value *Homemaker* listed as an occupation or employer, or addresses which are common to many people, like *235E 42nd Street*, the Pfizer World Headquarters. The same type of error will occur from the automated ID making as well.

Error type 2. These errors occur when a merge should have taken place, but the data made it hard to notice. For example, when two companies merge the employer field could change to something un-recognizable by a simple string comparison. Domain knowledge of company mergers would be required, something which we don't expect the workers to have. In these cases the crowd will return "unsure/different".

Looking at both these errors, we create a list of sets which need to be curated by an expert. The expert will need to take extended time on each record, cross-referencing every field using third-party data. Whenever the trained coder finds a merge of all entries in the set, we can consider this a unique ID. Whenever the trained coder decides two entries in a set should be different, we make both of those their own IDs. If the trained coder can't be sure, we can't make a unique ID, so we mark these sets as IDs having potential matches.

3.7 Validation

Ground-truth data is not available for the FEC dataset. Instead, we had to rely on hand-curated data. In order to test the quality of the crowd's work, we generated our own ground-truth data by having our expert perform a complete AMT HIT. We evaluated the quality of worker output by evaluating the level of agreement betweeen the crowd's responses and that of the expert. We also performed visual inspection of some of the final IDs to ensure that they were correct, showing that the entire work-flow results in high quality IDs.

CHAPTER 4
Results

4.1 Set Splitting

We start by evaluating the benefits of the set splitting phase. The main goal of the set splitting phase is to reduce the comparison space to allow for $O(N^2)$ comparisons within the sets. To evaluate success, we looked at the number of sets generated at each phase, and their sizes. We also consider the distribution of set sizes, which give us an idea for average and worst case performance. The results are summarized in Table 4–1.

The general trend is towards an increasing number of sets, each containing fewer entries. This is consistent with expected behavior. The gain with the first set split into S1 sets is the largest and the gain between S1 and S2 sets is much greater than that between the S2 and S3 sets. This shows that the majority of the filtering occurs on name. Often, S2 sets already contain only one individual's contributions. This is consistent with intuitive ideas of name frequency. The chance that two people with the same name donate money in the same state is not particularly high, especially for smaller states like Delaware.

Within each split phase, the distribution of set sizes fell off very quickly. The difference between the 99th and 90th percentile set sizes was substantial. The largest S1 set represented 221 entries, or 0.89% of all contributions, and only seven S1 sets had over 100 entries. The vast majority of sets contained much fewer contributions. This sharp drop off in set sizes means that the average time to compare within a set will be relatively small, leaving only a few sets which take longer to complete. The sum of all these shorter comparisons

Table 4–1: Distribution of record counts in the sets. We see a trend towards decreasing set sizes between each set-splitting pass. By comparing the 99th and 90th percentiles for set sizes, we can also see that there is a steep drop-off in the set sizes within each split phase.

	All Data	**S1 sets**	**S2 sets**	**S3 sets**
Number of sets	24,949	4,339	7,450	8,265
Largest set size	24,949 (100%)	221 (0.89%)	138 (0.55%)	138 (0.55%)
Average # records per set	24,949	5.7	3.3	3.0
# of singletons	0 (0.0%)	1,681 (38.7%)	3,877 (52.0%)	4,617 (55.9%)
99th percentile set size	N/A	64	31	27
90th percentile set size	N/A	12	7	6

will be much smaller than that of a complete pairwise comparison over the entire state.

Particularly, it took 180.1 seconds to execute the disambiguation code on all the S2 sets, and 295.8 seconds to execute the same code on the S1 sets. The estimated runtime to run over the entire dataset, without subdividing into subsets, determined by a partial run of the algorithm, was over 40 hours. Table 4–2 shows the runtime for a selection of S2 sets according to size. These numbers show the clear benefits of splitting the dataset into subsets, an advantage which will become even greater with more populous states.

