PRESIDE: A Judge Entity Recognition and Disambiguation Model for US District Court Records

Adam R. Pah
Kellogg School of Management
Northwestern University
Evanston, IL
a-pah@kellogg.northwestern.edu

Christian J. Rozolis
Northwestern Institute on Complex Systems
Northwestern University
Evanston, IL
chris.rozolis@northwestern.edu

David L. Schwartz
Pritzker School of Law
Northwestern University
Chicago, IL
david.schwartz@law.northwestern.edu

Charlotte S. Alexander
Robinson College of Business
Georgia State University
Atlanta, GA
calexander@gsu.edu

SCALES OKN Consortium
Northwestern Institute on Complex Systems
Northwestern University
Evanston, IL
scales-okn@northwestern.edu

***Abstract*—The docket sheet of a court case contains a wealth of information about the progression of a case, the parties' and judge's decision-making along the way, and the case's ultimate outcome that can be used in analytical applications. However, the unstructured text of the docket sheet and the terse and variable phrasing of docket entries require the development of new models to identify key entities to enable analysis at a systematic level. We developed a judge entity recognition language model and disambiguation pipeline for US District Court records. Our model can robustly identify mentions of judicial entities in free text (∼99% F-1 Score) and outperforms general state-of-the-art language models by 13%. Our disambiguation pipeline is able to robustly identify both appointed and non-appointed judicial actors and correctly infer the type of appointment (∼99% precision). Lastly, we show with a case study on *in forma pauperis* decision-making that there is substantial error (∼30%) attributing decision outcomes to judicial actors if the free text of the docket is not used to make the identification and attribution.**

***Index Terms*—named entity recognition, disambiguation, court records, judicial entities**

I. Introduction

The court system is a pillar of US government, charged with resolving disputes, issuing punishment, and deciding how to interpret the law. A fundamental aspect of these functions is that courts should operate impartially, showing favor to neither the plaintiff nor defendant. While judges may view their work as akin to an umpire [1], managing cases based only on procedure and precedent, there is continued academic and industrial interest in analyzing court activity and decision-making at the judge level [2, 3, 4, 5, 6]. Indeed, studies have documented variations in judicial decision-making across both ideologically hot-button topics and seemingly neutral procedural issues, [7, 8, 9] warranting further research to help improve the function of the courts.

However, at the lowest level of the federal court system – the 94 district courts, which collectively handle over a quarter of a million civil and criminal cases per year – systematically attributing decisions to individual judges is not trivial for two main reasons.

The first has to do with how responsibilities within the courts are structured. There are two types of judges who primarily work in the federal district courts: district judges, who are Presidentially-nominated and Senate-confirmed judges appointed under Article III of the U.S. Constitution, and magistrate judges hired as fixed-term employees of the courts. Magistrate judges' responsibilities are not universally defined; instead the scope of their authority is decided on a district-by-district basis, within some broad statutory limits [10, 11, 12, 13]. This makes it nearly impossible to establish *a priori* rules governing when or how a magistrate judge could be involved in a case without consulting the data. The picture is complicated further by transfers of cases between and within district courts or the appearance of other Article III judicial actors (such as those from appeals and bankrupcty courts) when a case moves out of or into the district courts. Though court dockets have current judge fields, those fields do not reliably record which of the many possible judicial officers is making decisions in a case at any given moment in litigation. A judicial actor may recuse themselves, be promoted, retire, or die during the course of litigation, precipitating a change in the judge presiding over the case and complicating the attribution of actions and decisions.

The second reason is simple: while court records are public documents, the judiciary charges the public for electronic access on the Public Access to Court Electronic Records

This work was supported by the National Science Foundation Convergence Accelerator Program under grant nos. 1937123 and 2033604.

(PACER) site [14]. This has largely prevented the creation of a comprehensive, public dataset that systematically represents all districts and case types that would be necessary to develop a robust pipeline for named entity recognition and disambiguation of judges in court records.

There is prior work on training named entity recognition (NER) models on legal corpora, generally in countries where legal data is more freely available or on limited sets of court opinions in the United States [15, 16, 17, 18, 19]. These efforts have sought to develop models that recognize traditional entity classes (e.g., Person, Organization, Document) like generally available NER models, but are tuned to legal documents, which have a number of unique structural variations. While the performance of standard NER models and training weights has improved greatly with the advent of transformer based models [20, 21], it is still possible to improve on this performance for specific tasks with subject matter training data [22].

