From Data to Information: Automating Data Science to Explore the U.S. Court System

Andrew Paley
andrewpaley@u.northwestern.edu
Northwestern University

Andong L. Li Zhao
andong@u.northwestern.edu
Northwestern University

Harper Pack
harper.pack@northwestern.edu
Northwestern University

Sergio Servantez
servantez@u.northwestern.edu
Northwestern University

Rachel F. Adler
r-adler@neiu.edu
Northeastern Illinois University
Northwestern University

Marko Sterbentz
marko.sterbentz@u.northwestern.edu
Northwestern University

Adam Pah
a-pah@kellogg.northwestern.edu
Northwestern University

David Schwartz
david.schwartz@law.northwestern.edu
Northwestern University

Cameron Barrie
cameron.barrie@u.northwestern.edu
Northwestern University

Alexander Einarsson
aeinarsson@u.northwestern.edu
Northwestern University

Kristian Hammond
Kristian.Hammond@northwestern.edu
Northwestern University

ABSTRACT

The U.S. court system is the nation's arbiter of justice, tasked with the responsibility of ensuring equal protection under the law. But hurdles to information access obscure the inner workings of the system, preventing stakeholders – from legal scholars to journalists and members of the public – from understanding the state of justice in America at scale. There is an ongoing data access argument here: U.S. court records are public data and should be freely available. But open data arguments represent a half-measure; what we really need is open information. This distinction marks the difference between downloading a zip file containing a quarter-million case dockets and getting the real-time answer to a question like "Are pro se parties more or less likely to receive fee waivers?" To help bridge that gap, we introduce a novel platform and user experience that provides users with the tools necessary to explore data and drive analysis via natural language statements. Our approach leverages an ontology configuration that adds domain-relevant data semantics to database schemas to provide support for user guidance and for search and analysis without user-entered code or SQL. The system is embodied in a "natural-language notebook" user experience, and we apply this approach to the space of case docket data from the U.S. federal court system. Additionally, we provide detail on the collection, ingestion and processing of the dockets themselves, including early experiments in the use of language modeling for docket entry classification with an initial focus on motions.

ICAIL'21, June 21–25, 2021, São Paulo, Brazil

ACM ISBN 978-1-4503-8526-8/21/06…$15.00
https://doi.org/10.1145/3462757.3466100

CCS CONCEPTS

• **Information systems** → **Decision support systems**; • **Applied computing** → **Law**; • **Computing methodologies** → **Natural language processing**; • **Human-centered computing** → **Natural language interfaces**.

KEYWORDS

notebook interface, information extraction, data analytics, natural language processing, visualization

ACM Reference Format:
Andrew Paley, Andong L. Li Zhao, Harper Pack, Sergio Servantez, Rachel F. Adler, Marko Sterbentz, Adam Pah, David Schwartz, Cameron Barrie, Alexander Einarsson, and Kristian Hammond. 2021. From Data to Information: Automating Data Science to Explore the U.S. Court System. In *Eighteenth International Conference for Artificial Intelligence and Law (ICAIL'21), June 21–25, 2021, São Paulo, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3462757.3466100

1 INTRODUCTION

In the United States, the federal judicial system serves as a vital umpire, ideally ensuring equal protection and justice under the law. Its mechanics are a tapestry of countless unique decisions made by individuals across 94 district courts, 13 circuit courts and the Supreme Court. While this system is at the core of the management of justice in the United States, its operation is essentially inaccessible. Distributed decision making, inaccessible data, and the general public's lack of technical skills sufficient to analyze data mean that the actual mechanics of U.S. justice are largely obscured. As citizens, we trust our laws are being enforced equally but – absent the occasional headline-grabbing case – it's all opaque to the majority of us.

One issue is that data access is prohibitively expensive. Public Court records are available only through a paywall called the Public Access to Court Electronic Records (PACER) system. Other research

has explored such limitations on access [1, 35], as well as the questionable completeness of the data available [30, 32]. Legislation to eliminate the PACER fees is progressing through Congress – one step towards opening the courts to public scrutiny and understanding.

But making court records free won't eliminate all barriers to access. Many open-government initiatives in the U.S. and abroad have yielded a growing array of public datasets [2, 24], and work towards data transparency is an ongoing effort [14]. However, while access to data is necessary, it's insufficient: the applied value of that data to the end goal of increased public understanding – of access to information – remains stymied by the limited analytical skills and resources of the majority of those afforded that data. A survey detailed in [47] found in part that while citizens acknowledge and appreciate moves towards open data, most don't know people in their social circles who take advantage of it. Further, the authors note "most open data released by the government is available in the raw format, which restricts its understandability by all people" and that "this data is mostly usable by experts with some technical knowledge to interpret and develop applications" [47]. Separately, in a case study of Data.gov, [23] argue that open data "generates its value when it is not only available and accessible but also made sense by its users to solve problems" and conclude that "public agencies should invest in new technologies and craft new data management techniques to make data readily accessible to users...providing real-time analysis and updates."

To date, the bridge between raw data and meaningful information has generally been built ad-hoc and on-demand by data scientists, but that resource-intensive approach doesn't scale when considering the information needs of a broader subset of the public. And, in the space of the legal system, even questions as simple as, "Are there differences in how judges handle fee waiver requests?" or "Is there any correlation between a judge's tenure and the length of cases they oversee?" are impossible to answer without significant data expertise or the resources to pay for it. Clearly, open data access isn't enough; we need a mechanism to access the information contained within.

To build that mechanism, in essence, is to automate work that would be done by a data scientist to extract information. Thus, we endeavor to outline what the data scientist's role entails and identify those functions as requirement sets for building the platform.

1.1 The Domain Expert/Data Scientist Interaction

One set of requirements mirrors the domain expert/data scientist interaction: the ability to understand the user's intent and translate that into queries and analysis and to provide guidance and guardrails around what's possible given a dataset – and then to translate the results of analysis back to users in a way that is intelligible to them.

To help us better understand and frame the set of potential users in the space of the U.S. court system, we conducted 28 sets of interviews with a total of 38 people (25 male and 13 female). Some of those interviewed included faculty in law, sociology, and economics; lawyers; and journalists. Participants generally reported wanting to answer advanced questions beyond their analytical capabilities. They felt they were limited by the tools they were currently using and wanted to ask questions of the data that they weren't able to.