As for the match between frequencies of fields to their real-world values, we found that last names did not correspond with their distribution in the general population. The largest S1 set was for the last name *Davis*, not *Smith*, and represented 221 contributions, or 0.89% of all contributions in the state. This result can be explained by certain contributors contributing substantially

Table 4–2: The time it took to disambiguate some of the S2 sets. The faster processing time of smaller sets can clearly be seen.

S2 Set Name	**Set Size**	**Disambiguation Time (s)**
Beverly Bove	138	5.70
Thomas Connelly	76	2.34
Lee Bye	53	1.73
Nicholas Caggiano	20	0.395
Robert Clark	7	0.0229
Joseph Dipinto	2	0.00442
Kathy Doty	1	0.00135

more frequently than others. For example, of the 221 contributions from people named *Davis*, 135 ended up being placed together in the same S3 set for *Davis Chester*, Vice President of a pharmaceutical company named AstraZeneca. We don't expect the number of contributions made by the same donor to increase when looking at more populous states. This is because there is a cap on the total amount one person can donate per election cycle. Assuming that reaching the maximum contribution limit is rare, we can expect high-frequency donors to skew the data less in large states, as other contributors with the same name make donations.

4.2 IDs

Having shown that sets could be built efficiently, we now show that these sets gave highly informative information. The set splitting phase allowed us to capture a large portion of the unique identities using traditional computer computation alone.

Returning to our earlier discussion about ID formation, remember that the distribution of S3 sets will govern the computation of IDs. IDs can only be

determined automatically when an S2 set contains only one S3 subset. Note that this definition is different from the definition of set sizes given above, here we are counting subsets, not total number of contributions.

The distribution of S3 sets within S2 sets is shown in Table 4–3.

We see that the majority of S2 sets only contain a single S3 set; 6,777 S2 sets, representing 77.9% of contributions fall into this category. Our automated process has done very well. Our system found 569 sets with two S3 children, representing 17.0% of all contributions. Of these, we sent 540 sets to AMT for processing by the crowd. We were unable to fit all 569 sets into our AMT tasks because of the setup of our task, which only took 90 pairs at a time. Those with three or more children sets accounted for the remaining 5.1% of contributors, and were not considered in this work. It took a total of 23 hours and 24 minutes to run the AMT tasks. The rather long time to run is explained by the fact that we only used *Master* level workers[43].

Of the sets sent to AMT, 56% were found to be "matching" by majority vote. The remaining 234 pairs of which the crowd was "unsure" about, were sent to an expert for manual inspection. The expert shared the crowd's lack of confidence in most cases. Of these 234 unsure cases, only 22 were changed to a match and 4 were identified as belonging to different individuals. It should be noted that the expert coder did not exhaust all possible sources of third-party information. Hand-curation relied mainly on Google and LinkedIn searches. The results could be improved if better third-party data were available. However, what we have shown is that the remaining cases are very hard. We discuss this in more detail in the next section.

4.3 AMT

We now evaluate the quality of the crowd's work by comparing it to a trained expert. We show the crowd helped eliminate a majority of sets to

Table 4–3: The number of S2 subsets with various S3 subset counts. The majority of contributions fall into S2 sets with a single S3 subset, these are handled automatically by computer alone. Another 17.0 % of records are found in S2 sets with two subsets, and are handled by crowdsourcing. The remaining 5.0% of records were not handled by our system.

Subset Count #	**Frequency**	**Contributions Contained**
1	6777	19,442 (77.9%)
2	569	4,236 (17.0%)
3	75	833 (3.3%)
4+	29	438 (1.7%)

be looked at by the expert. We will also present and discuss some of the interesting cases that caused a discrepancy between the crowd and the expert.

We compared 90 AMT responses to that of an expert coder. We were curious about the effect on outcomes as the crowd increased in size. Since we had nine workers perform the task on AMT, we compared each odd-number-sized permutation of those nine workers. It should be noted that this lead to unequal sampling of the different crowd sizes since there are more permutations of, for example, two workers, at 32 permutations, than all nine, where there is just one.

