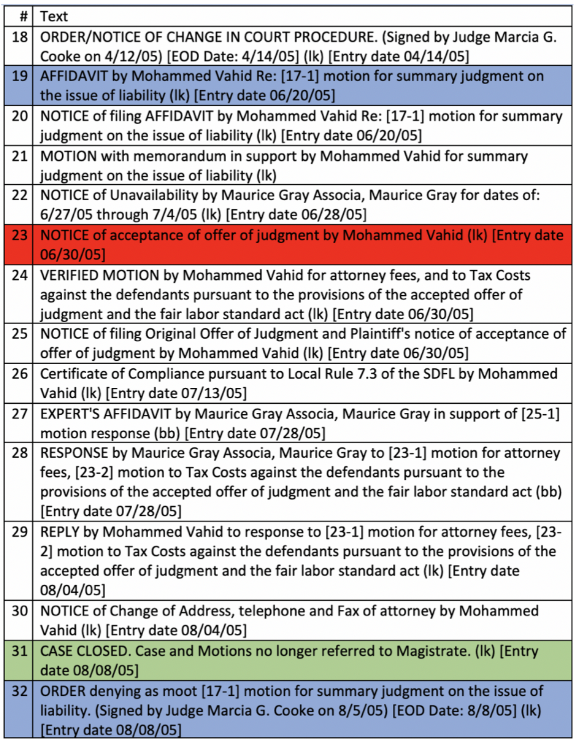
Litigation Analytics: Case outcomes extracted from US federal court dockets

Thomas Vacek*, **Ronald Teo***, **Dezhao Song***,
Conner Cowling* **and Frank Schilder***
*Thomson Reuters R&D
610 Opperman Drive
Eagan, MN 55123, USA
FName.LName@TR.com

Timothy Nugent†,
†Refinitiv
30 South Colonnade
Canary Wharf
London E145EP, UK
FName.LName@Refinitiv.com

Abstract

Dockets contain a wealth of information for planning a litigation strategy, but the information is locked up in semi-structured text. Manually deriving the outcomes for each party (e.g., settlement, verdict) would be very labor intensive. Having such information available for every past court case, however, would be very useful for developing a strategy because it potentially reveals tendencies and trends of judges and courts and the opposing counsel. We used Natural Language Processing (NLP) techniques and deep learning methods allowing us to scale the automatic analysis of millions of US federal court dockets. The automatically extracted information is fed into a Litigation Analytics tool that is used by lawyers to plan how they approach concrete litigations.



#	Text
18	ORDER/NOTICE OF CHANGE IN COURT PROCEDURE. (Signed by Judge Marcia G. Cooke on 4/12/05) [EOD Date: 4/14/05] (lk) [Entry date 04/14/05]
19	AFFIDAVIT by Mohammed Vahid Re: [17-1] motion for summary judgment on the issue of liability (lk) [Entry date 06/20/05]
20	NOTICE of filing AFFIDAVIT by Mohammed Vahid Re: [17-1] motion for summary judgment on the issue of liability (lk) [Entry date 06/20/05]
21	MOTION with memorandum in support by Mohammed Vahid for summary judgment on the issue of liability (lk)
22	NOTICE of Unavailability by Maurice Gray Associa, Maurice Gray for dates of: 6/27/05 through 7/4/05 (lk) [Entry date 06/28/05]
23	NOTICE of acceptance of offer of judgment by Mohammed Vahid (lk) [Entry date 06/30/05]
24	VERIFIED MOTION by Mohammed Vahid for attorney fees, and to Tax Costs against the defendants pursuant to the provisions of the accepted offer of judgment and the fair labor standard act (lk) [Entry date 06/30/05]
25	NOTICE of filing Original Offer of Judgment and Plaintiff's notice of acceptance of offer of judgment by Mohammed Vahid (lk) [Entry date 06/30/05]
26	Certificate of Compliance pursuant to Local Rule 7.3 of the SDFL by Mohammed Vahid (lk) [Entry date 07/13/05]
27	EXPERT'S AFFIDAVIT by Maurice Gray Associa, Maurice Gray in support of [25-1] motion response (bb) [Entry date 07/28/05]
28	RESPONSE by Maurice Gray Associa, Maurice Gray to [23-1] motion for attorney fees, [23-2] motion to Tax Costs against the defendants pursuant to the provisions of the accepted offer of judgment and the fair labor standard act (bb) [Entry date 07/28/05]
29	REPLY by Mohammed Vahid to response to [23-1] motion for attorney fees, [23-2] motion to Tax Costs against the defendants pursuant to the provisions of the accepted offer of judgment and the fair labor standard act (lk) [Entry date 08/04/05]
30	NOTICE of Change of Address, telephone and Fax of attorney by Mohammed Vahid (lk) [Entry date 08/04/05]
31	CASE CLOSED. Case and Motions no longer referred to Magistrate. (lk) [Entry date 08/08/05]
32	ORDER denying as moot [17-1] motion for summary judgment on the issue of liability. (Signed by Judge Marcia G. Cooke on 8/5/05) [EOD Date: 8/8/05] (lk) [Entry date 08/08/05]

Figure 1: The final case outcome is *Settled*, as entry [23] indicates. Entries [19, 31, 32] are candidate entries for potential outcomes, but ultimately incorrect because of [23].