To help bridge that gap, we introduce a novel platform and user experience that provides users with the tools necessary to explore data and drive analysis via natural language statements. Our approach leverages an ontology configuration that adds domain-relevant data semantics to a database schema for the sake of supporting search and analysis without user-entered code or SQL. This configuration allows us to abstract away the underlying schema complexities from user concern, understand what filters and analysis are possible and domain-relevant, infer relevant analytics from the data semantics, and provide guided outcomes during both search and analysis.

The associated notebook-style experience is an early embodiment of a new form of human-data, or human-information, interface – a user experience imbued with a set of assistive capabilities where interactions happen in natural language rather than code. The system also generates responses in modalities intuitively appropriate to the nature of the analysis results – from text to various types of visualizations.

1.2 The Data Scientist/Data Interaction

The second set of requirements mirrors the data scientist/data interaction: the wrangling of data into coherent and controlled schemas through various modes of ETL (extract, transform, load), text extraction, data cleaning, and the more complicated arenas of machine learning and language modeling.

To support explorations of the U.S. court system, this includes the structuring and harmonization of court records, with the initial focus here on a snapshot of roughly 270,000 case dockets. This involves consultation with domain experts; the definition of a complex schema (across 30 tables ranging in size from two to thirty-one columns); a pipeline to extract, transform and harmonize the unstructured and semi-structured components of dockets; the integration of additional datasets to expand the information space (starting with background information on federal judges); the creation of a novel dataset for training language models for classification tasks (initially for classifying various types of motions within the scope of a case); and the model training/fine-tuning and validation process in pursuit of proving the utility of framing motion type detection as a classification task.

1.3 Automating Data Science

In sum, those two tracks build to our end goal: to democratize access to information associated with the U.S. court system, eliminating barriers to access and understanding, and providing journalists, legal scholars, lawyers, government officials, social justice advocates and others with relevant information derived from data about the mechanics of the federal courts.

We first detail our primary novel contribution – the platform for information exploration and analysis, and the natural language notebook frontend – and then provide detail about the ETL and data enrichment processes that serve as a backdrop for the search and analyses this instantiation supports. We engage in user testing and report preliminary results as well as explore how our platform's capabilities map to existing data science approaches through a case

study. Our discussion elaborates on the goals of our work, including challenges to be addressed.

Our approach to court docket search and analysis is one early step in the development of an open-source platform aimed at democratizing access to information. In discussion of future work, we outline dual and distinct tracks: the first aimed at continuing to build and augment our U.S. court records database, and the second focused on the ongoing development of the core platform. On the platform side, we point to a future in which additional data can be brought in by technical users who manage data wrangling and define data semantics – the steps we now think of as getting to "open data," but with a newly imagined purpose – and our system scales to new domains, communities, and geographies.

2 RELATED WORKS

Reducing the costs associated with PACER has been pursued as a way to achieve judicial transparency. However, studies have shown the limits of open data in providing greater transparency [44]. Notably, problems persist across many user personas, from citizens to data scientists, government agents, and even academics [7, 18, 19, 23]. We aim to address a subset of these challenges – pertaining to data utility and barriers to information access – by applying automated analytical and visualization capabilities on top of the data.

Much research has focused on automating legal processes [28], predicting outcomes [5], or assessing the value of AI for the two former areas [13, 45]. There have been some recent developments in legal question-answering (QA) systems [11, 21]. However, these have had limited data analytics capabilities [22] and often rely on simple data retrieval for generating answers [33]. While some commercial tools support exploration of court documents they are prohibitively expensive, and limited in terms of scope and consistency of results [1].

More broadly, general QA systems have been the subject of research for decades [16, 40, 46] and are some of the most prominent examples of AI systems [12]. There has been significant progress in neural QA systems [10], with transformer-based models [27] achieving state-of-the-art results on benchmark tasks [39]. However, these QA systems are best suited for unstructured text data where the answer is plainly stated in the corpus itself, unlike our system which can infer or derive the answer through follow-on analysis. Other approaches aim to understand and decompose the structure of complex questions into discrete parts as a plan for deriving an answer [49]; however, the representation is high-level and distinct from our approach which constructs runnable queries against a given datasource.

Extensive research has parsed natural language queries into SQL queries [20], using techniques from deep learning [17], rules-based methods [42], or a mixture of both [43]. Instead, our approach automatically generates the space of possible analysis from an ontology configuration, and then translates the underlying analysis plans to natural language, drawing inspiration from prior work [37, 41].

Beyond current work in information retrieval via conversational systems, our approach utilizes a notebook-style interface, with inspiration coming from Jupyter notebooks [38] as well as their forebear, IPython [36]. Automated visualization is a related area of research [31, 50] focusing primarily on presentation layers for a given dataset rather than intent-driven question-answering.

3 THE NATURAL-LANGUAGE NOTEBOOK

Notebook-style interfaces are a standard part of the modern data science toolkit, and for good reason: they support a logical process flow and marry exploratory and presentation layers in one cohesive experience. However, they are the tools of experts – users who know how to code, run analysis, interpret stack traces and explore complex results. They bring order in the form of scaffolding, but remain largely agnostic about content or the specifics of a particular dataset or domain.

We borrow from that scaffolding, but our system leverages simplified data filtering mechanisms and natural language statements. And where other notebooks display a variety of outputs (defined by the near-infinite space of possibility supported by arbitrary code), our system outputs natural language and annotated visualization as a means of conveying information.

This approach maintains the intuitive flow of the notebook user experience but brings its power to people unfamiliar with programming. Our notebooks are domain- and dataset-aware, and the user experience speaks the language not of the data scientist, but of a user reasonably fluent in the domain. Further, they provide assistive mechanisms to surface what the system knows it is capable of, guiding even novice users to understand the range of capabilities available to them.

An exploration of the current iteration of interface mechanisms and output capabilities can be found in Figure 1 and Figure 2, and a deeper discussion of the paradigm follows.