Disambiguation of named entities has not received as much research attention in US legal corpora at scale. The Federal Judicial Center (FJC) maintains an official biographical dataset for all appointed district judges since the 1700s [23], but does not maintain records on non-appointed judges. Efforts to identify and profile all magistrate judges that are employed in the federal courts is one of the most significant research contributions in the area of disambiguation [24, 25]; however, these efforts are built by hand and require continuous effort to maintain. There is a wealth of research in the area of author name disambiguation for scholarly works [26, 27], which includes the use of advanced methods like feature extraction from the published text [28, 29] and graph embedding [30, 31]. However, the appearance of a judge on a court docket sheet differs dramatically from the context of academic authors in scholarly works. The former appears by procedural random assignment, while the latter generally appears in relation to the subject matter of the work. This limits the utility of more advanced approaches as the only additional information present on a docket about a judge apart from their name is their title (e.g., District Judge, Magistrate Judge), which has no relation to the case matter.

Here we present a model to recognize and disambiguate judicial entities (PRESIDE) in the free text of docket sheets. We develop our model using federal district court cases filed in 2016 and find that it outperforms a general model in identifying judges and their titles. Using our disambiguation methods we are able to uniquely attribute judge mentions in free text to a single entity and even recognize judges who are not identified in external data sources. We further test the model on cases filed in a different year to verify its ability to generalize. Finally, we use decisions on *in forma pauperis* applications as an example to demonstrate the inaccuracy of an approach that uses only the docket sheet current judge field to identify the responsible actor for a decision within a case.

II. Data and Methods

A. Federal District Court Records

To establish the universe of cases that were filed in 2016, we queried for a list of all cases filed from '01-01-2016' to '12-31-2016' on PACER in all 94 US district courts. Using the query results, we then collected the docket sheet for each case from one of two sources: a free online archive called RECAP [32] (accounting for 135,867 docket sheets) and the paid PACER system (accounting for 205,399 docket sheets), producing a corpus of 341,086 dockets. We processed the HTML docket sheets with a custom parser [33] that produces a structured JSON data format.

Court docket sheets are a semi-structured data source and contain two primary sections of research interest, the header and the docket entry table. The header of a docket sheet is a structured layout that lists specific case-level information (e.g., current judge, parties, nature of suit) with a preceding key (e.g., 'Judge:', 'Plaintiff:', 'Nature of Suit:') that provides the data mapping. The docket entry table contains the chronological record of the case progression and is structured HTML, where each litigation event is logged as a table row with the event index, date, and entry text. Docket entries are entered into the record by the parties or court actors (e.g., judges, clerks) and are generated from a drop-down menu followed by a free text entry field.

B. Dataset Construction

To bootstrap the creation of our judge entity dataset we exploited the structured header of the docket sheets and the prevalent usage of honorific titles in the data. We extracted all judge names and titles from the structured header of each case, cleaned the title, and combined this name list with the FJC Article III judge biographical data to create a dictionary of both appointed and non-appointed judge names. Supplementing the FJC biographical dataset with header judge names increased the universe of known names 52%, from 3,762 to 5,730. We then enhanced the judge name dictionary by generating additional name format variations, like "Sara Ellis", "S. L. Ellis", and "Ellis, Sara". This expansion included generating name variants with abbreviated initials, name reversals, standalone surnames, suffix detection, and possessive detection, which increased the dictionary to 159,063 variants. This judge name dictionary is also combined with a dictionary of 40 different honorific variations.

To generate training data, we performed a brute force extraction of judge entities from the unstructured docket entry text based on the judge name dictionary. To combat the spurious identification of entity mentions in the docket entry text, e.g., 'Judge John Doe' triggering a search for just 'Doe' across all text, we developed a different logic rule for single-token (surname) than multi-token (full names) pattern extraction from the text. For single-token names to be extracted as a training sample we require that the single-token entity be directly preceded or followed by a honorific token ('Judge', 'Magistrate Judge', etc.). As an example, we would extract

Judge Doe as training data from the following two examples, but not the third example.

1) Order signed by District Judge Doe
2) Preliminary hearing held in chambers of Doe, District Judge. All parties advised...
3) Plaintiff Doe's motion for extension of time.

Our compiled dictionary of honorific variations and judge names resulted in over 10 million string combinations that would have to be considered during brute force text extraction. In order to scale the extraction process to our corpus of >300,000 docket sheets we used the FlashText package [34], which implements the trie-based FlashText algorithm [35]. As a first pass on a docket sheet we search for the appearance of any surname in the judge name corpus. If it is found then we apply the trie patterns for both judge name entities and combined honorific and name entities in a single pass. This process takes approximately 40 minutes (Windows 10 OS, with a 2.60 Ghz i7-10750H processor and 32 GB RAM) to consider 427 million tokens from 10.7 million docket entries and extracted 8.4 million tagged spans. In total, this extraction method generated 3.4 million docket entries with tagged spans and was split 80/14/6 into training, validation, and holdout datasets.