1 Introduction

This paper focuses on the creation of an index of *case outcomes* for a given docket, which we define as the legal procedure which resolves the case. By the nature of this definition, a case may have only one outcome. The case outcome is distinguishable from the outcomes for each party in the case, as some parties may be dismissed or receive judgment prior to the end of the case.

Dockets of US Federal Court cases contain the description of the various steps that lead to the overall outcome (e.g., settlement, verdict). The language describing these steps (i.e., filing a motion, an order by a judge, a dismissal) are not standardized among the various courts. In addition, the outcome is derived from a sequence of docket entries and requires global information.

The current work explores how various machine learning approaches can be used in order to solve the problem of assigning an outcome to a given docket. We start with an SVM approach inspired by (Nallapati and Manning, 2008), who developed an approach determining one specific procedure type (i.e., summary judgment). This approach does not take into account any sequence information, whereas the other two deep learning based approaches we utilized do. The first approach uses a CNN-GRU architecture based on the TF-IDF vectors created for each docket entry. The second approach is a simplified hierarchical RNN approach called Nested GRU modeling the words of each docket entry and using those for modeling the sequence of all docket entries in an RNN sequence model. Finally, an ensemble method via a GBM combines the outputs of all three classifiers

Proceedings of the Natural Legal Language Processing Workshop 2019, pages 45–54
Minneapolis, Minesotta, June 7, 2019. ©2019 Association for Computational Linguistics

in order to determine the final outcome.

Results show that the deep learning approaches outperform the SVM based approach, but there is no statistically significant difference between the two deep learning methods and the system that combines all three approaches. The combined system also provided the input for an actual system deployed to customers who utilize the analytics derived from the 8 million US Federal dockets for their litigation planning.

The US Federal Court system, including district trial courts, bankruptcy courts, and appellate courts, all use an electronic records system that provides public access via a government computer system called PACER (Public Access to Court Electronic Records). The system maintains databases giving metadata associations of parties to cases, attorneys to parties, filing and closing dates of the cases, related groups of filings, and a high-level outcome of each case. Pacer also holds the official record of the case, which is all the documents pertaining to the case filed by the parties, their counsel, and the court. In addition to the documents themselves, there is a concise summary of each document written by the filer (and in recent times, based on a generated suggested text created by template), as well as the record of events for which no record document exists such as minor hearings. We believe that the intricacy and nuance of court procedures, as well as attorneys' perception of how to use procedure to their clients' advantage, has and will continue to cause the court system to be resistant to the adoption of fully digital workflows. Thus, dockets will contain significant unstructured data for the foreseeable future, and the task of defining, extracting, and indexing important litigation events falls to third parties and requires NLP techniques.

The metadata outcome information from PACER and the case outcome that we seek to index are indeed similar. There are two reasons why the metadata element is not sufficient by itself: First, it is frequently inaccurate, apparently because of differences in interpretation among the clerks of different courts. Second, a more specific taxonomy can be defined and extracted.

Applying machine learning and NLP capabilities to all federal dockets allowed us to collect outcomes for almost 8 million past dockets and also enables us to keep up with all newly closed dockets. In addition to extracting the outcome, the system is able to accurately determine a small percentage of cases that are likely to have an inaccurate extracted outcome, which should be reviewed by a human.

The case outcome task is distinguishable from other classic problem formulations in the NLP space. Classical approaches to document classification fail for several reasons: First, distributional assumptions in document classification are not valid because parties can spend a great deal of effort on issues that ultimately have no bearing on the outcome of the case. For example, a docket may contain minutes of many days at trial, but the judgment was granted as a matter of law, indicated by a few terse words in the docket. Second, negation is frequently critical. For example, there are a significant number of docket entries which say something like, "Settlement conference held. Case not settled." Finally, the problem requires extraction of a large classes of related facts. For example, a great deal of time and effort may pass before a judge issues a ruling on a motion. In addition, even though the case outcome problem is inherently sequential, dockets don't satisfy the Markov assumption, as events can have skipping dependencies.

Figure 1,[1] for example, describes a case that ends with a settlement even though the last two entries simply state that the case was closed (i.e., dismissed) and all pending motions including a motion for summary judgment were dismissed. Based only on these entries, the case would be dismissed, but entry [23] contains language that points to a settlement without actually mentioning settlement, but the acceptance of an offer of judgment indicates this kind of outcome.

