



et al. Health

Ellen Van Wyk, John Semerdjian, Ricky Holtz, William Chambers

Problem Statement

For most people, a trip to the doctor is part of a routine - and even when it's an emergency, they can feel confident that their doctor will have some way to treat whatever is ailing them. However, things become all but routine when a patient is diagnosed with a rare condition. In many cases, the diagnosing physician is not capable of providing treatment, and patients have to start a difficult journey to find a specialist that understands the nuance of the condition and state of the art in treating it effectively.

Our team learned about this problem when we read a comment posted online by a distressed parent. After exhausting his personal network for leads to specialists, the parent visited an online forum to ask strangers for guidance in searching for a qualified physician. Some responses urged the parent to parse through publically available medical journal metadata from PubMed, a database of 25 million biomedical journal articles and abstracts. After identifying articles relevant to his son's condition, and assessing their validity, the parent was supposed to cross-reference the article's authors with a national physician database to determine whether the author was a licensed clinical professional.

We were appalled by this absurd requirement to find treatment. Although the advice aligned with the National Institutes of Health's (NIH) recommendation for patients diagnosed with a rare condition, the task is extremely labor intensive and requires highly specialized knowledge. As a team of graduate students we knew this was an information access problem that, if improved, could have a big impact on patient health outcomes. With our backgrounds in data science and design, we knew that we were the team to help solve it. So we began the process of talking to patients and in the process observed several themes:

- The physician who makes a rare disease diagnosis may not necessarily be qualified to provide treatment or know a specialist to refer them
- The fragmented US healthcare system often forces patients to perform the search for rare disease clinical experts on their own
- While relevant healthcare provider and research metadata (e.g. journal reputation, author's clinical affiliation, date published, etc.) are freely available, they're difficult to find and comprehend
- Once patients have found the right information, it is dense and requires healthcare domain expertise to fully comprehend
- No single source of information measures the clinical research experience of health care providers

We found the same themes repeated by rare disease experts and researchers as well as confirmed by our independent research. This led us to identifying the aforementioned themes and gaining a better understanding of the systematic issue faced by patients diagnosed with rare diseases. In order to better understand their own conditions, these patients engage in what is known as a dynamic, or 'berry-picking', model of search.^{1 2} In the berry-picking model, people go from one source to another, refining their query as they acquire more information and a better understanding of the problem they are trying to solve. According to our interviewees, this often meant learning complicated medical terminology related to their rare condition. This process can take a significant amount of time, and while this method of searching can be valuable for somebody trying to gain domain knowledge, patients are most often seeking the best treatment, not the most knowledge.

Our interviews also suggest that the berry-picking search model is physically, mentally, and emotionally taxing even for patient's family members without a rare disease diagnosis. Many of the patients we interviewed, or parents/family members of patients, were coming home after long days at work or the hospital and devoting hours upon hours to researching the condition. This was reinforced by what we found across the web. We listened to segments on NPR where people talked about their feelings of anxiety and isolation after receiving a one-in-million diagnosis. We heard stories during interviews about patients being misdiagnosed with a condition for over a decade, only to find out the truth about their illness after relentlessly researching their symptoms and identifying the a clinical expert.³ In a few instances, patients remarked that they often knew more about their rare condition than their initial physician, a feat they only achieved after extensive research and independent study. For many others, they relied on the efforts of family and friends with connections to a specialist.

While the berry-picking model for search is effective, it's intensive. For people that are seeking treatment for themselves or a loved one, it should be reduced to what is known as a standard search model.⁴ In the standard search model an individual makes a query and gets a result that's meaningful to them. Patients and their families need a simpler, more focused solution that shows them specialists who can treat their rare disease.

¹ Bates, M. J. *The Design of Browsing and Berry-picking Techniques for the Online Search Interface*. 5th ed. Vol. 13. 1989. 407-424. Web. 06 May 2016.

² Hearst, Marti, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. "Finding the Flow in Web Site Search." *Communications of the ACM Commun. ACM* 45.9 (2002). 42-49. Web. 06 May 2016.

³ "Something Only I Can See | This American Life." *This American Life*. Web. 06 May 2016.