3.1 The UX Paradigm: "Search First, Then Converse"

Our approach separates concerns between search (winnowing the available dataset to a space of interest) and converse (the user inputting statements that drive analysis upon the filtered dataset and the generation of responses). This approach embodies the strengths of the notebook format in focusing on one task at a time and presenting interstitial output as feedback. Further, this has the indirect effect of separating concerns on the backend, supporting a generalizable approach to the specification of filter and analysis configuration.

As depicted in Figure 1, each exploration in our notebook interface starts from a "search" (or "filter") panel: a paginated view of the dataset that matches the current set of filters. The primary entities presented in this view are court cases in the Northern District of Illinois. On initialization, the user is presented with the full space of available data absent any applied filters and can opt to apply filters or skip right to adding analyses of the full dataset.

Below that is the partitioned "converse" step, where the user can enter natural-language statements that drive analysis (Figure 2). Of note, users can enter multiple analysis statements against one data view, stepping through a set of questions while maintaining a thread of prior exploration.

This paradigm means our system does not have to manage statements like "Average case duration grouped by judge tenure for cases

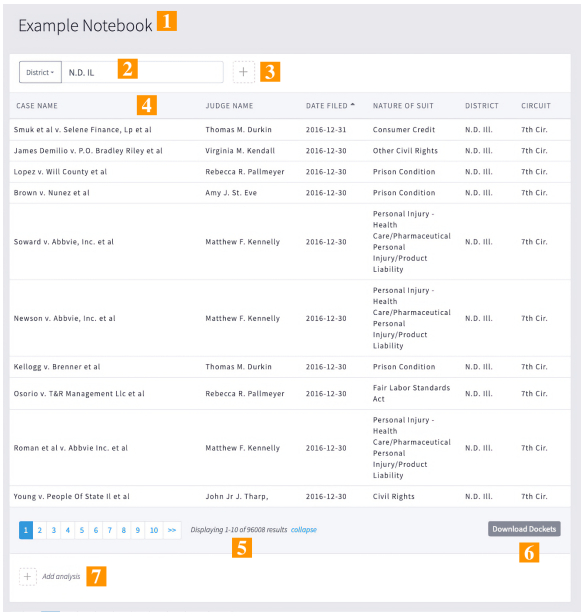

Figure 1: The primary interface for search/filtering within our notebook format. Annotations: 1) Notebook title, 2) Applied filter example, 3) Mechanism for adding filters, 4) Results interface with sortable columns, 5) Pagination and full results space, 6) Download button for raw data access, 7) Mechanism for adding analysis

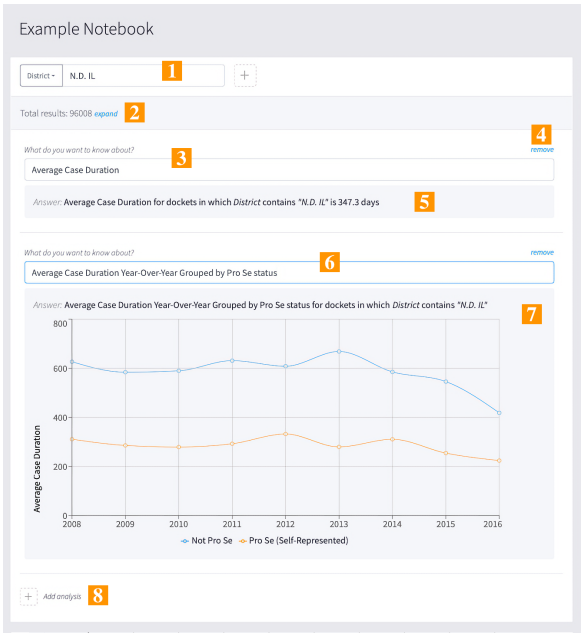

Figure 2: The primary interface for running analysis within our notebook. Annotations: 1) Applied filter example, 2) Collapsed results panel (seen in detail in Figure 1), 3) Analysis statement, 4) Mechanism for removing previous analysis output, 5) Result of analysis (basic NLG type to deliver single value), 6) Second analysis statement, 7) Result of analysis (interactive line chart with rollover states to display change over time, including legend with terms of art and associated definition), 8) Mechanism for adding additional analysis

that occurred in the Northern District of Illinois since February 2015 and involved property rights," (or require the user to repeat parts of this cumbersome statement for additional analysis), but instead simple, widely applicable statements like the "average case duration grouped by judge tenure" in the context of a previously filtered set of data. Thus, given an ontology and available analytics, the space of analysis statements is finite, but is made virtually infinite by the possibility of applying them to any slice of the data.

3.2 Mechanisms and Underlying Configuration

Necessary elements are abstracted out of the platform's core search and analysis engines as well as the user experience framework, API mechanics, and proactive caching and pre-fetching mechanisms. The system requires only a pointer to an SQL database, an object-relational mapping (ORM) defined in the open-source SQLAlchemy library [34], and an ontology configuration that references that ORM in order to provide all functionality, from generating the available filters and analysis statements to building queries and running analytics based on user input. These capabilities mean that additions to the underlying data schema or the onboarding of complementary datasets can be made available through the platform with only a small addendum to the already-required work of data management: the creation of or updates to the ontology configuration. See Figure 3 for a high-level architecture; specifics about the configuration follow for both the search and analysis components.

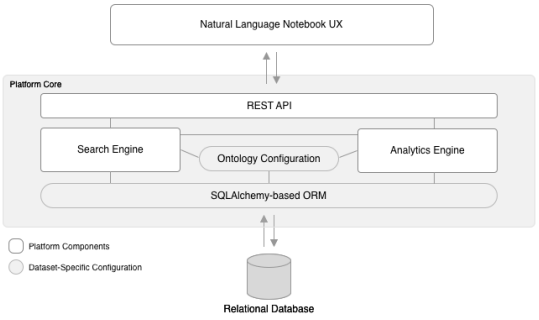

Figure 3: A simplified architecture diagram to illustrate the relationship between the different components of the platform and the dataset-specific Ontology Configuration and ORM

3.2.1 Search/Filter. As depicted in Figure 1, filters are applied by 1) adding them to the filter bar above a given data view, 2) selecting a filter type via the dropdown, and 3) entering values in the associated input. The set of additive filters that can be applied to winnow down the list of case dockets includes: district, circuit, case name, cause

of action, case status, filing date, nature of suit, party name, judge name, attorney name, as well as free text search in the docket entries associated with the case. Ultimately, users can make a few targeted selections and fill in a few inputs to get to searches equivalent to "all cases in the Northern District of Illinois between 2015 and 2017 where Kennelly served as judge" or "all cases with nature of suit property rights where one of the parties is Apple" – the SQL query versions of which only a fraction of those users could generate themselves.