C. Model Training

We used the spaCy non-transformer architecture [21] for entity recognition with a custom tokenizer. The custom tokenizer was employed to standardize span detection even with punctuation at the end of names. Specifically, samples with suffixes where the punctuation is an optional inclusion (e.g. 'Jr' versus 'Jr.') create sentence collision issues with out of the box tokenizers. The architecture trained a custom NER pipeline on two entity labels: "honorific" and "judge". Training scoring was balanced between both labels, and no preference was given to precision or recall.

Due to limitations in memory, our training corpus was cut in half and streamed into the model. The model was trained on roughly 1.4 million docket entries containing approximately 3.4 million tagged spans. The remainder of the dataset was unchanged, with a validation set of 480,000 docket entries containing 1.2 million tagged spans. Training was conducted with a dropout rate of 0.1 and batch sizes that compounded from 100 to 1000 at a rate of 1.001. Patience was set such that 1,600 steps with no improvement would terminate training. Data was continuously streamed and the process was manually stopped at 70 hours. The best model in terms of the unweighted F-score across both entity tags was selected from hour 66. The model weights and accompanying code for PRESIDE are available online [36, 37].

D. Extraction

We extracted entities from all 94 US district courts' dockets with the trained PRESIDE model and applied a uniform adjustment and cleaning method. The adjustment methods were developed to handle traditionally non-English multi-token surnames. Our training data handled punctuation based tokenization issues such as hyphenation or infixed apostrophes (such as "O'Sullivan") and trained the model for these tokenization patterns. However, the model struggled with multi-token surnames that are whitespace bounded, leading to some entities being partially captured. For example magistrate judge "Susan Gregory Van Keulen" was frequently captured as "Susan Gregory Van". The adjustment method examines nearby extracted spans for an entity token and expands the detected span to include the full name if recognizable patterns like "`a` Van `B`" or "`a` De `b`" were found and the expanded pattern matched a longer extracted entity.

As a cleaning method to reduce the chance of false positives, i.e. incorrectly identifying a party or lawyer as a judge, we compared our list of extracted judge entities with known parties and lawyers appearing alongside them in the same case. We performed this comparison on extracted judge predictions that appeared in only one case in the corpus. This is a restrictive choice operating under the assumption that a judicial entity prediction is unlikely to be a false positive if it appears on multiple dockets. However, if a representation appears on only on one docket out of 300,000, and that appearance is a strong fuzzy match (>95%) to a listed party or lawyer in that case, then the entity is dropped from the pool of extracted judicial entities.

E. Disambiguation

We use string similarity methods to determine if predicted entities match with each other and should both map to the same single entity. We leverage custom implementations of token subset and overlap algorithms, as well as fuzzy matching based on the Levenshtein distance (*fuzzywuzzy* library [38]). Our custom implementations of token subset matching were designed to handle the usage of abbreviations in names, which court clerks commonly use when writing judge names in docket text. For example, in our algorithm "Matthew Kennelly" is a total subset of "Matthew F. Kennelly", and "M. F. Kennelly" is a total subset of an abbreviated form of the same name. These functions also handle basic name standardization and nicknames. For example, a clerk may inadvertently spell "Matt" as "Mat" or abbreviate "Edward" to "Ed". We built a dictionary of known nicknames and universal spelling mappings associated with formal names and leveraged this dictionary when evaluating token overlap. When all other tokens are considered equal, the nickname or universal spelling enabled a proper match (i.e. "Sara Lee Ellis" matches to "Sarah Lee Ellis"). The algorithm maintains a directed network of all predicted entities and creates links between two entities when they are predicted to match. This network is updated throughout the disambiguation process and at the end we resolve what is the official name to use for the single entity based on token length and frequency.

The disambiguation methods are applied in a four step sequential process (Figure 1) to disambiguate the entire pool of predicted judicial entities: (i) intra-case matching, (ii) intra-court matching, (iii) external data matching, and (iv) clustering matches.