This paper describes in more detail how the problem of detecting the outcome for a case can be solved and provides an overview of how we utilized machine learning including deep learning capabilities in combination with manual review. First, we describe the background of the case outcome problem and previous work in this area in section 2. Then, we describe the overall solution architecture and the underlying machine learning approaches used in section 3. Section 4 provides more details on evaluating the different approaches. Section 5 outlines the content of a demo of the live system and section 6 concludes.

[1] Some entries are abbreviated for readability.

2 Background

2.1 Previous work

There have been only a few approaches that have dealt with information extraction and classification tasks of legal court proceedings. Nallapati and Manning (Nallapati and Manning, 2008) are one of the few researchers who investigated machine learning approaches applied to classifying summary judgment motions only. Their findings indicated that rule-based approaches showed better results than a machine learning approach such as using a Support Vector Machine (SVM) (Hearst, 1998). Their results indicated that a classification approach using an SVM with uni/bi-grams would achieve only an overall F1-value of about 0.8, while a specified rule-based approach is able to achieve almost 0.9 F1-value. In contrast to our approach they only used a docket entry classification for each docket entry. That is a component our system also has, but we complement the result from this component with two Deep Learning approaches. Their focus was also only on one motion type, whereas we determine the outcome of multiple outcomes including summary judgment. More generally, however, they sought to extract only granted summary judgment motions while our approach determines an outcome for all parties.

A more recent approach by (Vacek and Schilder, 2017) looks at a wider range of outcomes and uses a sequence tagging technique (i.e., CRF (Lafferty et al., 2001)) for determining the final outcome of a case for a party. The current work is an improvement over this approach in terms of performance and the set of outcome types is larger.

Related work has been presented by (Branting, 2017) addressing the issue of detecting errors in filing motions as well as the matching between motions and orders. He reports a mean rank of 0.5-0.6 on this task.

There has also been work on predicting the outcome of a court case based on the written decision (Aletras et al., 2016; Sulea et al., 2017). Those approaches take the opinion text into account and predict the ruling by the court (e.g. French Supreme Court). We focus on the information described in the dockets only. (Luo et al., 2017) propose an attention based neural network model for predicting charges based on the fact description alone. They also show that the neural network model outperforms an SVM based approach, but they do not rely on dockets descriptions. (Xiao et al., 2018) describe a large-scale challenge of predicting the charge, the relevant law article and the penalty for more than 5 million legal cases collected from the Supreme People's Court of China. Similar to one of our approaches, they use a CNN based approach to predict the outcome of a case. Although all of these recent outcome prediction approaches use similar neural network approaches, they do not base their prediction on dockets nor do they deal with the sequence information of different court actions as they are encoded in the court filings. Instead they base their predictions on the fact section of a written opinion. The problem definition differs from ours and we also cast a much wider net because many litigations are dismissed early and no court opinions are actually crafted for those cases.

Other work has focussed on the legal action for other courts such as the Delaware Court of Chancery (Badawi and Chen, 2017) or debt relief extracted from Bankruptcy court filings (Dobbie and Song, 2015).

3 Case outcomes

The system produces outcomes according to a hierarchical taxonomy. The top-level outcomes are dismissal by motion, dismissal without a motion (includes agreed dismissals), settlement, default judgment, summary judgment, verdict, and docketed elsewhere (a catch-all for transfer, consolidation, and remand.). For this paper, we evaluate only this top-level taxonomy; a finer taxonomy is desirable for most use cases, and our observation is that this can be accomplished by downstream steps that specialize each class. The population distribution of outcomes is highly imbalanced in favor of dismissals and settlements, with verdicts representing a very small percentage of outcomes. This is illustrated in Figure 2.

The overall architecture of the system should be understood in terms of two abstract steps, where each is implemented redundantly. The first step is the conditional analysis of a particular docket entry; the intent is to determine what outcomes the given entry would be consistent with, ie. $P(\text{entry}|\text{outcome})$. Note that estimating $P(\text{outcome}|\text{entry})$ is usually futile because outcomes have contextual dependencies on many entries. The second high-level step makes inferences based on the conditional evidence identified