The focus on case dockets being the "primary" searchable unit (as opposed to judges, parties, attorneys, etc.) and all associated filters are entirely configuration-driven and distinct from complexities of the underlying schema. The ontology config maps the machine representation to a user-friendly set of names and attends to the scaffolding of ids, foreign keys and joins. Thus, the filterable fields are a subset of those that exist at the schema level on various tables that join against the case table, and in some cases (such as Judge Name, depicted in Figure 4) actually span multiple fields in the schema (first_name, middle_name, last_name). The key point here is that the user does not need to consider the schema but simply makes decisions about domain-relevant ways to search with guidance from the system about the relevant search space (and the system then generates runnable queries of various types, including string matching and range finding, such as with dates). For domain expert users, our approach is a significant convenience over having to learn or write SQL, and, for less knowledgeable users, it also serves as guidance about relevance in the domain.

```
{...,
    "judgeName": {
        "nicename": "Judge Name"
        "type": "text",
        "allowMultiple": True,
        "autocomplete": acs.getJudges,
        "model": [db.JudgeOnCase, db.Judge],
        "fromTargetModel": ["judges", "judge"],
        "fields": ["first_name", "middle_name", "
            last_name"],
    },
... }
```

Figure 4: The config entry for the "Judge Name" entity for search/filter capabilities and the results view. 1) "nicename" is the user-facing name of this entity type, 2) "type" and "allowMultiple" inform the input style and query generation mechanisms, 3) "autocomplete" maps to a method on the autocomplete class (can be default or a plugin) and powers the autocomplete API endpoint, 4) "model" and "fromTargetModel" map the model join and relationship feature path from the db.Case table at the ORM level, 5) "fields" defines the field(s) this entity's name/id maps to (affording support for multi-field queries)

3.2.2 Analysis Statements and Query Generation. Once the user has arrived at filtered data they are interested in, they can add multiple analysis statements below the dataview panel. As seen

Figure 5: Two examples of the primary interface for adding analysis statements. Annotations: 1) The user-entered statement, having been auto-completed progressively via generated statement candidates, 2) Candidate matches given the previously auto-completed statement ("Average Fee Waiver Grant Rate Year-Over-Year") and the subsequently user-appended "grouped by," 3) A user-entered string that hasn't yet been auto-completed, demonstrating fuzzy string matching, 4) A set of fuzzily matched results

in Figure 5, this is realized on the UX as a fuzzy (i.e., approximate string matching) search across a set of natural language statements, each of which is generated dynamically by the system through inferring relevant analysis possibilities based on the underlying ontology configuration and a core model of analysis types. Because the system is inferring and defining the analysis space based on the ontology components, each generated statement corresponds to an underlying plan representation that is interpretable by the analysis engine for the sake of generating queries and running analytics on any set of filters.

As users select and add additional analysis statements to the notebook, the system responds with answers in the form of text and visualizations, as depicted in Figure 2. As per standard notebook mechanics, each analysis statement is tied to the active filtered set in the panel above such that changes to the filters (and thus the slice of the data presented) will flow through and update each linked result.

The core platform's model of analysis includes a growing set of available operations (e.g., average), as well as specifications on how the operation ought to be performed (e.g., can only be done on numeric fields, how many fields are needed). As seen in Figure 6, the ontology configuration then defines the fields relevant for analysis and their user-friendly names, as well as their attributes (e.g., semantic type, possible transformations into other data types, relevant units, and – in the case of discrete entities delimited by id – how to generate their user-friendly names) and their relationship to the primary model.

To illustrate the generation of the analysis space, in the instantiation referenced in our figures, an analysis configuration that lists ten relevant features for analysis (e.g., Judge Tenure, Nature of Suit, Case Duration, Fee Waiver Grant Status) alongside the metadata

```
{...,
    "caseDuration": {
        "model": db.Case,
        "field": "case_duration",
        "type": "float",
        "name": ["Case Duration", "Case Durations"
            ],
        "unit": ["day", "days"],
    },
... }
```

Figure 6: The config entry for the "Case Duration" feature as an analysis target. 1) "model" and "field" define where in the schema the relevant field(s) exist, 2) "type" informs the available analyses for that field, 3) "name" and "unit" define the singular and plural forms for user interaction/presentation.

associated with each is sufficient to generate 120 different possible analyses, each of which can be applied to any filtered view of the data. Augmenting this list of generated analysis statements requires simply adding new elements to the configuration or augmenting the core system's analytics library and tying new analytics to data or semantic types.

To compute an analytics statement at runtime, our system steps through the process of building an analysis chain and SQL queries based on the filters and the statement's underlying plan. The steps to do so are: 1) Do any necessary filtering (as specified by the search context the given analysis statement is run in); 2) Query the necessary fields from the analytics statement; 3) If needed, transform data to ensure data type compatibility; 4) Perform the necessary operations and grouping; and 5) Format the results based on the nature of the information to be conveyed.

For instance, if we wanted to know the "Average Fee Waiver Grant Rate Grouped by District" for cases where the Nature of Suit is "Property Rights," the steps would be: 1) We filter to get only cases where "Property Rights" is listed in the nature of suit field (leveraging the ontology mapping to the schema); 2) We query the fields associated with Fee Waiver Grant and Court District (again leveraging the ontology mapping); 3) Since we are taking an average rate and Fee Waiver Grant is stored as booleans, we convert it into integers; 4) We compute the average rate of Fee Waiver Grants for each Court District; 5) We convert the internal database Court District id into human-readable labels leveraging the name and units information from the ontology config.