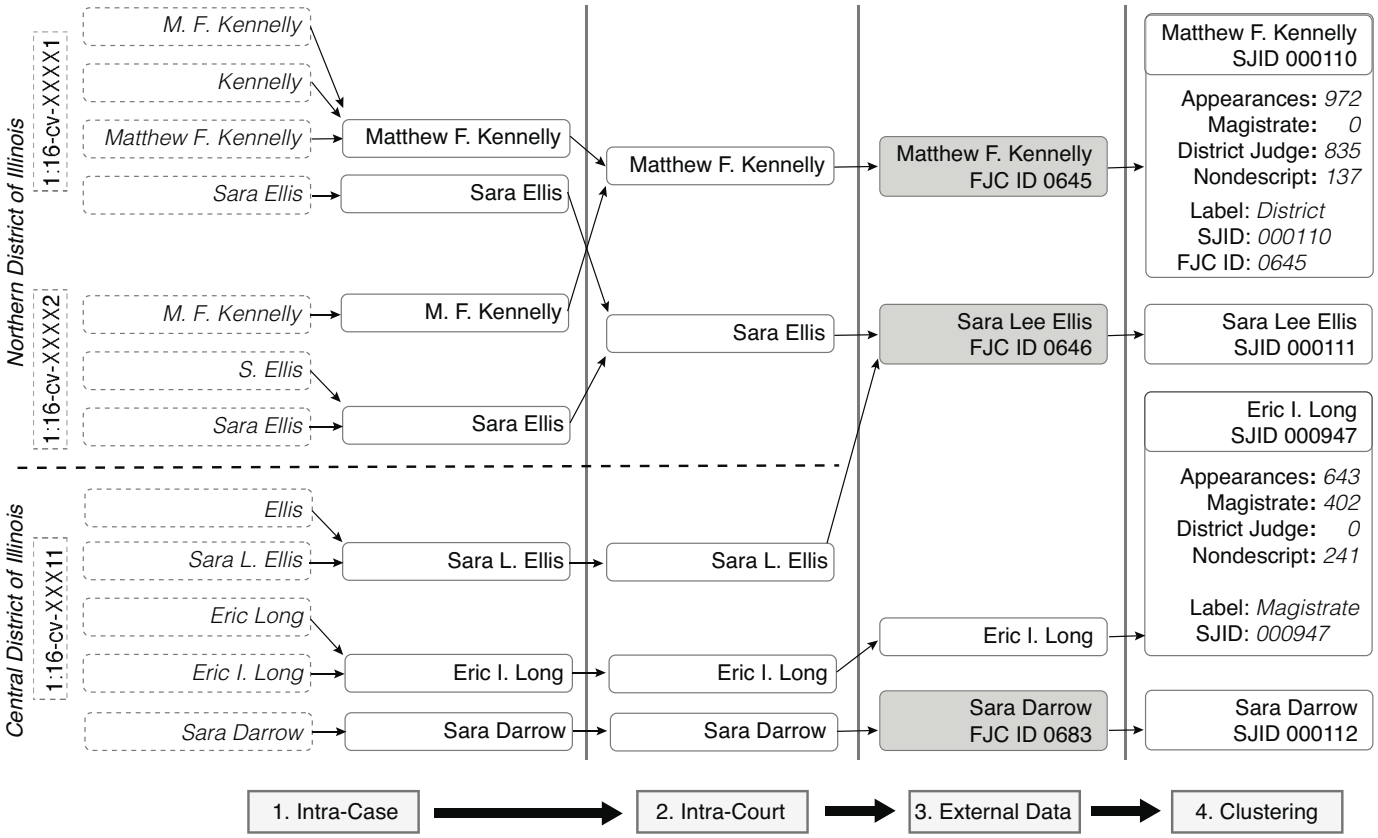

Fig. 1. Visual diagram of the four step disambigutation pipeline on entities extracted from docket text. In this example, we constructed three examples cases to demonstrate the flow of the disambiguation pipeline: two cases that originate in the U.S. District Court for the Northern District of Illinois and are reassigned to a new judge during the case (1:16-cv-XXXX1 and 1:16-cv-XXXX2) and one case that originates in the Northern District of Illinois and is transferred to the Central District of Illinois (1:16-cv-XXX11). Entities are first extracted from individual case docket sheets and matched within the case (1. Intra-Case). Entities are then matched between cases within the same court districts (2. intra-court) and then external datasets with official judge information (grey entity boxes) are added and matched (3. External Data). Finally, a clustering step is taken over the entire disambiguation network and titles are inferred for judges without a given title from an official external data source (4. Clustering).

We first disambiguate all predicted entities that appear within a single case since this is a constrained setting and repeated appearances, even with spelling variations, are likely to be the same single entity. Because fewer judges will be involved in a single case we loosen the thresholds on fuzzy matching to 90% if one spelling form appears 20 times more frequently than the other within the overall court. This helps to resolve infrequent typos and the selective inclusion of middle initials. Additionally, the restrictive case entity comparison matches standalone surnames to full names, even in the presence of spelling mistakes.