⁴ Hearst, Marti. *Search User Interfaces*. Cambridge: Cambridge UP, 2009. Print.

Existing Solutions

During our research we discovered several types of online solutions in the general space of helping to source a specialist for a given rare condition. These solutions fall into several distinct categories that each have significant shortcomings in the structures of their information systems. These groups are: Yelp and Yelp-like services, patient groups, and the NIH Call Center.

Yelp and Yelp-like services, like ZocDoc and Healthgrades only allows comparison across doctors based on user submitted reviews, which reflect the patient's subjective experience and not the doctor's expertise. While valuable to some, these reviews typically assess items that aren't relevant to clinical care such as the temperature in the waiting room or details relating to scheduling an appointment. Across these solutions, the information is too general to be useful to people with rare conditions. Since many rare diseases aren't recognized in the disease classification systems that healthcare providers use for billing, these solutions can't actually measure how many or how well doctors treat those patients.

Patients may also source information through online patient groups such as PatientsLikeMe. Patients are encouraged to talk to each other about their symptoms and treatments and recommend doctors who have treated them. The recommendations from other patients, however, may not be objective or useful for all patients with the same condition. For example, patients tend to concentrate on reviews of a doctor's bedside manner, while ignoring (and lacking qualification) to rate their clinical skill, similar to the problems encountered by Yelp or Zocdoc.com. Another problem with these communities is that the quality of the service they provide can vary dramatically. We analyzed the discussion taking place on these sites and found that some diseases had active communities with moderators and patients who would respond to requests for information. For other diseases and disease variants, a users' request for a doctor recommendation might never get an answer.

Lastly, the NIH provides a the Genetic and Rare Disease call center as a resource for patients to help patients find contact information for qualified providers by searching a number of the online resources described above as well as other more arcane sources like PubMed and clinicalTrials.gov. While they provide the most accurate service, the sheer number of diseases and varying patient needs limits their ability to provide a complete service. Given enough time they may be able to parse through thousands of documents about a rare condition but this is rarely the case simply because the process requires too much manual effort and time. . This process requires a patient or advisor to manually search through author metadata collected from clinical research tools such as PubMed and ClinicalTrials.gov. The NIH believes that clinical research activity is a reliable proxy for clinical skill and knowledge however given limited resources performing this task effectively is challenging even for those with domain knowledge. Ultimately, people are just trying to find a doctor - they should not need to become an armchair MD and information specialist to do it.

Our solution: et al. Health

et al. Health aims to put more information in the hands of patients to compare doctors and better navigate the vast amount of information available on the web. We do this by focusing on addressing the problem from two directions. One focus is efficient and accurate data parsing of the vast troves

of medical literature found on PubMed and other medical information systems and making this available via a REST API. The other focus is on creating a mobile-ready web interface to this information for patients with rare diseases in order to deliver the most accurate and most informative results concerning their condition. We begin by introducing how we focused on the users of our software and end by looking at the data extraction pipeline that we created in order to achieve this goal.

A Focus on the User

the patient

A person directly suffering from a rare disease. May not have a direct support network. Feels alone, because their condition isn't well-documented or supported by the healthcare system.



“Nobody knows what it feels like. Nobody really has it. Getting diagnosed is difficult, but getting treated can be just as hard.”

- Tired and/or otherwise incapacitated
- Has geographic concerns - worried about traveling far for treatment
- Knows a little about medicine

- Finding treatment to alleviate symptoms
- Activism around condition
- Hope

the doctor

Diagnosed a patient with a rare disease that they had never heard of. They care about their patient but it's outside of their responsibilities to find them a provider.



“I don't have time to give my patients the care they deserve. If I want to help, it's in my spare time.”

Key Characteristics

- Has too many patients to care for
- Averse to new technology/time wastes
- Medical expert, but not up to date in the latest/obscure fields

Driven By

- Professionalism
- Care for patient
- Saving time

the parent

A family member or close friend has a condition, and they are assisting in finding treatment. They spend significant amounts of their free time doing research on the condition.



“I was driving the questions. I feel like I had to - I owed it to my daughter. I was reading a lot of stuff I didn't understand.”

