PHILOSOPHICAL TRANSACTIONS A

royalsocietypublishing.org/journal/rsta

Research

Cite this article: Gray MA, Savelka J, Oliver WM, Ashley KD. 2024 Empirical legal analysis simplified: reducing complexity through automatic identification and evaluation of legally relevant factors. *Phil. Trans. R. Soc. A* **382**: 20230155. https://doi.org/10.1098/rsta.2023.0155

Received: 1 June 2023
Accepted: 19 December 2023

One contribution of 15 to a theme issue 'A complexity science approach to law and governance'.

Subject Areas:
artificial intelligence

Keywords:
machine learning, interpretability, annotation, large language models, generative classification

Author for correspondence:
Morgan A. Gray
e-mail: mag454@pitt.edu

Empirical legal analysis simplified: reducing complexity through automatic identification and evaluation of legally relevant factors

Morgan A. Gray[1], Jaromir Savelka[3], Wesley M. Oliver[4] and Kevin D. Ashley[1,2]

[1]Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA, USA
[2]School of Law, University of Pittsburgh, Pittsburgh, PA, USA
[3]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
[4]Thomas R. Kline School of Law, Duquesne University, Pittsburgh, PA, USA

MAG, 0000-0002-3800-2103; JS, 0000-0002-3674-5456

This paper investigates the potential for reducing the complexity of AI and Law and empirical legal studies projects through a novel annotation methodology that relies on GPT Family Models to assist human annotators. Improving the speed, cost and quality of annotation could greatly benefit such projects. In modelling types of legal claims, researchers in the fields of empirical legal studies and AI and Law have long relied on manually annotating factors in case texts. To demonstrate our methodology, we employ cases and factors regarding whether a police officer has constitutional authority to detain a motorist on the basis of the officer's suspicion that the motorist is trafficking drugs. Our results demonstrate how recent advances in text analytics can *reduce the burden* of identifying factors in large numbers of cases and improve machine learning models' predictions of case outcomes.

This article is part of the theme issue 'A complexity science approach to law and governance'.

THE ROYAL SOCIETY
PUBLISHING

1. Introduction

The need for annotated data is a universal concern for researchers in natural language processing (NLP), machine learning (ML) and beyond. In modelling types of legal claims, researchers in the fields of empirical legal studies and AI and Law have long relied on manually annotating factors in case texts, but such annotation is burdensome, costly and arduous. Methods that could reduce the need for manual annotation, such as fine-tuning a large language model (LLM) to classify sentences, still require annotating training sets of data. Methodologies that could alleviate the burden of manual annotation are therefore critical. We present a methodology employing generative LLMs from the GPT Family of Models (GPT-FMs) and zero-shot prompting to reduce the burden of labelling legally relevant sentences. To illustrate an example of these methods in context, we briefly describe the legal domain used to assess this framework.

Since the 1980s, law enforcement agencies in the USA have used ordinary traffic stops, particularly on interstate highways, to interdict the flow of illegal drugs from source cities to communities throughout the country [1]. Officers have the power to stop any driver who violates any of the myriad regulations governing vehicles and their operation. An officer is then permitted to detain the vehicle for a sniff by a drug dog if suspicious circumstances are observed, suggesting drug possession or trafficking [2]. These circumstances must constitute ‘reasonable suspicion’ to believe that drugs are present. This legal standard requires that officers point to specific observations that caused them to believe drug trafficking was afoot. These observations often rely on legal factors indicating drug trafficking. Descriptions of these factors and a court’s determination that they are sufficient to find reasonable suspicion can be found in court opinions where a detained motorist challenges the detention.

In prior work, we have identified these factors [3] and shown their utility in predicting the outcomes of cases challenging the presence of reasonable suspicion [4] and in providing empirical insights about the legal domain. By reducing the burden of annotation required to automatically identify the factors courts rely on in drug interdiction auto stop (DIAS) cases, we are developing a pipeline to assess courts’ findings that particular sets of circumstances satisfy the reasonable suspicion standard.