Finally, the system output – the result of running the analysis statements above – is delivered in a form best-suited to conveying the nature of the information for each result (and includes a description of the filters applied for added clarity). This is keyed off of features of the results themselves. As depicted in Figure 2, running "Average Case Duration" – a bit of analysis that will yield a single value – results in the system rendering the results via basic natural language generation mechanics. However, when looking at something "Year-Over-Year," the system pivots into change-over-time behavior, leading to the generation of an interactive line chart.

4 THE DATA PROCESSING PIPELINE

PACER, the official source for federal judicial records, houses a variety of document types and charges a per-page fee for access. Following the recommendations set forth in [35], we focused on the docket reports, "essentially a lawsuit's table of contents." [35] identified efforts to improve accessibility of these docket reports as the "most impactful" work to be done in building a more open justice system. Thus, using docket reports as our primary data source, we designed a 30-table database schema (plus relevant join tables) to represent them and all relevant entities (e.g., judges, attorneys, defendants, and districts) as well as key fields (e.g., nature of suit, date of filing, and docket entries).

Our initial dataset captures samples of both depth (ten years of docket reports from Northern Illinois district courts from 2007 to 2016) and breadth (docket reports from every district court in 2016). In total, our sample draws from more than a quarter-million case dockets in HTML format acquired through purchase and batch downloading from PACER. Taking advantage of their semi-regular structure, we parsed the files into meaningful sections (e.g., the docket header) and extracted information. While most information we extracted was listed explicitly on the docket report (e.g., case title), we also captured implicit (e.g., case duration) and interpreted (e.g., party acting as their own attorney) information. In this latter case, we relied on guidance from domain experts to analyze both docket text (e.g., recognizing "Pro Se" designations for attorneys) and docket contents (e.g., verifying instances of self-representation where a party has only one attorney, who is also the party). Such interpretation often required triangulation between multiple approaches (e.g., a "Pro Se" designation alone is insufficient for classifying self-representation; one must also count and check the attorneys).

5 FURTHER DATA ENRICHMENT

To garner a more complete data-level representation of the mechanics of the judicial system, we sought to enrich the initial core docket dataset in two distinct ways: blending additional sources and the use of language modeling for classification tasks.

5.1 Additional Sources

We blended additional data into the core database, serving both as supplemental fodder for search and analysis and in support of initial forms of entity disambiguation. We leveraged the Federal Judicial Center's database of appointed federal judges [6], which includes birthdate, gender, race/ethnicity, history of appointments, appointing parties, education, and professional career. We normalized the data into a multi-table schema, linked it to the extracted representations of judges, and then leveraged that join to expand the space of available analysis. For example, based on the judge's appointment date and the start of a given case, we derived how long a judge was on the bench prior to the start of that case, and then used that "judge tenure" as a metric in subsequent analysis (e.g., to derive "fee waiver grant rate grouped by judge tenure").

5.2 Classification Tasks

We identified a variety of information targets that we believe can be culled from the unstructured text components of dockets by

reframing them as classification tasks. For context, the main body of a docket is a series of time-stamped text entries, each marking events in the arc of a given case. These text-snippet representations contain various sorts of useful information, including motions (effectively discrete requests for a judicial decision), the outcome of a given motion, changes of representation or venue or presiding judge, references to evidence or testimony, eventual outcomes, and so on. Being able to identify and classify such information would prove highly valuable for both search and analysis.

To explore approaches, we started with the classification of motion types as our initial target. At first glance, it could be tempting to envision a solution to this classification task based on regular expression where motions are explicitly identified by name. However, as depicted in Table 1, a pure regex approach is far too rigid to capture the many complexities found in the docket entry sample space, including multiple motions being named in a single entry, non-motion events referencing motions by name (e.g., notices, orders), and obfuscation of the motion type through varying levels of docket entry metadata. These complexities are further compounded by naming convention variations across districts and the trappings of error-prone human data entry [1].

Thus we pivoted to language modeling. As no training dataset exists for such a task, we created a web application to view and tag the motions pulled from our docket dataset. For both the definition of the space of possible motion types and for the sake of actually tagging the motion entries, we solicited help from legal scholars and their law students. We implemented a voting mechanism in the app such that each motion will be tagged three times by three distinct users as a means of ensuring accuracy. Our dataset continues to grow through use of the application, though the experiments that follow leverage a subset of this data.

In order to effectively utilize this data for our classification experiments, we performed some preprocessing on the raw dataset. First, the raw dataset contained several motion classes with few data points. To address these rare motion classes in these initial tests, we set a threshold of 25 data points and merged all classes below this threshold into the "Other Motion" class. Second, we removed all duplicate docket entries that arose as a byproduct of the voting mechanism from the dataset to ensure that the models were not training on some docket entries more than others. After this preprocessing, the smallest motion class contained 25 samples, the largest contained 951, and the median and average of the motion classes were 50 and 152, respectively. For each of the models we used a train/validation/test split of 80/10/10 per class. In total after preprocessing, there were 2,064 training samples with 524 testing samples across 17 distinct motion classes.

Making use of two pretrained transformers, the 110M parameter BERT-base [9] and 125M parameter RoBERTa [29] models, we fine-tuned each on this processed dataset. We made use of the AllenNLP framework [15] as a wrapper around the Huggingface Transformers library [48] to fine-tune the models for 10 epochs, using a batch size of 8, and the AdamW optimizer. The RoBERTa model achieved training accuracy of 95.69%, validation accuracy of 91.22%, and test accuracy of 90.08%. The BERT-base model achieved training accuracy of 96.95%, validation accuracy of 89.69%, and test accuracy of 89.31%. These results exceeded the baseline bag of embeddings classification model, which achieved a test accuracy of 80.50%. For the transformer models, the training accuracy is slightly higher than the validation/test accuracy for both models, but we believe this margin is reasonable given the small size of the dataset. To reduce the likelihood of overfitting in the future, we continue to grow the tagged motion dataset.

6 EVALUATION

We evaluated our system's effectiveness in handling both search and analysis of data across two separate tracks: 1) usability testing in which target users completed tasks with the system and provided survey feedback, and 2) a case-study comparative analysis to assess the system's efficacy when benchmarked against a data scientist's ad hoc analysis.