- No medical background but doing lots of medical research
- Tenacious; not afraid to challenge doctors and make demands
- Bouncing between care providers

- Love/familial tenderness
- Community and connectedness
- Finding optimal treatment for family member

Design Process

We followed a standard design process, beginning with user research and moving into wireframing and coding at increasing levels of fidelity. Fortunately, our team was small enough and nimble enough that we weren't too beholden to any specific process. We made it a point to continue our research throughout the semester, and we kept the entire team involved in user interviews. Furthermore, our research methods were a response to specific design challenges that we faced, rather than a part of a predetermined process.

Designing for the General Case

Since there are nearly 7,000 rare diseases identified by the NIH, we entered our user research knowing it would be difficult to generalize one solution across all of them. We chose to interview people in the Castleman and ALS (amyotrophic lateral sclerosis) communities because these diseases represent opposite ends of the disease publicity and frequency spectrum. ALS is a well-known rare disease gaining significant awareness with the popularity of the "ice bucket challenge" while Castleman is a relatively unknown and - frankly - less 'sexy' disease.

Our interviews revealed yet another layer of complexity: disease subtypes and symptoms. As an example, Castleman has 4 different subtypes, and a Castleman patient seeking a specialist would specifically look for someone who has researched that specific subtype. On top of that, they would prefer someone who had experience with someone of their demographics and symptoms. Despite the rarity of their conditions, or perhaps because of it, people seem to seek highly specialized care. Because of this, we implemented the categorization of disease subtypes.

Doctor Quality

et al. Health claims to filter for the most qualified specialists, but we found no consensus on what truly differentiates the 'best' specialist. Our interviewees brought up signals for doctor quality that lay outside the scope of our information system, including patient reviews, bedside manner, and leadership in the disease community. From our competitive analysis, we understand that et al. Health exists in an ecosystem of other resources to which we can direct people. While some patients saw quantity of articles as a proxy for quality, academics pointed out that a proliferation of research might indicate infrequent or neglected clinical experience. Doctors also said that they compared other specialists across very specific dimensions including protocols, methods, and funding sources. Based on these findings, we decided to expose as much of the articles as possible and link out for further reading.

Accessibility

While we entered the interview process expecting to meet users with a range of demographics and

abilities, our actual interactions with them made the importance of accessibility far more personal. One individual in particular was in the late stages of a disease and had no motor control in his body; he used eye movements to type and interact with the screen. This highlights the importance of highly accessible, easy to use, and intuitive design. Designing an interface that accommodates the wide range of accessibility needs in the community became a cornerstone of our information system.

Usability Testing

Doctor Rating

As described in the interview section, we struggled with recommending a doctor rating system. During usability tests, we deferred to user expectations, including an unlabeled 3 unit rating system and asking participants what they thought the rating meant. We're learning that it's difficult to get the three dot visual to send the right message; even after changing the rating from stars to circles, one user continued to refer to them as "stars". Another user asked us why we would even show doctors that have a low rating. Based on these findings, we decided to exclude a rating system to prevent unnecessary confusion.

Publications

The wireframes we showed participants included doctors' publications. Both expert participants and lay participants expressed that they didn't understand the abstract snippets, though the expert participants found certain keywords useful and wanted a link to the article in order to inspect some of the figures and methods. Though they didn't understand the content, three of the participants also expressed that they were impressed with the publications. This presents an interesting design opportunity, and an ethical conundrum. While it is typically a good practice to not overwhelm users with information they can't understand, the abstract information can still help us inspire our users' trust. In this same vein of thought we recognize the limitations of research methods, the peer review system, and general document health. We need to consider whether this trust is ill-earned and do further investigation into how to manage user expectations of academic articles. In the meantime, we have decided to progressively disclose the article information to accommodate all our personas.

Photos

Our wireframes included photographs of the doctors, and these were generally well received by participants. One expert participant in particular helped us justify the recommendation to keep the photographs in spite of the additional overhead they require; she explained that the process of finding a rare disease specialist involves a lot of uncertainty. Seeing a photo on their webpage and knowing who to look for when they get to the doctor's office provides a little more certainty in an otherwise tumultuous situation. We recognize the bias that comes with the addition of photography - we hate the idea that a quality physician might be overlooked for their gender, skin color, or garb. Again, we'll need to do further investigation to understand how to navigate this tradeoff.