In order to compare answers, we needed to classify the different possible agreements and disagreements between the expert and the crowd. Due to the option of uncertainty, traditional notions of true and false positives and negatives don't apply. Instead, we create a domain-specific notion of correct and incorrect classification. We effectively have four situations:

Case 1. In the first case the crowd decides that the records match the same individual, and the expert agrees. In this case the crowd generated a correct match. This type of match would be represented in the disambiguated

Table 4–4: The frequency of the various possible types of agreement and disagreement between the expert's answer and that of the crowd. We vary the number of workers taken into account for the majority vote. As worker count increased, the level of agreement increased (Same-Same and Unsure-Unsure).

Expert Response	Crowd Response	Worker Count				
		1	3	5	7	9
Same	Same	55.8 %	59.6%	60.5%	61.3%	62.2%
	Unsure	22.0%	18.2%	17.3%	16.5%	15.6%
Unsure	Same	2.5%	1.8%	1.6%	1.4%	1.1%
	Unsure	19.8%	20.4%	20.6%	20.8%	21.1%

database as a correct match generated without the need for approval from an expert.

Case 2. Here the expert has seen a match, but the crowd does not. This is a failure of the crowd, but not a failure of the disambiguation system as a whole, since the error would have been caught by an expert on review.

Case 3. Here both the crowd and the expert agree on the uncertainty of a match, meaning the crowd was successful. In terms of workload on the expert, this situation is similar to the second case since further processing is needed after the crowd looks at the pair. The fact that neither group can detect the match means that our system has reached its maximum capacity to compare the records.

Case 4. The only case of a true failure by the crowd is when the expert is unsure of a match, but the crowd thinks that there is one. In this case the pair of S3 sets will be incorrectly merged, an error that would not be detected by any component of our system.

The frequency of these situations are shown in Table 4–4.

We only had one question result in a type 4 situation when using majority vote of all nine workers. This was a border-line situation, where both records

Table 4–5: Examples of various forms of agreement and disagreement between the expert and crowd responses. [1 of 2]

Expert: Same, Crowd: Unsure. Here a match is found by noticing that the Employer field includes the letters MD, indicating that both records represent physician. Also, the individual's name is found in the company title, while the other record indicates ownership.

Last Name	First Name	Occupation	Employer	Mailing Address
Bolourchi	Habib	Physician	Self Employed	16540 Coastal Highway
Bolourchi	Habib	Owner	Boulourchi MD	4503 Highway One

Expert: Unsure, Crowd: Unsure. The crowd was able to determine that the occupation and employer fields are not informative enough.

Last Name	First Name	Occupation	Employer	Mailing Address
Allingham	Pamela	Homemaker	Homemaker	927 Westover Road Highway
Allingham	Pamela	Housewife	Self-Employed	26 Foxhill Ln

Table 4–6: Examples of various forms of agreement and disagreement between the expert and crowd responses. [2 of 2]

Expert: Same, Crowd: Unsure. Online searches found a news article describing the acquisition of the individual by DLA Piper LLP from the other employer.

Last Name	First Name	Occupation	Employer	Mailing Address
Brown	Stuart	Attorney	DLA Piper LLP (US)	1000 N. West St.
Brown	Stuart	Partner	Edwards Angell Palmer & Dodge	26 Foxhill Ln

Expert: Same, Crowd: Unsure. The individual's name is part of the company title, indicating ownership, and the other record indicates self-employment. This is enough to be quite sure they are the same person.

Last Name	First Name	Occupation	Employer	Mailing Address
Callaway	Paul	Self Employed	Self Employed	326 Carpenter Bridge Road
Callaway	Paul	Owner	Callaway Furniture & Dodge	PO Box 232

indicted *homemaker* as the contributor's job. A post-processing phase which eliminates matches on common field values could catch this and send it to an expert coder for review. The other sets which resulted in type 4 situations with majority vote of fewer workers could generally also be solved by similar post-processing.