6.1 Usability Testing

We gave 15 subjects (14 legal professionals, one journalist) a set of prompts (e.g., "For all cases in the 'N.D. IL' district, which year had the highest average case duration?") and assessed their experiences in: (1) using the search filter, (2) conducting analysis on all the records, and (3) conducting analyses based on specific search criteria of varying complexity. In addition, we gave them time to test their own scenarios while "thinking aloud" so we could capture their intentions and strategies. Participants were then presented with a survey to complete at the end of the session consisting of the modified System Usability Scale (SUS), an evaluation framework shown to be effective at quantifying the complexity and ease of use of interfaces [4]. The average SUS score for our participants' overall experience across (1), (2), and (3) was 72.83, which is considered good usability [3, 26]. When answering the statement regarding whether they would use our system frequently, all but two participants (87%) agreed or strongly agreed with that statement (one was neutral and one disagreed and wrote that docket sheets are not used in their research). These results represent a preliminary round of user testing, and we intend to further analyze the associated feedback and conduct additional user tests targeting users with a wider variety of backgrounds.

6.2 System Evaluation: A Case Study

To further weigh the benefits of our approach, we compare it with prior work done by data scientists examining how fee waiver grant rates vary among judges using ad hoc data processing and analysis [35]. We answer the same question using our system (see Figure 7 for an example of the output) and through observation compare both approaches across three dimensions: speed to insight, flexibility of exploration, and barrier to entry.

The initial data processing pipeline looks similar for both. A systematic analysis of this issue requires paying to download case documents, creating an ETL process to structure the data, and identifying the fee waiver status of each case [35]. However, where the ad hoc method attends to ETL, aggregation, and visualization for a single target task, our approach looks to leverage that upfront data work to support a wide array of possible downstream analyses.

Thus, when considering a one-off query or single data point, we cannot definitively say that the ETL, schema and ontology work in support of our system will require less time than a data scientist taking the ad hoc approach. But one-offs aren't the goal of our

Motion Entry	Motion Class
MOTION by Defendant [Name Omitted] for extension of time to file response/reply as to motion for summary judgment 20 (Unopposed) ([Name Omitted]) (Entered: 05/18/2017)	Motion for Extension
Proposed Order re 13 MOTION to Stay Proceedings Pending Transfer by [Name Omitted]. ([Name Omitted]) (Filed on 4/16/2012) [Transferred from California Northern on 4/18/2012.] (Entered: 04/16/2012)	Not a Motion
Motion by [Name Omitted], Cook County Board of Review, [Name Omitted] for Leave to Cite Additional Authority ([Name Omitted]) (Entered: 01/13/2011)	Motion for Leave
Plaintiff's motion for leave to file a first amended complaint 54 is granted. WRITTEN Opinion entered by the Honorable George W. Lindberg on 4/29/2011: Signed by the Honorable George W. Lindberg on 4/29/2011:Mailed notice(pm,) (Entered: 04/29/2011)	Not a Motion

Table 1: Sample motion entries illustrating some complexities of classifying motion types.

platform and what matters to us is the speed to information for our end users – and once the setup of our system is complete, speed from question to information for end users will be much faster and easier on a per-query basis. Thus, our system demonstrates a clear advantage given that the cost of defining a configuration can be amortized over every question answered.

On flexibility of exploration, we consider the state of things once the fee waiver question is answered. For the ad hoc approach, we have access to the information needed to answer the question at hand, but any subsequent question or filter amendment requires fresh code and additional work on the part of the data scientist. In contrast, once a configuration has been defined, our system supports running new types of analysis or perhaps the same analysis on different slices of the data (e.g., "How do fee waiver grant rates vary among judges in the Northern District of Illinois vs the Southern District of Illinois?") – without a data scientist in the loop.

Last, and we believe most important, is the barrier to entry. Regardless of time or effort, there will always be those who lack the technical skills or resources necessary to convert data into information. Since most are not data scientists and cannot afford to hire one, the ad hoc approach doesn't scale. In contrast, our system provides a path to minimize the barrier to entry by abstracting away these technical skills through a one-off, upfront setup. This decouples the data scientist from the exploration of data and by doing so democratizes access to the underlying information.

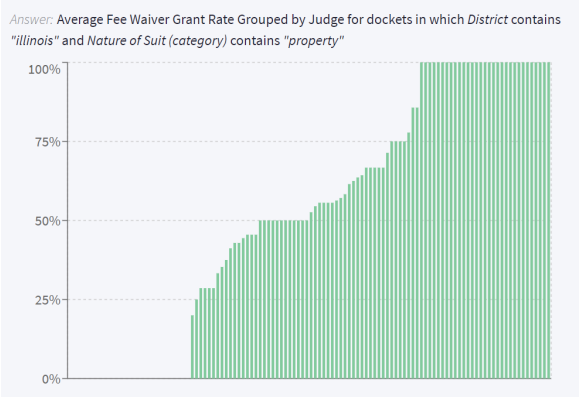

Figure 7: The system's response to the analysis statement "Average Fee Waiver Grant Rate Grouped by Judge" in a filtered data context. Each bar represents a given judge's fee waiver grant rate – judge names are available in rollover states (not depicted).

7 DISCUSSION

Though in its early stages, our work already demonstrates significant promise. User testing among legal scholars, attorneys and journalists in the U.S. confirms both the value and usability of our system. Beyond that, our approach has the potential to generalize well to a wide array of data sources, providing a new platform for the democratization of information across communities, sectors and geographies. We see broad opportunities for such an approach in the space of open government data, a domain rife with available datasets but with chronic challenges in terms of accessibility and use [18, 23]. This is a push towards the realization of the true goal of such initiatives: from code to content and from open data to open information.