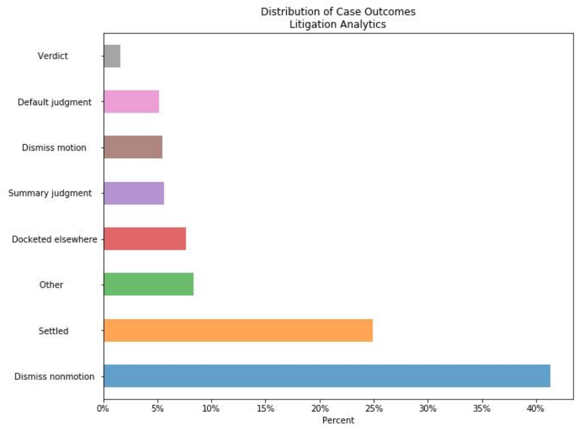


Figure 2: Federal cases are most likely settled or dismissed

in the first step. This could be interpreted as using machine learning to determine the normalization and interactions in applying the Bayes rule to determine $P(\text{outcome}|\text{entry sequence})$. The interactions are important; some outcomes such as jury verdicts are expected to have a trail of consistent evidence such as trial records and post-trial motions, while others like settlement can come rather out of the blue. In the implemented system, some components (such as the SVM classifier) can be neatly categorized as one of these two steps. However, the deep learning methods implement both steps simultaneously. The system architecture is depicted in Figure 3.

There is one component of the system that we have omitted from discussion. The system makes use of a large number of business rules, which all can be neatly categorized into the first abstract step. The rules have the form of a terms-and-connectors search, requiring the conjunction of various conditions within a certain number of words. We omit discussion for two reasons: First, they require expert knowledge that cannot be imparted with any brevity. Second, they are less useful in the prediction task than one might suppose. The likely explanation is that for rare events, a small mis-estimation of $P(X|Y)$ (i.e. a rule that is too broad) would lead to a wildly incorrect estimate of $P(Y|X)$. These rules are useful, however, as a post-check of a predicted outcome; at least one entry can be expected to have a rule match implying high probability given the predicted outcome.

The high-level components are the following, as indicated in Figure 3:

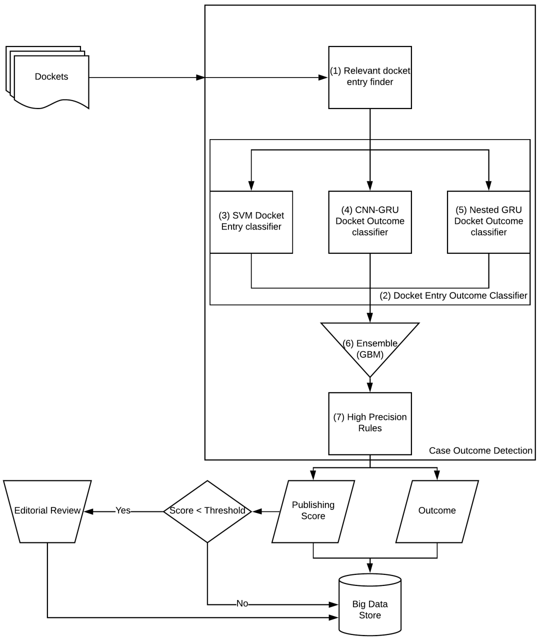

Dockets
(1) Relevant docket entry finder
(3) SVM Docket Entry classifier
(4) CNN-GRU Docket Outcome classifier
(5) Nested GRU Docket Outcome classifier
(2) Docket Entry Outcome Classifier
(6) Ensemble (GBM)
(7) High Precision Rules
Case Outcome Detection
Editorial Review
Yes
Score < Threshold
Publishing Score
Outcome
No
Big Data Store

Figure 3: The overall architecture of the Case Outcome system

1. A relevant docket entry finder. This module determines docket entries that are likely to have evidence as to the outcome of the docket. It is intended to have high recall, with little emphasis on precision.

2. A docket entry outcome classifier that predicts if a docket entry contains an outcome, and if so, which one. This classifier, similar to all the machine learning components, operates at the top level of the label taxonomy (see Figure 2). We developed three components to determine the final outcome of a docket.

3. An SVM was trained to provide an outcome per entry. Only the SVM approach uses the relevant docket entry.

4. A convolutional layer (CNN) followed by a Gated Recurrent Unit layer (GRU).

5. A nested recursive neural networks, one at the level of words in the docket entry and one at the level of docket entries.

6. A conventional gradient-boosted decision tree is used to predict the final outcome from features based on outcome SVM, CNN-GRU and Nested GRU classifier.

7. The next step applies human-written high-precision rules to sharpen the distinction be-

tween settlements and dismissals without a motion.

The final outcome is then localized (i.e., attached to a docket entry that gives evidence for it) using further rules that act on the docket closed date, the outcome entry classifier, and the outcome type classifier. Finally, the outcome is refined to add direction or open up the docketed elsewhere bucket using human-defined rules.

The output of the Docket Outcome component will provide the outcome as well as a confidence score. The confidence score is used for routing dockets either directly to the big data storage or to an editorial review system where cases and their outcomes are further reviewed by domain experts.

This paper will focus on the determination of a case outcome describing in more detail the components (3) SVM, (4) CNN-GRU, (5) Nested GRU, and (6) GBM.