As can be seen in the table, as the number of workers increase, the responses tend towards agreement with the expert. The improvement was never particularly large, maxing out at only 6.4 percentage points. The question of whether the improvement is worth the cost of hiring the extra workers depends on the relative cost of finding those 6.4% of merges using AMT versus an expert.

Examples of various causes for agreements and disagreements can be found in Tables 4–5 and 4–6.

4.4 Political Behaviour

Having shown that the quality of data is high, we now perform some analysis of the political behaviour involving campaign contributions. We present two findings that emerge from our data.

Our first finding is that the vast majority of contributors do not contribute large amounts: 3,545 of the 6,889 contributors only donated $200 over the decade. Contrast this with 2,718 contributors who donated over $1,000 and only 540 contributors who donated over $5,000. Delaware's population was 897,936 in 2010 [8], this means that only 0.77% of the population gave $200 or more, a substantial concentration of political participation via contributions.

Unfortunately, our second finding shows that the disparity in financial representation is even larger. We found that the aggregate contribution from small donors is far overshadowed by the contributions of a few big donors. The largest 100 contributors accounted for 30.1% of total financial contributions.

It took the aggregate contributions from the 5,861 smallest donors to match this amount. These findings may have substantial implications if politicians cater more to high-spending individuals.

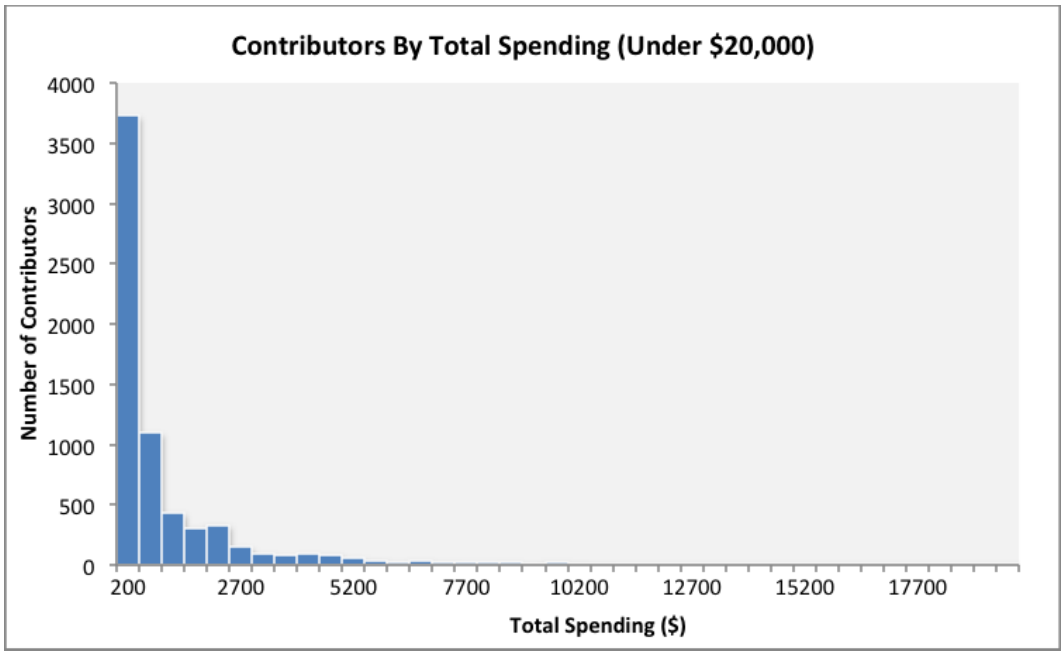

Figure 4–1: Total counts of contribution sizes. Most contributions are small.