To reach those goals, we must automate. We need systems that bridge the gap between questions and answers – routinely employing individual data scientists to serve that function simply doesn't scale. The ultimate objective is repositioning as much of the burden of that complexity on the machine as possible, and our work here is a step in that direction. And while our approach certainly won't eliminate the role of data scientists, we believe a significant amount of analysis can be standardized, systematized and automated, bringing access where previously there was none for lack of expertise or resources. And of note, where access does already exist, our approach has the potential to free data scientists from some of the repetitive "query generation" aspects of their roles, affording them more time to drive novel exploration. In some sense the approach detailed here could scale their expertise, allowing them to teach our platform about their datasets and then offload stakeholder questions to the system whenever possible.

That said, we see challenges in this approach to the U.S. court system, and anticipate them in scaling to new domains as well:

- **Issues of ethics and responsibility**: One such example is privacy. Court documents are rife with personally identifiable information, and reliably de-identifying documents at scale is a non-trivial problem. Further, the tension between de-identification and information completeness (say, for the sake of mapping to geographies) adds another complication. The use of highly regulated medical records data in research and machine learning provides a promising precedent [8, 25] for reference as we move forward.
- **Issues of information misuse**: Protecting against misuse of analysis, especially when the barrier of expertise to arriving at such analysis has been lowered, is a significant issue in our increasingly fraught information landscape. In the realm of law, the politicization of judicial decision making or the use of judicial analytics as a means of influencing future outcomes are both potential issues.
- **Issues of explainability and data quality**: Our scalable approach to data analysis adds a new layer of importance to the explainability of results and also runs the risk of obscuring incomplete or deficient data. To fully realize the promise of data science automation, additional research will be focused on ensuring our system can explain itself and handle issues of data quality gracefully and transparently.
- **Issues associated with novel analysis**: Inarguably, data scientists can flexibly address novel questions or analysis requirements on the fly, and while our platform's library of analytics will grow, there will continue to be question types it can't answer. In future work, we will expand our nascent plugin framework to support custom analysis and continuously grow the built in libraries.

8 FUTURE WORK

Going forward, various members of our team are pursuing in tandem the dual roadmap we laid out in the introduction.

One thread is aimed at making the raw data emitted from the U.S. court system increasingly machine-readable. This entails everything from the continued evolution of the ingestion pipeline (sourcing data from a wider variety of districts and tackling corner cases in the data) to improvements to the data already obtained through various forms of enrichment. In the near term, we intend to pursue entity disambiguation on parties and attorneys, as well as the creation of additional datasets to train language models for classification outside the scope of the motions described above (such that we can attempt to capture additional data points such as judicial rulings, charge severity, changes in representation, and various forms of case outcome).

The other thread is the work with the core platform itself. This will take a number of forms, including: 1) Expansions to the analytics capabilities and plugins (including the introduction of new response types and visualizations); 2) An evolution of the ontology configuration and support for ontology management through the user experience, allowing for user-driven updates as well as the introduction of new data sources; 3) Ontology-driven derived fields, providing support for adding new data points dynamically and introducing new possibilities for downstream explanations; 4) Support for localization such that the platform could be used by non-English speakers (of note, our ontology-driven approach means very little actual language is coded into the UI, making this an easier pursuit), opening up the possibility of legal documents and open data from other countries being made available through the platform; 5) UX improvements, including changes to analysis statement selection (with fuzzy semantic matching on colloquial terms against terms of art), support for more interactivity in visualizations, and additional explanations associated with analysis results; 6) Support for interactive machine learning by bringing the capabilities of our separate motion tagging application directly to the platform and augmenting them to cover both the extraction/creation of novel tagged datasets and in-platform model training/fine-tuning, validation and testing. This presents a significant opportunity for research in the space of making machine learning accessible to non-technical users.

While we believe deeply in the importance of bringing transparency to the U.S. court system and will continue the data work necessary to do so, we also see this data-information rift throughout the government and public sector in the United States and globally. Thus, we are excited by the prospect of platform improvements to support bringing a variety of new datasets to our application.

9 CONCLUSION

In this work we've detailed a novel platform and user experience to allow non-data scientists to drive exploration and analysis of data associated with the U.S. Court system. In support of that experience, we defined the process by which we ingested, extracted and structured the data from 270,000 case dockets. Given the results of usability testing presented in our evaluation, we believe we have early confirmation that this new natural language notebook approach marks a step in the direction of democratizing access to data analysis and could have significant impact not only in the space of the U.S. court system, but also more broadly across a variety of publicly available data. Subsequent work is already underway to further develop the capabilities, refine the UX mechanics, and stand up new components of the ecosystem. In tandem, the ingestion, structuring and enrichment of U.S. court records continues as we work towards a comprehensive database mirroring the federal court system.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Convergence Accelerator Program under grant no. 1937123 and grant no. 2033604.

REFERENCES

[1] Charlotte Alexander and Mohammed Javad Feizollahi. 2019. On Dragons, Caves, Teeth, and Claws: Legal Analytics and the Problem of Court Data Access. *Computational Legal Studies: The Promise and Challenge of Data-Driven Legal Research (Ryan Whalen, ed., Edward Elgar, 2019, Forthcoming)* (2019).

[2] Judie Attard, Fabrizio Orlandi, Simon Scerri, and Sören Auer. 2015. A systematic review of open government data initiatives. *Government Information Quarterly* 32, 4 (2015), 399–418.

[3] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies* 4, 3 (2009), 114–123.

[4] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[5] Karl Branting, Brandy Weiss, Bradford Brown, Craig Pfeifer, A Chakraborty, Lisa Ferro, M Pfaff, and A Yeh. 2019. Semi-supervised methods for explainable legal