3.1 Ensembling deep learning methods

In order to achieve high performance we ensembled various machine learning approaches including a SVM based approaches similar to (Nallapati and Manning, 2008). An SVM classifier focussing only on the outcome classification of the docket entry was trained in addition to two deep learning approaches.The first deep learning approach is a CNN-RNN combination that has a CNN (LeCun et al., 1998) layer followed by a GRU (Cho et al., 2014) layer before it is fed into a dense layer. The second deep learning approach is a nested RNN approach that first models each docket entry via an RNN (Schuster and Paliwal, 1997). Every docket entry is then the input for another RNN layer that models each docket entry as a step in the RNN model. The CNN-GRU and Nested-GRU model utilizes a custom trained word embeddings baed on Google's word2vec (Mikolov et al., 2013) to fine tune the embeddings to the docket corpus.

In the end, all scores retrieved from these models are used as features for a GBM (Friedman, 2000) model that combines the weights in order to determine the final outcome of the case.

SVM The purpose of this classifier is to predict the outcome associated with each entry of a docket. Note that this classifier does not take into account any interaction with previous outcomes or party information. The classifier used as input a feature vector for the top 3000 most frequent words/tokens, ordered by term frequency across the corpus words weighted by TF-IDF. A range of parameters were optimized including the maximum number of features (n=3000), the use of uni/bi-grams, lemmatization, removal of stop words, additive smoothing of IDF weights, sublinear term frequency scaling (i.e., $tf = 1 + log(tf)$), and regularizer choice. Some domain specific features were included such as binary encodings for the presence or absence of different party types, links etc, but these did not result in a significant performance improvement.

The classifier provides a robust prediction of whether an entry is consistent with one of the outcomes in scope. Often, however, the meaning of an entry can only be determined based on its context. A common example of this is when a lawsuit is dismissed because of technical but correctable deficiencies in the initial pleading. An explicit order dismissing the case may be followed shortly thereafter by a corrected complaint. Thus, the outcome of the case can only be determined by considering all of the entries in the docket, and more complex classifiers are required to determine the correct outcome of the docket as a whole. Hence, we incorporated two further deep learning models.

CNN-GRU In addition to predicting the associated outcome of each docket entry, we adopted a neural network based approach to predicting the outcome of one entire docket similar to (Wang et al., 2016). We first designed and experimented with a few Convolutional Neural Network (CNN) based approaches by adopting different architectures, e.g., single-input and multi-input (Yang and Ramanan, 2015) networks. In our single-input model, the input is vectorized features (e.g., word embeddings or TF-IDF scores) and we predict the outcome for each docket. When using word embeddings (our embedding has 300 dimensions), we concatenate all the docket entries and use the last 150 words (i.e., the input is a tensor with shape 150 * 300), since descriptions towards the end of a docket may be more indicative of the outcome. When using TF-IDF scores as the input, we first build a 322-word vocabulary by selecting words whose document frequency is above 20. Then, we use the last 150 docket entries and turn each docket entry into a 322-dimension vector (i.e., the input is a tensor with shape 150 * 322).

In our model, the input is first sent to a Convolutional layer and then a MaxPooling layer. Then, the intermediate results are sent to an GRU layer (a type of recurrent layers). At the bottom of our architecture, we use a Dense layer with softmax activation to obtain the final prediction. Differently, in our multi-input network, in addition to using the vectors (e.g., TF-IDF scores or word embeddings), we also utilize the output of the SVM classifier (i.e., the probabilities that a docket entry has certain outcomes) as additional input. By trying out different ways of combining these two inputs (e.g., combining them at the beginning of the network or running each input through a similar network and then combining them later), we found out that our multi-input model generally performs better than our single-input model.

Nested GRU The Nested GRU (cf. Figure 4) addressed the need to incorporate information from the entire sequence, as indicated by the docket excerpt in Figure 1. Compared to the SVM model, the Nested GRU is an end-to-end model that takes a matrix of shape (batch_size, MAX_ENTRIES, MAX_WORDS, EMBEDDING_SIZE) as input and produces a single outcome for the docket, which enables the network to learn directly from the docket outcome rather than the entry outcome that lacks all global information to determine the docket outcome. The Nested GRU utilizes the same idea of progressive encoding used by Hierarchical Attention Networks (HAN) as described by (Yang et al., 2016) but does not use an attention network to perform a ”soft-search.”

Using a hierarchical approach, we can preserve the natural structure of the docket (e.g., each entry consist of words and each docket consist of entries) for encoding. We summarize the ”meaning” of each entry by encoding the sequence of words (e.g. ”order granting motion,” ”consent order”, ”consent judgment”) and propagate the encoding to the corresponding sequence to the next hierarchy consisting of GRU cells. This ”docket entry level” hierarchy encodes the ”meaning” of the entire docket and propagate the encoding to a fully-connected network with a softmax activation to obtain the classification of the entire docket.