We frame the use-case of our methodology in §3. We describe the steps we have undertaken in a pilot study to assess the automatic identification of legally relevant factors and to analyse them empirically using interpretable ML models. In §4, we discuss how to leverage the zero-shot capability of GPT-FMs by prompting them with instructions that would normally be given to human annotators to identify relevant factors in sentences and to provide suggested labels to human annotators. We provide evidence that GPT-FMs’ use in sentence annotation may significantly reduce the need for manual efforts, due to its proficiency in accurately labelling sentences, identifying various factors in each case and maintaining a reasonable level of factual consistency with few instances of hallucination. Specifically, we demonstrate that, by employing our methodology, GPT-4 and GPT-4-Turbo can identify factors in sentences with accuracies of 91% and 89%, respectively.

2. Related work

In the fields of NLP and AI and Law, researchers have applied automated techniques to classify texts of contracts (e.g. in terms of 41 categories involving general information, restrictive covenants, revenue risks [5], clause fairness [6], statutes by topics [6,7] and legal cases [8]).

The work on classifying legal cases has focused on classifying them by:

(i) argument organizational categories including fact, issue, rule/law/holding, analysis and conclusion/opinion/answer [9],
(ii) judicial subtasks in connection with predicting judgments of civil law cases [10],
(iii) whether the case overrules a previous one or by the type of procedural motion addressed [6],

(iv) types of applicable legal claims [7],
(v) applicable civil code articles [11] and statutory elements [10],
(vi) domain concepts [9],
(vii) relevance to a query case [12],
(viii) a winning or losing factual scenario for particular types of claims [7],
(ix) factual features that strengthen or weaken a claim [13].

Items (i) and (vi) through (ix) are of special interest in empirical legal studies and, in particular, to the use of statistical methods such as ML algorithms, 'to study legal doctrine through the use of fact-pattern analysis' [14]. Our project contributes to this work in empirical legal studies by automatically identifying factors in legal cases and applying ML to analyse judicial decisions of such cases. The concept of a factor, namely 'a consideration a decision maker must or may take into account to determine an outcome' is 'a foundational and ubiquitous concept ... ' in law [15, p. 2, 3]. Legal factors 'can be prescribed in a statute or regulation, or created by courts', and are employed in diverse areas of law including determining consumer confusion as to the source of goods in trademark infringement, determining works made for hire and copyright fair use [16, p. 1584f], [17,18], assessing spousal support, determining violations of the right to a speedy trial [15, p. 2f], and others.

In empirical legal studies, researchers have focused on multi-factor tests in legal domains, often with the aim of applying ML to determine which factors are most important. For example, Beebe identified 192 federal district court opinions from a 5-year period that involved preliminary injunctions or bench trials, and employed a multi-factor test for the likelihood of consumer confusion as to the source of goods in trademark infringement cases [16, p. 1584]. He manually classified them recording whether a factor was 'found to favor [a/no] likelihood of confusion or otherwise not to favor [no/a] likelihood of confusion'. Then, he applied simple classification trees [16, p. 1603], revealing that 'judges determine the test outcome based on a limited number of core factors and then adjust the rest of the factor outcomes to accord with that result' [16, p. 1587]. Similarly, Shao *et al.* applied decision trees with factors in child custody law to identify the three most significant factors [19]. To analyse case fact-patterns, Kastellec applied decision trees to search and seizure decisions of the US Supreme Court [14]. He noted certain characteristics of decision trees that make them useful in the legal domain. Decision trees are flexible because they do not rely on the assumptions of regression models, they inherently reveal interactions between input variables, they conform to the 'hierarchical and dichotomous structure' often employed in judicial decisions to answer questions and they are easily interpretable. Rissland & Friedman [20] applied decision trees with factor tests to model the evolving concept of 'good faith' in bankruptcy cases. By applying metrics to characterize the degree of change in the decision tree structures, they identified conceptual shifts over time in the legal decisions.

In the field of AI and Law, computational models of case-based argument employ factors to represent stereotypical fact patterns that tend to strengthen or weaken a plaintiff's argument in favour of a legal claim [21–26]. Until very recently, researchers had to manually identify the factors in the texts of legal cases in order to represent cases in a way that these computational models could process.