Next we pool the predicted judicial entities from all cases within a single district court together and repeat disambiguation. This allows us to further construct edges in the disambiguation network, without introducing the potential collisions of judges in different courts with similar names. After disambiguation at the court level, we add all entities from the FJC biographical dataset [23] as nodes in the network. With the addition of these official entity nodes we repeat the disambiguation process across the network. Finally, we reduce the pool of all predicted entities across all 94 districts with a network clustering approach. Throughout disambiguation a node is restricted to only one edge to another entity or itself, creating clusters of entity representations that all map to the same single entity once comparisons are done across the network.

Once the disambiguation network is fully generated the final entity representation is chosen for each entity cluster and the metadata is then generated in order to produce an inferred entity label as a district or magistrate judge. The metadata includes cumulative case appearances, district appearances, case header appearances, the total number of appearances alongside five honorific labels (District, Magistrate, Bankruptcy, General Judicial Actor, None), and, if available, the FJC biographical data unique identifier. While bankruptcy judges are not regular judicial actors in district courts, our model effectively extracts their honorifics and names whenever proceedings involve their interaction, and thus we account for them in entity labelling. Additionally, a subset of honorifics do not clarify a judge's true role, for instance "Honorable" could be referring to any number of judge types and "Justice" may refer to state or federal appeals judges when a case is transferred out of or into district courts. These honorifics are not discarded, but instead considered as a generalized label indicating the entity is a

judicial actor in some capacity. Entities that possess an official FJC biographical dataset unique identifier are automatically labeled as an Article III judge. For all other entities that do not contain a link to a node with a unique identifier from the FJC dataset, the proportion of honorific appearances helps determine their entity labels. If a judge appeared 70% of the time with a "Magistrate" label and 20% with a "General Judicial Actor" label (such as "Honorable"), and 10% of the time standalone, we infer this entity was in fact a Magistrate Judge.

III. Results

A. NER Model Performance

The trained NER model performs well, with an F-Score of 99.2% on prediction for both the honorific and judge entity tokens (Table I). This performance remains effectively the same in the validation data, removing concerns that the model is overfit to the training data to produce the high F-Score. As an additional test to make sure that the model is not overfit we built a dataset of ∼180,000 docket entries from cases that were filed in 2017 and assessed model performance (Table I). We find that its ability to detect honorifics is largely unchanged and the performance in detecting judge entity tokens is slightly decreased (99.2% to 98.7%).

TABLE I
Model performance on training, validation, and out of sample 2017 data (N of $1.4 * 10^6$, $2.05 * 10^5$, and $1.81 * 10^5$, respectively.)

	Training		**Validation**		**2017 Data**	
	Honorific	Judge	Honorific	Judge	Honorific	Judge
F-Score	99.2%	99.2%	99.3%	99.2%	99.1%	98.7%
Precision	98.6%	99.1%	98.7%	99.2%	98.4%	98.6%
Recall	99.9%	99.2%	99.9%	99.2%	99.8%	98.8%

As a comparison, we also performed predictions of judge entity tokens with the standard spaCy `en_core_web_lg` NER model on the 2017 data. To perform a comparison we consider any "PERSON" entity that the standard spaCy model predicts as a potential judge entity and do not penalize it for the identification of any additional "PERSON" entity predictions, which restricts us to only considering precision as a comparison metric. We find that the standard `en_core_web_lg` has a precision of 85.90%, which is nearly a 13 percentage point decrease in comparison to the custom trained PRESIDE model result (98.6%).

B. Disambiguation Performance

The disambiguation pipeline identified 2,549 distinct judge entities in the 2016 dockets, of which it captured 1,056 out of 1,124 Article III judges who were sitting on the bench between 2016 and 2018 according to the FJC biographical dataset. Of the 68 district judges missing from our results, 58 attained senior status before the start of 2016 and, while were technically commissioned district judges, were functionally retired. Upon further inspection, we did not find those 58 judges mentioned in our docket samples. Nine of the remaining ten judges are explained by undetectable nicknames (such as "Margaret Catharine Rogers" going by "M. Casey Rogers"), initialisms ("Paul Kinloch Holmes, III" going by "P.K. Holmes, III"), and suffix dropping ("Stephen Nathaniel Limbaugh, Jr." sans "Jr."). All of these variations were captured as their nicknames, but inadequately mapped back to their official identifier. The remaining judge did not appear on any docket entries in the entire corpus. This results in a precision of 99.2% for district judges when we consider only the judges who appeared in the dataset.

The model also identified an additional 285 Article III appointed judges who do not primarily preside in district courts (i.e. Appeals or the Supreme Court) and about 1,200 judges who were not in the FJC biographical dataset and were labeled with their appropriate non-Article III judge title. Of note, our model detected and labelled five judges in the Districts of Guam and the Northern Mariana Islands, despite not being in the FJC biographical dataset. Though judges in these territories are not appointed via Article III, the districts themselves enjoy near equivalent Article III jurisdiction, making their judicial entities relevant as a 'district judge' in research. We manually verified their appointments and found that the model extracted all five of these judges and correctly inferred "district judge" as their honorific.