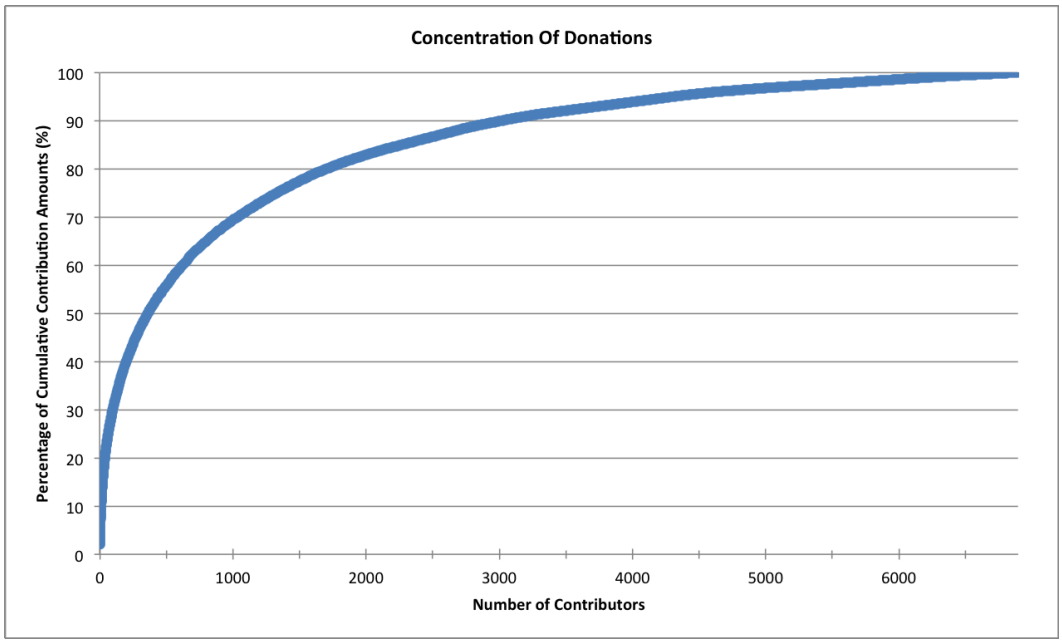

Figure 4–2: The cumulative distribution of contributions. A few individuals donate most of the money.

CHAPTER 5
Discussion and Future Work

We have shown a design for a system that can efficiently disambiguate a substantial portion of the records found in the FEC database by splitting records into well constructed subsets. Our system is based on intuitive notions of what is required for records to match, considering the nature of transitions which may occur between records. By maintaining a very strict definition of what constitutes a match, we can be highly certain that the sets of records which constitute an ID all belong to the same person.

We successfully took advantage of crowdsourcing to help resolve difficult matches, replicating similar successes that have shown Amazon Mechanical Turk as a useful tool for entity matching [50, 19].

Our system is built to be extensible. The comparison algorithms used to split sets up could be improved with more complex versions. For example, our comparison algorithms currently use little third-party information, but could easily be modified to include more information, allowing the system to detect a wider range of field transitions.

We applied our system to the state of Delaware. Although Delaware is a small state, the statistical distribution of fields and set sizes show that our system would be able to scale-up to larger states and maintain an efficient runtime. Our disambiguation of Delaware appears to be reaching the limits of high-certainty matches that are identifiable. We reach this assumption by considering that with each stage of our disambiguation, we found diminishing returns. Our computer algorithm was able to capture the majority of matches, leaving only a small subset to be handled by the crowd, who in turn, sent a

subset to our expert. The fact that our expert was not able to find many new matches indicates that we are approaching the limit for high-certainty matching.

5.1 Limitations and Suggestions for Improvement

Our system for disambiguation within a state is not perfect. There are several situations under which this system would not find all records belonging to the same individual. Some of these are now discussed.

A change in either first or last names would lead to records being placed in different S1 or S2 sets, and hence different S3 sets and IDs. This would mainly happen in the case of spelling errors, or marriages. In the case of spelling errors, loosening the exact string match restrictions would help reduce the problem, use of Levenshtein distance could help. In the case of marriage transitions, access to marriage records would be easiest, but it may also be possible to detect marriages by looking at the temporal flow of records. If a person gets married, their records before the marriage will carry one name, and those after the marriage will carry another, while other fields, mainly occupation and employer, should stay the same.