GBM The system mediates the ensemble of predictors by means of a gradient boosted decision tree. The model takes an input of roughly 100 expert-designed features. For the ensemble predictors that solve the problem directly (deep learning models), obvious features arise, for instance, from the softmax probability estimate for each outcome type. For ensemble predictors that have scope limited to a single docket entry (SVM and low-level patterns written for the manual-review flagging business rules discussed below), features are created from aggregations of the information extracted from each entry. The expert craft lies in how these aggregations are defined. Moreover, PACER provides limited metadata about the outcome of the case, so these factors can also be used to define various aggregations.

We treat the features generated by the SVM system (e.g., outcome probabilities) feeding into the GBM as the base system configuration. The experiments described in the next section will report on different combinations of the base system with the 2 deep learning approaches in order to keep the number of system combinations manageable.

3.2 Manual review

The output of party outcome detection may be flagged for manual review based on the prediction confidence scores output by the classifier and the numerous business rules mentioned previously. If an outcome is flagged, the docket is routed to an editorial tool that allows legal domain experts to review the extracted data. The automatically published and the reviewed dockets and their extracted motions/orders and outcomes are stored in a big data store.

4 Evaluation

4.1 Data

We sampled and acquired outcome annotations of 10,602 dockets. For each docket, one human annotator examined the entire docket and determined the outcome and associated docket entry for every party in the case. The case outcome, as defined for this task, is the last such outcome (for a party) in the case, assuming the case has been closed. A pre-study determined that overall inter-annotator agreement is relatively high with a kappa > 0.8. We used a fixed set of approximately 80% of the human annotated dockets for training and validation, and held out the remainder for testing.

The dataset used in this work is proprietary in accordance with the sponsor's requirements; however, an equivalent dataset could be acquired by any researcher inexpensively. Unannotated dockets can be obtained for free through the Free Law Project.[2] Moreover, courts can waive PACER fees

[2] `http://free.law`, also some of this collection has

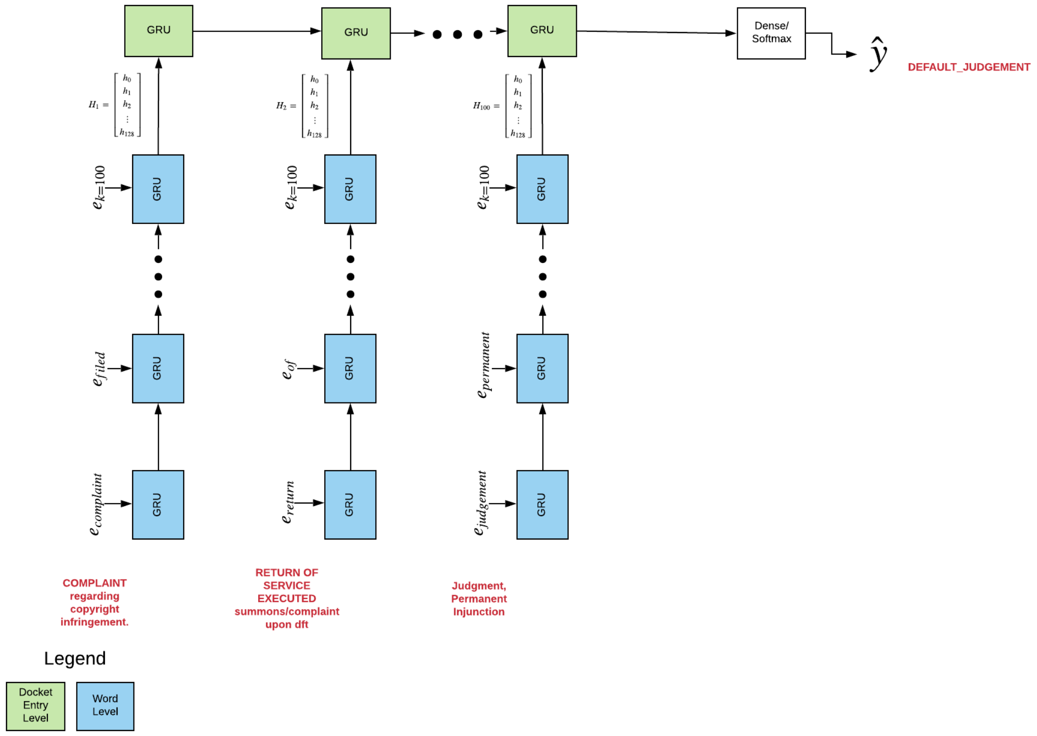

GRU
GRU
GRU
Dense/ Softmax
$\hat{y}$
DEFAULT_JUDGEMENT
$H_1 = [h_0, h_1, h_2, \vdots, h_{128}]$
$H_2 = [h_0, h_1, h_2, \vdots, h_{128}]$
$H_{100} = [h_0, h_1, h_2, \vdots, h_{128}]$
$e_{k=100}$
e_{filed}
$e_{complaint}$
e_{of}
e_{return}
$e_{permanent}$
$e_{judgement}$
GRU
COMPLAINT regarding copyright infringement.
RETURN OF SERVICE EXECUTED summons/complaint upon dft
Judgment, Permanent Injunction
Legend
Docket Entry Level
Word Level