To examine the disambiguation performance on magistrate judges we evaluate how many magistrate entities we capture that are in the publicly available BR/MAG database [25], which has been developed through hand-review of printed and online reference materials. The BR/MAG dataset lists 504 judges who would have been active magistrate judges between 2016 and 2018. Our disambiguation model uniquely identified all but five of those 504 judges. Upon further inspection, two of those judges are listed as part time and appeared only three times in all 300,000 dockets, two do not appear at all in docket text, and one was mislabeled by our model to a similarly named Article III judge ("Karen M. Williams" was mapped to District Judge "Karen J. Williams"). This results in a precision of 99.4% for magistrate judges when we consider only judges who appeared in the BR/MAG dataset.

C. Case study on decision attribution

Many analytical questions that involve judges require aggregating decision outcomes, like how often Judge *X* grants Motion type *Y* or how long discovery lasts in *Z* type of case with Judge *X*. By default the only named judge on a docket sheet is the judge who is currently presiding over the case and listed in the docket header. To demonstrate why entity recognition and disambiguation are critical for court analytics, we perform a case study that examines *in forma pauperis* (IFP) application decision attribution. An IFP application is a request by a plaintiff in a civil case for a filing fee waiver. The IFP application and the judge's grant or denial can determine whether the courthouse doors are effectively open or closed to indigent plaintiffs.

We identified all instances of IFP applications with a regular expression algorithm [8] in the 2016 case corpus and identified

which judge made the decision on the application using the PRESIDE model. We then assessed how frequently the deciding judge was the same as the judge who was listed in the current judge field in the header of the case docket (Figure 2). We find that attributing the decision to the judge listed in the header is only the correct inference about 70% of the time. Further, incorrect attributions are more prevalent with magistrate judges, where more than half of the magistrate judges who decided on IFP application outcomes were not currently presiding over the case and listed in the docket header.

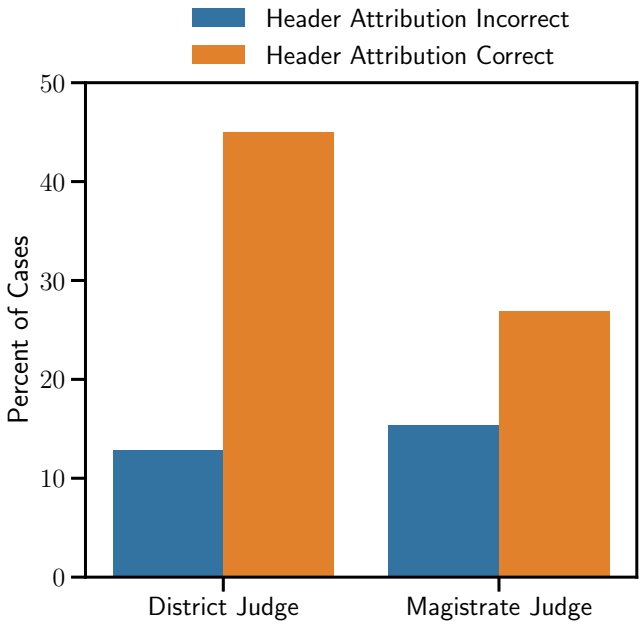

Fig. 2. Assessing whether the attribution of a decision on an *in forma pauperis* request to the judge listed in the header of a docket sheet is correct. Out of all *in forma pauperis applications* in 2016, 45,499 were decided by district judges and 33,238 were decided by magistrate judges. Decision attribution was more often correct when an Article III district judge issued the IFP decision (45% correct); however, there is still a high error in the base rate of decision attribution whether the deciding judge was a district or magistrate judge when the header is used for assignment (28% incorrect for both district and magistrate judges).

IV. Discussion

Here we present the PRESIDE model for the recognition and disambiguation of judicial entities in federal district docket sheet text. The model is able to robustly recognize and identify judge entities in unstructured text and appropriately classify them as district or magistrate judges. Importantly, it is able to correctly classify the title of entities even if they are not captured in external datasets.

There are limitations to the PRESIDE model related to generalization because of training data homogeneity and its ability to fully resolve judicial entities with common names. In addition, a known issue with NLP models trained on an English corpus is their ability to recognize and fully capture non-English names, which we observe in the PRESIDE results with Dutch surnames in particular. Without the creation of synthetic data to train the model on a larger lexicon of names our model will largely rely on the identification of honorific titles to identify judges with non-stereotypically English names. This results in the likely chance the model makes errors in detecting the full token span of their name (e.g., predicting ‘Judge van’ instead of the full span of ‘Judge van Gogh’).