As for the failed matchings between S3 sets, it is possible that fields underwent difficult-to-identify transitions, like employer mergers, or there just wasn't enough information due to missing values, or low-information values like *Retired* which could apply to many people. Decisions will need to be made about how to handle cases where there is ambiguity. For example, if two people work as lawyers but different at law firms, how to determine if they are the same person. One option is to look at the order of transitions, since a person won't switch back and forth between multiple jobs in random order.

When considering these alternate definitions of a match, it is important to remember that so far we have only attempted to capture high-certainty matches. By loosening the definition of what constitutes a match, we would increase the chance of linking records that should not be matched. In general, as we start allowing for looser defined matches, we may need to move towards a probability model, and assign confidence scores to the various matching pairs.

5.2 Increasing Automated Matches

Ideally, we would like to increase the percentage of matches that are found using computer algorithms alone. This would allow us to reduce the amount of money spent outsourcing to the crowd. We could use the information about matches that we made in Delaware to create new models to process the data, perhaps applying machine learning. For example, we now have a dataset of known job transitions; using this data, we could extend the equivalence classes used to find equivalent jobs. Perhaps we could assign a probability to the transitions based on how frequently it was found in the Delaware job transition dataset.

5.3 Future Work

Assuming that the changes suggested above are implemented, and an entire state can be entirely disambiguated with minimal errors, the next step would be to repeat the same process on all states. We will then have to expand our search to look for transitions between states. Luckily, our contributions have been grouped into IDs, meaning that there are fewer sets we need to compare between. We would not be required to compare all contributions to each other, but only the contributor identities to each other.

Finally, performing a disambiguation going back to the beginning of the FEC data collection would allow disambiguation of the entire dataset, and make it possible to understand political behaviour for the past three decades.

Once the FEC database is fully disambiguated, political behaviour can be studied across the country. Similar analysis as performed in section 4.4 could be extended both geographically, to the full country, and temporally, going back to the beginning of data collection.

Further, other interesting topics such as political polarization could be studied. It is well known that political polarization in the Congress has increased over the past decades[46, 45, 28]. Understanding the interaction between the polarization of each member of Congress and that of their financial supporters could unveil interesting, possibly causal, relations. The level of political polarization of individuals could be determined by the ratio of aggregate donations given to candidates of each party.

Once a measure of individual political polarization is established, analysis could be expanded to social networks. In particular a website such as LinkedIn, which provides an individual's name, a list of employers, and a list of social contacts, could be used to identify individuals in the FEC database, as well as their friends. This would give insight into the state of political diversity within social circles, likely related to the diversity of political opinions that the individual is exposed to in their daily life.

Another avenue of study would consist of examining the demographics of the database. Gender and ethnicity could be determined with some certainty based on first and last names, and then be used to determine the political behaviours within these demographic groups. Additionally, lists of estimated net-worth, such as the Forbes 400, could be used to study the behaviour of the wealthiest individuals in the country.

CHAPTER 6
Conclusion

We have presented a disambiguation system which successfully finds, and assigns unique IDs to a substantial portion of political contribution data found in a state. We proved that our system works, by performing a disambiguation of Delaware. Our system can handle large states, by splitting the records up into progressively smaller sets, and comparing within these sets.

While there are improvements to be made, we are confident that the disambiguated data for Delaware places the majority of contributions into their correct identities.

We have shown that this data is beneficial to political analysis by detecting a concentration of financial activity in the political system. With the recent ruling in *McCutcheon v. Federal Election Commission*, the Supreme Court has removed the cap on cumulative campaign donations [1]. This decision increases the potential for even more concentration of political contribution activity. Access to a fully disambiguated dataset of campaign contributions will help track the ramifications of this legal change, as well as political behaviour going back three decades. We hope that the release of this data will help unlock many other avenues of research which were, up until now, inaccessible.

APPENDIX A
Failures of our System

Table A–1: D. Robert was placed in the same S2 set as someone else named Robert, rather than with their other records. The error was due to different given names being written in full.