Figure 4: The sequence of words for each docket entries are nested into another layer of RNN modeling the sequence of entries

for research in the public interest.[3] Outcomes for these cases can be scraped from the Pacer Summary Report for $0.10 per case, or obtained for free with a fee waiver.

4.2 Experiments

We evaluated the overall system's performance by comparing how much the three different ML approaches contribute to the overall performance. Table 1 shows how the singular approaches behave. The nested GRU approach has the best overall performance and almost all individual outcomes are detected with higher F1-scores by this method (except for docketed elsewhere). The CNN-GRU methods shows better or equal results for each outcome compared to the results achieved by the SVM method we deployed.

We tested whether the performance of the respective system combinations are statistically different. We used the McNemar's test for identifying whether a machine learning classifier outperforms another one following the study by (Dietterich, 1998).

Table 3 indicates that the results created by the CNN-GRU and the Nested GRU approaches are significantly different from the baseline system that only uses SVM features for the GBM classification. The combined approach utilizing both CNN-GRU and Nested GRU features in addition to the SVM features outperforms the baseline system as well, but the performances of the CNN-GRU and Nested GRU looked at individually are not significantly different as indicated by the p-values obtained from the McNemar's test. There is also no statistically significant difference between the results of the combined approach and each of the results of the two deep learning approaches.

5 Demo

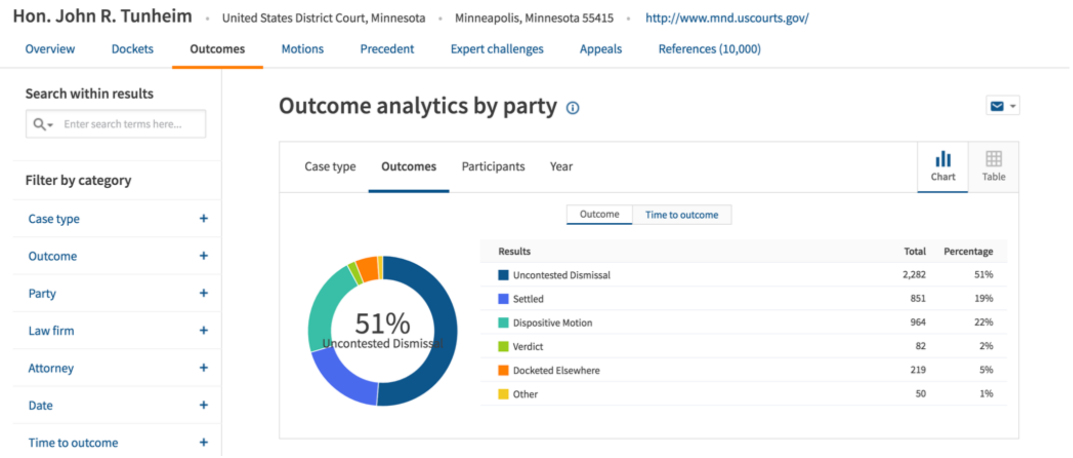
The outcome detection system described in this paper has been implemented in order to provide the case outcome information for all US federal judges and feeds into a Litigation Analytics program that allows lawyers to determine their litigation strategy. Lawyers can, for example, explore how often judges have ruled on a case resulting in a settlement, dismissal or trial. In addition, the

been uploaded to http://archive.org

[3] See Discretionary Fee Exemptions in the Pacer Fee Schedule at https://www.pacer.gov/documents/epa_feesched.pdf

	SVM			CNN-GRU			Nested GRU		
Outcome	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
DEFAULT JDG	0.79	0.85	0.81	0.85	0.85	0.85	0.92	0.92	0.92
DISMISS MOTION	0.90	0.81	0.85	0.94	0.83	0.88	0.90	0.89	0.90
DISMMISS	0.92	0.91	0.91	0.94	0.91	0.92	0.94	0.94	0.94
DOCKETED E.	0.88	0.85	0.86	0.88	0.85	0.86	0.90	0.79	0.84
OTHER	0.89	0.96	0.92	0.92	0.98	0.95	0.98	0.94	0.96
SETTLED	0.89	0.91	0.90	0.89	0.95	0.92	0.92	0.94	0.93
SUM JDG	0.87	0.82	0.84	0.89	0.89	0.89	0.89	0.89	0.89
VERDICT	0.69	0.71	0.70	0.79	0.65	0.71	0.72	0.76	0.74
Micro avg	0.88	0.88	0.88	0.90	0.90	0.90	0.91	0.91	0.91
Macro avg	0.85	0.85	0.85	0.88	0.86	0.87	0.90	0.88	0.89
Weighted avg	0.88	0.88	0.88	0.90	0.90	0.90	0.91	0.91	0.91