Identifying and Linking Authors to Doctors

Our data process breaks down into three core tasks which are the focus of this section.

1. Tokenizing and parsing institutional affiliation information for each article to determine an author's geographic location
2. Canonicalizing author identities in order to determine all articles associated with a given author
3. Linking canonical authors to a list of known physicians

Properly executing on each step of the pipeline became important for us to achieve state optimal results. This required significant academic research which we summarize below.

Related Work

Conditional Random Field

Conditional Random Fields (CRF) are a class of discriminative models that predict sequences of labels for sequences of input samples. CRFs are similar to Hidden Markov Models in the sense that they operate over sequences of features to identify a latent value, such as a change in state or part-of-speech. We use CRFs as a preprocessing step for disambiguating an author's affiliation string.

Disambiguation

Disambiguation is the process of identifying unique entities and associated all possible variations of that entity with a canonicalized version. Our project involves two rounds of disambiguation: author names and author affiliations. Beginning with author name disambiguation, Torvik and Smalheiser describe a probabilistic approach using shared title words, journal name, co-authors, medical subject headings, language, affiliations, author name features (middle initial, suffix, and prevalence) and other recent enhancements.⁵ By extracting and comparing this information they achieved 98.8% recall using a dataset of 15.6 million articles from PubMed. Strotmann and Zhao compare traditional approaches using surnames and first initials to an algorithmic technique using last-author-based citation ranking and co-citation mapping, which are both relatively immune to the author name ambiguity problem that results from Romanized names of Chinese and Korean authors.⁶ Ferreira et al. provide a taxonomy for characterizing current supervised and unsupervised author name disambiguation methods and discuss open challenges.⁷ A majority of the surveyed methods perform disambiguation using some type of similarity function.

⁵ Torvik, Vette I., and Neil R. Smalheiser. "Author Name Disambiguation in MEDLINE." *ACM Trans. Knowl. Discov. Data ACM Transactions on Knowledge Discovery from Data TKDD* 3.3 (2009): 1-29. Web. 06 May 2016.

⁶ Strotmann, Andreas, and Dangzhi Zhao. "Author Name Disambiguation: What Difference Does It Make in Author-based Citation Analysis?" *J Am Soc Inf Sci Tec Journal of the American Society for Information Science and Technology* 63.9 (2012): 1820-833. Web. 06 May 2016.

⁷ Ferreira, Anderson A., Marcos André Gonçalves, and Alberto H.f. Laender. "A Brief Survey of Automatic Methods for Author Name Disambiguation." *ACM SIGMOD Record SIGMOD Rec.* 41.2 (2012): 15. Web. 06 May 2016.