An additional limitation stems from the prevalence of names such as ‘James’ or ‘Thomas’ that may double as first names and surnames. These names create a chance of false positive brute force tags if we erroneously tag ‘Judge James’ as an entity when ‘James’ is actually one of three name tokens such as ‘James J. Jameson’. These false positives occur when newer judges are appointed but we do not have the docket metadata to know their full name patterns, and thus their first name is erroneously tagged as a different judge’s surname. As a result, when testing the model, a limited subset of validation samples are incorrect and the model is penalized when it extracts the correct entities as all three tokens instead of the single ‘James’ we claimed was a surname. This issue was trivial for the 2016 dataset; however, we detected a twofold increase in false positives with the out of sample 2017 data because of multiple newly appointed judges with a first name of ‘James’.

A further limitation of the model is that it is only trained on federal court docket sheets. We have not developed or tested its ability to recognize judge entities in long-form prose, like in court opinions or party-filed documents, or in state court docket records. Increasing the variety of data sources that are a part of the training corpus and scaling it to handle English language court documents generally would dramatically increase its utility for researchers. The model might also be further extended to recognize judge entities even in non-English legal corpora, to the extent that honorifics are present as identifiers in those texts.

Overall, court dockets provide a wealth of data for downstream analyses about the function and activity of courts and greatly expand the types of research questions that can be asked and answered about the judiciary. However, they also require the development of new language models that can appropriately extract entities if the resulting analyses are to be trusted. Our model is one piece of the downstream analysis pipeline on judge actions or decisions that can aid researchers.

Acknowledgment

The authors acknowledge the many individuals and organizations who have given their time and helped to identify the key analytical components that are necessary for systematic analysis of court records, which is what motivates the present work. The SCALES OKN Consortium author list is available at http://www.scales-okn.org/research.

References

[1] Chief Justice Roberts Statement - Nomination Process. [Online]. Available: https://www.uscourts.

gov/educational-resources/educational-activities/chief-justice-roberts-statement-nomination-process

[2] S. Danziger, J. Levav, and L. Avnaim-Pesso, "Extraneous factors in judicial decisions," vol. 17, no. 108, pp. 6889–6892.

[3] J. D. Weinberg and L. B. Nielsen, "Examining empathy: Discrimination, experience, and judicial decisionmaking," vol. 8, no. 2, pp. 313–352.

[4] C. M. Smith, N. Goldrosen, M.-V. Ciocanel, R. Santorella, C. M. Topaz, and S. Sen, "Racial disparities in criminal sentencing vary considerably across federal judges."

[5] P. T. Kim, M. Schlanger, C. L. Boyd, and A. D. Martin, "How should we study district judge decision-making?" vol. 29, no. 1, pp. 083–112.

[6] J. Tashea. France bans publishing of judicial analytics and prompts criminal penalty. [Online]. Available: http://www.abajournal.com/news/article/france-bans-and-creates-criminal-penalty-for-judicial-analytics

[7] C. R. Sunstein, D. Schkade, and L. M. Ellman, "Ideological voting on federal courts of appeals: A preliminary investigation," vol. 90, no. 1, pp. 301–354.

[8] A. R. Pah, D. L. Schwartz, S. Sanga, Z. D. Clopton, P. DiCola, R. D. Mersey, C. S. Alexander, K. J. Hammond, and L. A. N. Amaral, "How to build a more open justice system," vol. 369, no. 6500, pp. 134–136.

[9] M. F. de Figueiredo, A. D. Lahav, and P. Siegelman, "The six-month list and the unintended consequences of judicial accountability," vol. 105, no. 2.

[10] R. L. Dessem, "The role of the federal magistrate judge in civil justice reform civil justice reform act," vol. 67, no. 4, pp. 799–842.

[11] D. A. Lee and T. E. Davis, "Nothing less than indispensable: The expansion of federal magistrate judge authority and utilization in the past quarter century symposium: Magistrate judges and the transformation of the federal judiciary," vol. 16, no. 3, pp. 845–948.

[12] P. G. McCabe, "A guide to the federal magistrate judge system," p. 66. [Online]. Available: https://perma.cc/6ZXJ-4JHT

[13] C. S. Alexander, N. Dahlberg, and A. M. Tucker, "The shadow judiciary," vol. 39. [Online]. Available: https://papers.ssrn.com/abstract=3426768

[14] A. Bielen, D. Kwan, D. Luetger, N. Maki, V. McFadden, and M. Myers, "Administrative office of the u.s. courts CM/ECF path analysis," p. 53. [Online]. Available: https://free.law/pdf/pacer-path-analysis.pdf

[15] I. Angelidis, I. Chalkidis, and M. Koubarakis, "Named entity recognition, linking and generation for greek legislation," in *Legal Knowledge and Information Systems*, M. Palmirani, Ed. IOS Press.