		S3 ID #835		
Last Name	First Name	Occupation	Employer	Mailing Address
Buccini	D Robert	Vice President	Edward I Deseta	611 Adams Dam Rd
Buccini	Robert	Owner	Buccini Pollin Group Inc	908 Greenhill Ave

		S3 ID #833		
Last Name	First Name	Occupation	Employer	Mailing Address
Buccini	Donato	Vice President	Edward J Deseta Co Inc	611 Adams Dam rd.

References

[1] Mccutcheon, et al. v. federal election commission, April 2014. Supreme Court of the United States Case 572 U.S.

[2] Amazon. Amazon mechanical turk. https://www.mturk.com/mturk/welcome, 2014.

[3] Amazon. Faqs, amazon mechanical turk. https://requester.mturk.com/help/faq, 2014.

[4] American Bar Association. Lawyer demographics, 2011. Number of licensed lawyers - 2010.

[5] Melissa Bjelland, Bruce Fallick, John Haltiwanger, and Erika McEntarfer. Employer-to-employer flows in the united states: estimates using linked employer-employee data. *Journal of Business & Economic Statistics*, 29(4):493–505, 2011.

[6] Adam Bonica. Database on ideology, money in politics, and elections: Public version 1.0 [computer file]. Stanford University Libraries, 2013.

[7] United States Census Bureau. Frequently occurring surnames from the 1990 census. https://www.census.gov/genealogy/www/data/1990surnames, 2011.

[8] United States Census Bureau. State and county quickfacts, delaware. http://quickfacts.census.gov/qfd/states/10000.html, 2014.

[9] Jesse Chandler, Pam Mueller, and Gabriele Paolacci. Nonnaïveté among amazon mechanical turk workers: Consequences and solutions for behavioral researchers. *Behavior research methods*, pages 1–19, 2013.

[10] Peter Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer, 2012.

[11] Tim Churches, Peter Christen, Kim Lim, and Justin X Zhu. Preparation of name and address data for record linkage using hidden markov models. *BMC Medical Informatics and Decision Making*, 2(1):9, 2002.

[12] Matthew JC Crump, John V McDonnell, and Todd M Gureckis. Evaluating amazon's mechanical turk as a tool for experimental behavioral research. *PloS one*, 8(3):e57410, 2013.

[13] R.H. Davidson, W.J. Oleszek, F.E. Lee, and E. Schickler. *Congress and Its Members, Fourteenth Edition*. Congress And Its Members. CQ Press, 2013.

[14] FEC. About the fec. http://www.fec.gov/about.shtm.

[15] Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[16] John Folkesson and Henrik I Christensen. Closing the loop with graphical slam. *Robotics, IEEE Transactions on*, 23(4):731–741, 2007.

[17] Jim Giles. Learn a language, translate the web. *New Scientist*, 213(2847):18–19, 2012.

[18] Leicester Gill. Ox-link: The oxford medical record linkage system, 1997.

[19] Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F Naughton, Narasimhan Rampalli, Jude Shavlik, and Xiaojin Zhu. Corleone: Hands-off crowdsourcing for entity matching. Technical report, Technical report, UW-Madison, 2014. http://pages. cs. wisc. edu/˜ cgokhale/corleone-tr. pdf, 2014.

[20] Linda S Gottfredson. Circumscription and compromise: A developmental theory of occupational aspirations. *Journal of Counseling psychology*, 28(6):545, 1981.

[21] David A Griffith, Michael Y Hu, and John K Ryans Jr. Process standardization across intra-and inter-cultural relationships. *Journal of International Business Studies*, pages 303–324, 2000.

[22] Matthew A Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[23] Matthew A Jaro. Probabilistic linkage of large public health data files. *Statistics in medicine*, 14(5-7):491–498, 1995.

[24] Durga M Kandasamy, Kristal Curtis, Armando Fox, and David Patterson. Diversity within the crowd. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*, pages 115–118. ACM, 2012.

[25] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk. In *AMCIS*, volume 11, pages 1–11, 2011.