Record Linkage

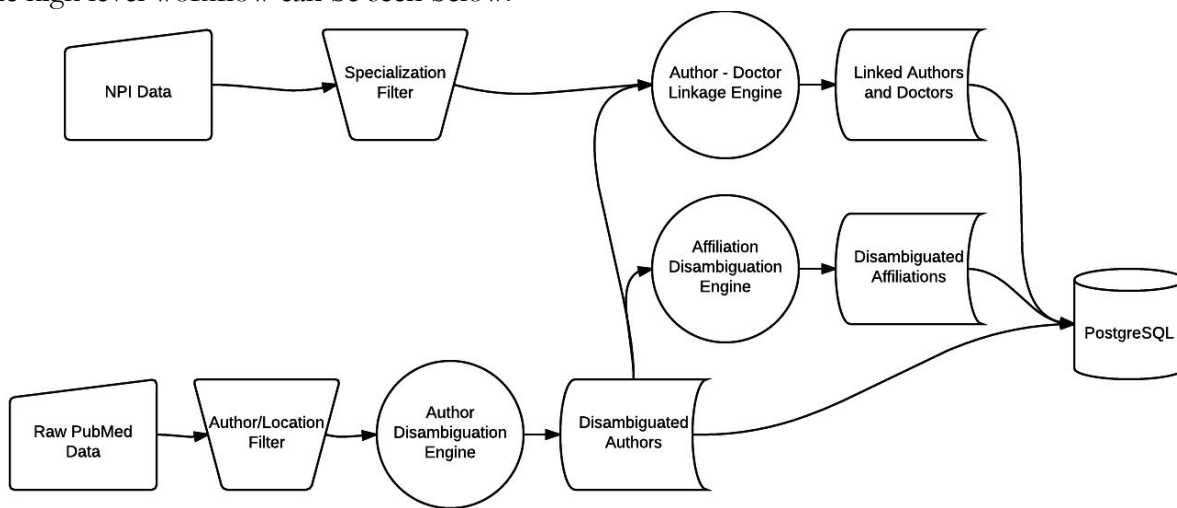
Jaro outlines a probabilistic strategy for record linkage in large public health datasets.⁸ He describes the concept of blocking (or grouping data into manageable subsets to reduce the number of pairwise comparisons), applying weights to fields to match on, and presents the AUTOMATCH record linkage algorithm. Christen provides a general outline of the process (see diagram) and outlines 12 variations of 6 different indexing techniques and implements each of them in *Febrl* (*Freely extensible biomedical record linkage*) record linkage system, written in Python.^{9,10}

One of the most important factors for efficient and accurate indexing is the proper definition of blocking keys. Sayers et al. describe the process of performing record linkage; pre-merge and record cleaning; the Fellegi-Sunter statistical framework; blocking and stratification; and evaluating linkage errors.¹¹

Methodology

Given the messy and heterogenous nature of our data, our process begins by excluding articles and authors that do not meet our basic specifications. Articles must be written in english, while authors must have a first name, last name, and an affiliation located within the United States.

The high level workflow can be seen below:



By initially focusing on building an end-to-end pipeline that ingests raw data and outputs probable list of physician-researchers, we have created a modular system that has allowed us improve each segment of the pipeline without having to write the complete system from scratch.

⁸ Jaro, Matthew A. "Probabilistic Linkage of Large Public Health Data Files." *Statist. Med. Statistics in Medicine* 14.5-7 (1995): 491-98. Web. 06 May 2016.

⁹ Christen, Peter. "A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication." *IEEE Trans. Knowl. Data Eng. IEEE Transactions on Knowledge and Data Engineering* 24.9 (2012): 1537-555. Web. 06 May 2016.

¹⁰ "Fgregg/febrl." *GitHub*. Web. 06 May 2016. <<https://github.com/fgregg/febrl>>.

¹¹ Sayers, Adrian, Yoav Ben-Shlomo, Ashley W. Blom, and Fiona Steele. "Probabilistic Record Linkage." *International Journal of Epidemiology Int. J. Epidemiol.* (2015). Web. 06 May 2016.

This allowed us to leverage improvements in author disambiguation later on in the pipeline in author - doctor linkage.

Affiliation Parsing

As described above, our first task is to filter authors and articles based on their research affiliation. This information appears to be manually entered by either editors or article authors and is therefore highly inconsistent and messy. In order to parse and label this data into structured fields for later use, we use a conditional random field classifier.

Given an input string -- e.g. "Division of Otolaryngology, The Children's Hospital of Philadelphia, Pennsylvania 19040-4399, USA." -- our objective is to tokenize and then assign one of eleven possible labels which include: department, hospital, school, research, company, address, city, state, zipcode, country, and other.

Once the affiliation string has been parsed, we then pass all geographic fields (e.g. city, state, zipcode, and country) and at least one additional token from school, research, or hospital to a geocoding service. The geocoder resolves geographic entities that may be similar, e.g. "California" and "CA", and then returns an approximate longitude and latitude. Geographic data is stored for later use, while the resolved geographic location and additional token are saved for the affiliation disambiguation step.

We used over twenty features for our CRF model, including sequential features such as the previous tokens associated feature values. These features included the presence of specific terms (e.g. "Hospital", "Institute", "University"), term/character counts (e.g. # of stopwords, # of digits, # of uppercase letters, string length), and punctuation used.