[16] P. H. Luz de Araujo, T. E. de Campos, R. R. R. de Oliveira, M. Stauffer, S. Couto, and P. Bermejo, "LeNER-br: A dataset for named entity recognition in brazilian legal text," in *Computational Processing of the Portuguese Language*, ser. Lecture Notes in Computer Science, A. Villavicencio, V. Moreira, A. Abad, H. Caseli, P. Gamallo, C. Ramisch, H. Gonçalo Oliveira, and G. H. Paetzold, Eds. Springer International Publishing, pp. 313–323.

[17] E. Leitner, G. Rehm, and J. Moreno-Schneider, "A dataset of german legal documents for named entity recognition." [Online]. Available: http://arxiv.org/abs/2003.13016

[18] ——, "Fine-grained named entity recognition in legal documents," in *Semantic Systems. The Power of AI and Knowledge Graphs*, ser. Lecture Notes in Computer Science, M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, and Y. Sure-Vetter, Eds. Springer International Publishing, pp. 272–287.

[19] J. Savelka and K. D. Ashley, "Detecting agent mentions in U.S. court decisions," in *Legal Knowledge and Information Systems*, A. Wyner and G. Casini, Eds. IOS Press.

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "HuggingFace's transformers: State-of-the-art natural language processing." [Online]. Available: http://arxiv.org/abs/1910.03771

[21] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength natural language processing in python." [Online]. Available: https://doi.org/10.5281/zenodo.1212303

[22] D. Alexander and A. P. de Vries, ""this research is funded by...": Named entity recognition of financial information in research papers," p. 10.

[23] History of the Federal Judiciary. [Online]. Available: https://www.fjc.gov/history/judges

[24] T. E. George and A. Yoon, "Article I judges in an article III world: The career path of magistrate judges."

[25] J. Ashley, S. New, and R. Owen. BR/MAG Judge Database. [Online]. Available: https://legaldatalab.law.virginia.edu/brmag-judges/

[26] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender, "A brief survey of automatic methods for author name disambiguation," vol. 41, no. 2, pp. 15–26.

[27] D. K. Sanyal, P. K. Bhowmick, and P. P. Das, "A review of author name disambiguation techniques for the PubMed bibliographic database," vol. 47, no. 2, pp. 227–254, publisher: SAGE Publications Ltd.

[28] J. M. B. Silva and F. Silva, "Feature extraction for the author name disambiguation problem in a bibliographic database," in *Proceedings of the Symposium on Applied Computing*, ser. SAC '17. Association for Computing Machinery, pp. 783–789.

[29] Q. Sun, H. Peng, J. Li, S. Wang, X. Dong, L. Zhao, P. S. Yu, and L. He, "Pairwise learning for name disambiguation in large-scale heterogeneous academic networks," in *2020 IEEE International Conference on Data Mining*

(ICDM), pp. 511–520, ISSN: 2374-8486.

[30] W. Zhang, Z. Yan, and Y. Zheng, "Author name disambiguation using graph node embedding method," in *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 410–415.

[31] L. Zhang and Z. Ban, "Author name disambiguation based on rule and graph model," in *Natural Language Processing and Chinese Computing*, ser. Lecture Notes in Computer Science, X. Zhu, M. Zhang, Y. Hong, and R. He, Eds. Springer International Publishing, pp. 617–628.

[32] The Free Law Project, "RECAP Archive [Data file]." [Online]. Available: https://www.courtlistener.com/recap/

[33] SCALES OKN, "scales-okn/PACER-tools: PACER scraping and parsing tools." [Online]. Available: https://doi.org/10.5281/zenodo.5056609

[34] V. Singh, "flashtext." [Online]. Available: https://pypi.org/project/flashtext/

[35] ——, "Replace or retrieve keywords in documents at scale." [Online]. Available: http://arxiv.org/abs/1711.00046

[36] SCALES OKN, "PRESIDE-lm NER model." [Online]. Available: https://doi.org/10.5281/zenodo.5562888

[37] ——, "scales-okn/research-materials: PRESIDE-lm: NER model and disambiguation pipeline." [Online]. Available: https://doi.org/10.5281/zenodo.5562932

[38] A. Cohen, "fuzzywuzzy." [Online]. Available: https://pypi.org/project/fuzzywuzzy/