Table 1: Single approaches and respective performances



Hon. John R. Tunheim • United States District Court, Minnesota • Minneapolis, Minnesota 55415 • http://www.mnd.uscourts.gov/

Outcome analytics by party

Results	Total	Percentage
🟦 Uncontested Dismissal	2,282	51%
🟦 Settled	851	19%
🟩 Dispositive Motion	964	22%
🟩 Verdict	82	2%
🟧 Docketed Elsewhere	219	5%
🟨 Other	50	1%

Figure 5: A screenshot of the Litigation Analytics system

Outcome	Prec.	Recall	F1
DEFAULT JDG	0.92	0.85	0.88
DISMISS MOTION	0.94	0.91	0.92
DISMMISS	0.93	0.93	0.93
DOCKETED E.	0.90	0.82	0.86
OTHER	0.96	0.96	0.96
SETTLED	0.92	0.94	0.93
SUM JDG	0.90	0.90	0.90
VERDICT	0.78	0.74	0.76
Micro avg	0.92	0.92	0.92
Macro avg	0.91	0.88	0.89
Weighted avg	0.92	0.92	0.92

Table 2: Results of all approaches combined

	CNN-GRU	Nested	All
SVM	**0.013**	**0.002**	**0.000**
CNN-GRU		0.256	0.071
Nested			0.549

Table 3: P-values for the McNemar's test for system combinations

user can determine how long it takes for a particular judge to reach a settlement etc.

Figure 5 indicates what the distribution of different high level outcomes is for the federal Judge John Tunheim. The user can then further explore the outcomes and identify more fine-grained outcomes. Furthermore, they can select further categories such as law firms, parties, attorneys or simple date restrictions in order to research similar cases that would inform them regarding their best strategy for their clients.

6 Conclusion

We have described how to extract the case outcome from the docket entry summaries, and provided justification for why this task is important. While the system is very accurate for the scope of the defined task, the future challenges almost all revolve around making sure that the metadata events in this large-scale case catalog are relevant, accurate, unbiased, and useful. For example, it is critical to ensure that the mistakes of the system are unbiased as to the selection criteria that a user might wish to study. We use audits, user feedback, and specific queries to investigate the accuracy of outcomes.

More generally, determining what legal events in a case should be detected and indexed requires considerable collaboration between legal experts and NLP experts. The definition of ”case outcome” as we have used it here was the result of a great deal of investigation and consultation. There are many additional events that could be extracted.

Finally, the system described here relies entirely on the concise summaries of the events of the case described in the docket entries, while ignoring the official record documents themselves. This is due, in part, to the difficulty in large-scale access to those documents. Access to the records of the case would open the possibility to track *issue outcomes*, or the success of failure of each claim in a case instead of the case as a whole.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. *PeerJ Computer Science*, 2:e93.

Adam B Badawi and Daniel L Chen. 2017. The Shareholder Wealth Effects of Delaware Litigation. *American Law and Economics Review*, 19(2):287–326.

Luther Karl Branting. 2017. Automating judicial document analysis. In *Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Texts co-located with the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017), London, UK, June 16, 2017.*

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

Will Dobbie and Jae Song. 2015. Debt relief and debtor outcomes: Measuring the effects of consumer bankruptcy protection. *American Economic Review*, 105(3):1272–1311.

Jerome H. Friedman. 2000. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

Marti A. Hearst. 1998. Trends & controversies: Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to predict charges for criminal cases with legal basis. *CoRR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Ramesh Nallapati and Christopher D Manning. 2008. Legal docket-entry classification: Where machine learning stumbles. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 438–446, Honolulu, Hawaii. Association for Computational Linguistics, Association for Computational Linguistics.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45(11):2673–2681.

Octavia-Maria Sulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P. Dinu, and Josef van Genabith. 2017. Exploring the use of text classification in the legal domain. *CoRR*, abs/1710.09306.

Tom Vacek and Frank Schilder. 2017. A sequence approach to case outcome detection. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 209–215. ACM.

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2428–2437, Osaka, Japan. The COLING 2016 Organizing Committee.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018. CAIL2018: A large-scale legal dataset for judgment prediction. *CoRR*, abs/1807.02478.

Songfan Yang and Deva Ramanan. 2015. Multi-scale recognition with dag-cnns. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1215–1223.

Zichao Yang, Diyi Yang, Chris Dyer, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL 2016*, pages 1480–1489.