[26] Alexander Kawrykow, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Luis Sarmenta, Mathieu

Blanchette, Jérôme Waldispühl, et al. Phylo: a citizen science approach for improving multiple sequence alignment. *PloS one*, 7(3):e31362, 2012.

[27] Gregory Koger and Jennifer Nicoll Victor. Polarized agents: campaign contributions by lobbyists. *PS: Political Science and Politics*, 42:485–488, 2009.

[28] Geoffrey C Layman, Thomas M Carsey, and Juliana Menasce Horowitz. Party polarization in american politics: Characteristics, causes, and consequences. *Annu. Rev. Polit. Sci.*, 9:83–110, 2006.

[29] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[30] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences*, 108(22):9020–9025, 2011.

[31] Ian McGraw, Chia-ying Lee, I Lee Hetherington, Stephanie Seneff, and Jim Glass. Collecting voices from the cloud. In *LREC*, 2010.

[32] Andreas Myka and Ulrich Güntzer. Fuzzy full-text searches in ocr databases. In *Digital Libraries Research and Technology Advances*, pages 131–145. Springer, 1996.

[33] Harri Oinas-Kukkonen. Network analysis and crowds of people as sources of new organisational knowledge. *Knowledge Management: Theoretical Foundation. Informing Science Press, Santa Rosa, CA, US*, pages 173–189, 2008.

[34] OpenSecretsBlog. Money wins presidency and 9 of 10 congressional races in priciest u.s. election ever, November 2008.

[35] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.

[36] Joel Ross, Lilly Irani, M Silberman, Andrew Zaldivar, and Bill Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 2863–2872. ACM, 2010.

[37] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311(5762):854–856, 2006.

[38] Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278. ACM, 2002.

[39] Daniel R Saunders, Peter J Bex, and Russell L Woods. Crowdsourcing a normative natural language dataset: A comparison of amazon mechanical turk and in-lab data collection. *Journal of medical Internet research*, 15(5), 2013.

[40] Sascha Schimke, Claus Vielhauer, and Jana Dittmann. Using adapted levenshtein distance for on-line signature authentication. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 931–934. IEEE, 2004.

[41] U.S. Securities and Exchange Commission. Division of corporation finance: Standard industrial classification (sic) code list. http://www.sec.gov/info/edgar/siccodes.htm, 2011.

[42] Maurizio Serva and Filippo Petroni. Indo-european languages tree by levenshtein distance. *EPL (Europhysics Letters)*, 81(6):68005, 2008.

[43] Matthew Staffelbach, Peter Sempolinski, David Hachen, Ahsan Kareem, Tracy Kijewski-Correa, Douglas Thain, Daniel Wei, and Greg Madey. Lessons learned from an experiment in crowdsourcing complex citizen engineering tasks with amazon mechanical turk. *CoRR*, abs/1406.7588, 2014.

[44] James Surowiecki. *The wisdom of crowds*. Random House LLC, 2005.

[45] Sean M Theriault. The case of the vanishing moderates: party polarization in the modern congress. In *annual meeting of the Midwest Political Science Association*, 2004.

[46] Sean M Theriault. *Party polarization in congress*. Cambridge University Press, 2008.

[47] USPS. Postal addressing standards. http://pe.usps.gov/cpim/ftp/pubs/pub28/pub28.pdf, January 2013.

[48] Luis Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

[49] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[50] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494, 2012.

[51] William E Winkler. Matching and record linkage. *Business survey methods*, 1:355–384, 1995.

[52] Ben Wofford. How one brown student took down the nra. http://www.brownpoliticalreview.org/2014/02/brown-student-takes-down-national-rifle-association/, February 2014.

[53] David L Word, R Colby Perkins, United States. Bureau of the Census, et al. *Building a Spanish Surname List for the 1990's–: A New Approach to an Old Problem*. Population Division, US Bureau of the Census Washington, DC, 1996.

[54] Bo Zhang. Computer vision vs. human vision. In *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, pages 3–3. IEEE, 2010.