Author Disambiguation

The next step is to canonicalize and disambiguate authors that have met our inclusion criteria. Our disambiguation engine depends on the Python-based library Dedupe.¹² Dedupe combines many of the disambiguation strategies noted in our primary resources. In order to train a specific disambiguation model, we must first determine author features, measures of author similarity, and then manually label data. Instead of one long name string, we take advantage of structured fields using three common features: an author's last name, first name, and initials. Given our focus on authors for a given disease rather than all authors cited in PubMed, using a smaller set of features may still be adequate for this particular disambiguation task. Future work may incorporate co-author occurrence and the overlap of terms used in an article's title as measured by cosine similarity, but for now these features meet our needs.

For string comparisons, we've chosen to use the Affine Gap Distance, a common scoring system used in the field of bioinformatics, which is used to align a small portion of genetic code. This similarity measure is optimal for strings of differing length where alignment and gap insertion is crucial. The cost of extending a gap (a deletion or insertion) is less than opening the gap.

We leverage an active learning module within the Dedupe library for manual labeling. The active learning procedure in Dedupe prompts the user to manually determine whether two records are the same individual or indicate that they are unsure. Making pairwise comparisons between all records in dataset would take too much time, so we limit the search space by using Blocking. All records must have at least some thing in common, e.g. last name, and one of our goals is to

¹² "Datamade/dedupe." *GitHub*. Web. 06 May 2016. <<https://github.com/datamade/dedupe>>.

determine the rules that best cluster the data into a smaller search space. Dedupe uses predicate blocks, disjunctive blocks, index blocks, and various combinations of blocking rules. These blocking rules allow the user to quickly determine whether records are similar or not and greatly limit the number of comparisons necessary to achieve adequate performance.

Once the data have been labeled and blocking rules determined, feature weights are determined using logistic regression with L2 regularization. For example, the similarity of two full names is intuitively based more on the last name rather than the first name, which may include only a single first initial or perhaps a nickname. Determining these weights allows us to take this important consideration into account.

Lastly, once pairwise distances between records within blocks is completed, we cluster similar records using hierarchical cluster with centroid linkage. Points within some distance threshold from a centroid record are part of the same of the group. The threshold used for clustering is a free parameter that a user may modify based on their preferences.

Affiliation Disambiguation

The lack of standardization across affiliations strings makes it difficult to know at what level entities should be disambiguated. For example, many affiliations start with the specific hospital department followed by the full hospital name, city, state, zipcode, and country. There are also affiliations which contain only the hospital name. Furthermore, university names may also be associated with hospital, such as UCSF. We decided to ignore department-level information and instead disambiguate at the hospital, school, or research institute-level. For example, “Department of Pathology, UCSF Medical Center, San Francisco” should be associated with “UCSF Medical Center”. It may be also fine to associate “UCSF Medical Center” with “UCSF School of Medicine” or “University of California, San Francisco” or even “UCSF”. Our preference for reducing the number of clusters lead us to cluster all of the aforementioned entities together. We recognize the tradeoff between the specificity and generalizability of entity resolution, and the research reinforces that the optimal balance is still an open question.

Setting aside the variance of affiliation specificity, the disambiguation process is nearly the same as author disambiguation except for a few minor details. Following the affiliation tokenization, parsing, and geocoding process, we are left with several structured fields to use as features: city, state, “institution” (this is concatenation of just the “hospital”, “research”, and “school” fields), and the full affiliation string. Affine gap distance is used for comparing cities between records, while cosine similarity based on the intersection of terms is used for the longer strings: “institution” and affiliation. We made this determination based on trial and error. States between records are compared by a 0/1 measure of whether they match or not.

Author Doctor Linkage

Doctor author linkage also uses the same process as author and affiliation disambiguation, except this time each doctor/canonical author cluster must contain at most one record from each dataset. We again use doctor/canonical-author first name, middle name, last name, and add state based on a canonical author’s affiliation and a doctor’s practice location. Both canonical author and doctor location data may be out-of-date, we decided to include this feature and would allow the algorithm to empirically determine it’s value based on training examples. All fields for this step were compared using the affine gap measure, while state was compared using the 0/1 measure discussed above.