prediction. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 22–31.
[6] Federal Judicial Center. 2011. Biographical directory of federal judges.
[7] Jonathan Crusoe, Anthony Simonofski, Antoine Clarinval, and Elisabeth Gebka. 2019. The impact of impediments on open government data use: insights from users. In *2019 13th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 1–12.
[8] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association* 24, 3 (2017), 596–606.
[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[10] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 3558–3567.
[11] Biralatei Fawei, Jeff Z Pan, Martin Kollingbaum, and Adam Z Wyner. 2018. A methodology for a criminal law and procedure ontology for legal question answering. In *Joint International Semantic Technology Conference*. Springer, 198–214.
[12] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building Watson: An overview of the DeepQA project. *AI magazine* 31, 3 (2010), 59–79.
[13] Anthony W Flores, Kristin Bechtel, and Christopher T Lowenkamp. 2016. False positives, false negatives, and false analyses: A rejoinder to machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *Fed. Probation* 80 (2016), 38.
[14] World Wide Web Foundation. 2018. *Open Data Barometer - Leaders Edition*. World Wide Web Foundation.
[15] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson H S Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2017. A Deep Semantic Natural Language Processing Platform.
[16] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*. 219–224.
[17] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. *arXiv preprint arXiv:1704.08760* (2017).
[18] Maxat Kassen. 2018. Adopting and managing open data: Stakeholder perspectives, challenges and policy recommendations. *Aslib Journal of Information Management* (2018).
[19] Muhammad Mahboob Khurshid, Nor Hidayati Zakaria, Ammar Rashid, and Muhammad Nouman Shafique. 2018. Examining the Factors of Open Government Data Usability From Academician's Perspective. *International Journal of Information Technology Project Management (IJITPM)* 9, 3 (2018), 72–85.
[20] Hyeonji Kim, Byeong-Hoon So, Wook-Shin Han, and Hongrae Lee. 2020. Natural language to SQL: Where are we today? *Proceedings of the VLDB Endowment* 13, 10 (2020), 1737–1750.
[21] Mi-Young Kim, Randy Goebel, and S Ken. 2015. COLIEE-2015: evaluation of legal question answering. In *Ninth International Workshop on Juris-informatics (JURISIN 2015)*.
[22] Mi-Young Kim, Ying Xu, and Randy Goebel. 2014. Legal question answering using ranking svm and syntactic/semantic similarity. In *JSAI International Symposium on Artificial Intelligence*. Springer, 244–258.
[23] Rashmi Krishnamurthy and Yukika Awazu. 2016. Liberating data for public value: The case of Data. gov. *International Journal of Information Management* 36, 4 (2016), 668–672.
[24] Karim R Lakhani, Robert D Austin, and Yumi Yi. 2002. *Data. gov*. Harvard Business School.
[25] Joffrey L Leevy, Taghi M Khoshgoftaar, and Flavio Villanustre. 2020. Survey on RNN and CRF models for de-identification of medical free text. *Journal of Big Data* 7, 1 (2020), 1–22.
[26] James R Lewis. 2018. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction* 34, 7 (2018), 577–590.
[27] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).
[28] Tomer Libal and Matteo Pascucci. 2019. Automated reasoning in normative detachment structures with ideal conditions. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 63–72.
[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[30] Lynn M LoPucki. 2001. Politics of Research Access to Federal Court Data. *Tex. L. Rev.* 80 (2001), 2161.
[31] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show me: Automatic presentation for visual analysis. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1137–1144.
[32] Peter W Martin. 2018. District Court Opinions That Remain Hidden Despite a Long-Standing Congressional Mandate of Transparency-the Result of Judicial Autonomy and Systemic Indiffernece. *Law Libr. J.* 110 (2018), 305.
[33] Gayle McElvain, George Sanchez, Sean Matthews, Don Teo, Filippo Pompili, and Tonya Custis. 2019. WestSearch Plus: A Non-factoid Question-Answering System for the Legal Domain. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1361–1364.
[34] Michael Bayer. [n.d.]. *SQLAlchemy*. https://www.sqlalchemy.org/
[35] Adam R Pah, David L Schwartz, Sarath Sanga, Zachary D Clopton, Peter DiCola, Rachel Davis Mersey, Charlotte S Alexander, Kristian J Hammond, and Luís A Nunes Amaral. 2020. How to build a more open justice system. *Science* 369, 6500 (2020), 134–136.
[36] Fernando Pérez and Brian E Granger. 2007. IPython: a system for interactive scientific computing. *Computing in science & engineering* 9, 3 (2007), 21–29.
[37] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. 2008. Linking data to ontologies. In *Journal on data semantics X*. Springer, 133–173.
[38] Min Ragan-Kelley, F Perez, B Granger, T Kluyver, P Ivanov, J Frederic, and M Bussonnier. 2014. The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication. *AGUFM* 2014 (2014), H44D–07.
[39] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2383–2392.
[40] Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual meeting of the association for Computational Linguistics*. 41–47.
[41] Mariano Rodriguez-Muro, Roman Kontchakov, and Michael Zakharyaschev. 2013. Ontology-based data access: Ontop of databases. In *International Semantic Web Conference*. Springer, 558–573.
[42] Diptikalyan Saha, Avrilia Floratou, Karthik Sankaranarayanan, Umar Farooq Minhas, Ashish R Mittal, and Fatma Özcan. 2016. ATHENA: an ontology-driven system for natural language querying over relational data stores. *Proceedings of the VLDB Endowment* 9, 12 (2016), 1209–1220.
[43] Jaydeep Sen, Chuan Lei, Abdul Quamar, Fatma Özcan, Vasilis Efthymiou, Ayushi Dalmia, Greg Stager, Ashish Mittal, Diptikalyan Saha, and Karthik Sankaranarayanan. 2020. ATHENA++ natural language querying for complex nested SQL queries. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2747–2759.
[44] Md Shamim Talukder, Liang Shen, Md Farid Hossain Talukder, and Yukun Bao. 2019. Determinants of user acceptance and use of open government data (OGD): An empirical investigation in Bangladesh. *Technology in Society* 56 (2019), 147–156.
[45] Songül Tolan, Marius Miron, Emilia Gómez, and Carlos Castillo. 2019. Why machine learning may lead to unfairness: Evidence from risk assessment for juvenile justice in catalonia. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 83–92.
[46] Ellen M Voorhees et al. 1999. The TREC-8 question answering track report. In *Trec*, Vol. 99. 77–82.
[47] Vishanth Weerakkody, Zahir Irani, Kawal Kapoor, Uthayasankar Sivarajah, and Yogesh K Dwivedi. 2017. Open data and its usability: an empirical view from the Citizen's perspective. *Information Systems Frontiers* 19, 2 (2017), 285–300.
[48] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* abs/1910.03771 (2019).
[49] Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics* 8 (2020), 183–198.
[50] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2017. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2648–2659.