Analysis and Results

Given the several modeling steps along the way, it was important for us to determine what was important to measure and why. As we ingest new data and tweak our models, errors can compound all the way down our pipeline. We also lacked a held out test set to assess the performance of our entire pipeline, which lead us to manually sample records at various milestones to calculate performance metrics.

Beginning with our conditional random field classifier of affiliation labels, we observed promising results almost immediately. However, we decided not to formally evaluate the performance since this model functioned as a preprocessing step for our geocoder and affiliation disambiguation steps. For example, if we incorrectly labeled the the token “California” as a city, the geocoder may have still correctly identified the string as a state. One could then evaluate the performance of the geocoder, but since this service was not under our direct control, we did not believe it was a worthwhile activity.

For author disambiguation, we manually sampled 50 unique authors and found all associated names for each author. Our disambiguation model achieved 100% precision and 100% recall, with less than ten authors co-authoring more than a single article in our dataset. Affiliation disambiguation is more challenging to assess for reasons discussed in the “Affiliation Disambiguation” section above. Given a small sample of records, we achieved 98% precision and 75% recall on affiliation disambiguation.

To properly validate the performance of our doctor linkage algorithm, we obtained a list of known physicians that treat a rare condition from a rare disease patient community. We’ve manually associated these physicians with their national provider unique identifier, NPI. However, the patient group we are working with recognizes their list of physicians is not exhaustive. This prevents us from calculating recall or precision since we can’t truly determine the number of false negatives or false positives. In addition, if an author does not publish any research but is still present on the curated list, then we will never be able to identify them. With these important caveats in mind, we were still able to achieve precision of 18% and recall of 30% if we assume the curated list is exhaustive of all rare disease doctors. We intend to share our list of predicted doctors with our rare disease community colleagues to ascertain their relevance in the near future.

Web Application

Architecture

et al. Health’s front end was built using the Python-based Flask framework. Flask works well for relatively simple applications like ours; it handles data well, and most importantly, it allowed for every member of our team to work in Python. As the ‘lingua franca’ of the I School, this helped ensure that we stayed agile. Our views were made using the Jinja templating engine (which comes bundled with Flask) and SCSS. We used a full range of semantic HTML5 elements, as well as ARIA. Approaching our CSS, the site is styled mobile-first and with a Block-Element-Modifier (BEM) naming strategy to help reduce CSS conflicts. To this extent, each view in our application was

considered programmatically, taking into consideration principles of good software design: it works, it communicates intent, it's DRY, and it's fairly lean.

Although we used the regular Flask engine (along with flask-Scss, an extension to Flask) initially, we switched to Gulp as a way of handling our assets. Gulp has a robust library of tools for automating and optimizing CSS creation, and so it currently handles SASS compilation, autoprefixes our resulting CSS file for cross-browser compatibility, combines various media queries to reduce file size, and minifies the entire file. Additionally, Gulp calls Flask as a subprocesses, and watches across the server for changes, and reruns the Flask or SASS processes as necessary.

Conclusion

In this report, we present the motivations and foundations of et al. Health, a health information system designed to help patients with rare disease find qualified medical experts. We have approached this challenging problem from multiple perspectives, applying the principles of data science, information systems and user-centered design to build a conceptually and technically solid end-to-end data pipeline and web application. By combining publicly accessible data with an elegant user interface, patients with rare diseases can, for the first time, factor a medical expert's research experience into their selection of a healthcare provider.

As it turns out, others agree with the vision of et al. Health. We were finalists in the UC Berkeley Big Ideas Competition, an intercollegiate competition involving multiple rounds of written proposals and in-person presentations. et al. Health earned 3rd Place out of nearly 250 entrants.

A few months ago, we identified that patients need a simpler, more focused solution that shows them specialists who can treat their rare disease. Today, our website is available at <http://www.etalhealth.org> and our free API is available at <http://api.etalhealth.org>. We intend to add more rare diseases over time, with the goal of eventually reaching the 7,000 known rare disease that affect nearly 30 million people in the United States alone. We believe et al. Health can change the way patients make healthcare decisions, and we hope that we can start seeing real users soon!