As explained below, our work suggests that researchers who perform empirical legal studies or build computational models of legal argument could employ text analytics to automatically classify factors in cases. If so, their decision trees and other ML models of case outcomes could be based on much larger numbers of cases, potentially increasing their accuracy and scope. Computational models of legal argument could contribute more effectively to legal practice if a program could automatically identify factors in a textual description of facts in case opinions or problem scenarios [27].

Research in legal text analytics has made some progress in automatically identifying factors in opinion texts—for example, developing annotation pipelines to extract factor-related information from trade-secret legal opinions [28], or from case summaries prepared by law students [29] and classifying trade-secret misappropriation opinions by applicable factors [30]. In [31], Gretok *et al.* applied an ML model to determine which test the US Supreme Court applied in Fourth Amendment cases to assess if police had demonstrated probable cause to search a car: 'totality of circumstances' or a 'bright line' test. Alcantara *et al.* [32] surveyed projects by applying text mining to court opinions to predict, based on input fact descriptions, criminal charges [33,34], the resulting judgment [35] or inconsistent cases [36]. In [37], factor values were automatically extracted from divorce cases using rules, augmented with word embeddings, and employed to predict and explain outcomes. In the semi-supervised case annotation for legal explanations (SCALE) project, Branting *et al.* trained an ML program to identify factual-finding tags pairing issues in World Intellectual Property Organization (WIPO) domain name dispute cases with applicable factors [13]. For example, the issue of 'No Rights or Legitimate Interests' (NRLI), a required element of a claim in a WIPO case, might be coupled with the related factor of PriorBizUse (i.e. 'Bona fide business use of Domain Name or demonstrable preparations to do so, prior to notice of the dispute'). SCALE automatically identified factual finding tags in case texts and predicted case outcomes. The team anticipated that eventually SCALE would explain outcome predictions in terms of the issue and factor labels; that is, in terms of reasons that legal professionals understand [13].

In our project, we apply supervised ML to automatically identify factors of suspicion in DIAS cases. This appears to be more challenging than identifying factors in the divorce cases in Li *et al.* [37] or the WIPO domain name cases of Branting *et al.* [13]. Auto stop fact situations likely are more factually diverse than those in WIPO domain name disputes and unlikely to be addressed by the rule-based extraction approach of Li *et al.* [37]. Since state and federal judges across the country write the auto stop domain name decisions, the opinion texts are more stylistically diverse than WIPO domain name arbitration cases written by arbitrators. Unlike [13], we apply a transformer language model, RoBERTa [38], which has been pre-trained on an extensive text corpus. We then fine-tune the model by applying it on a training set of auto stop cases. In [3], our multi-label approach to automatically classify factors of suspicion in auto stop cases achieved an average F1 score of 0.63. The classifier struggled with classes having low numbers of cases, specifically those with a test sample size of $n < 11$. As explained below, in this work we address that problem by employing a single-label multi-class approach.

Like the SCALE project, we employ a pipeline to identify factors in cases, but we use them to both predict and explain case outcomes. This is distinct from work that predicts case outcomes from the *full text* of legal decisions, such as Aletras *et al.* [39] and Chalkidis *et al.* [7,40]. As Branting *et al.* point out, 'such systems have very limited inherent explanatory capability' since they lack tags for factors or issues [13]. Unlike the SCALE project, we employ a combination of interpretable ML models known to be intuitively understandable, such as decision trees [14,41], and case-based techniques to illustrate positive and negative examples. In [42], Shaikh *et al.* created an ML model to explain outcomes of murder-related cases based on case judgments of the Delhi District Court. Since the explanatory features were extracted manually, only 86 cases were analysed. By contrast, we plan to use ML/NLP techniques to identify factors, eventually, in thousands of cases. In addition, in this work we employ the latest ML/NLP techniques, including the RoBERTa transformer language model.

There is a sizeable body of work in AI and Law that explores making the annotation effort more effective. Westermann *et al.* describe a method for building strong, explainable classifiers in the form of Boolean search rules [43] and a method based on sentence semantic similarity [44]. Savelka & Ashley evaluated the approach where a user labels the documents by inspecting predictions of an ML algorithm [45]. The application of active learning has also been analysed in the classification of statutory provisions [46] and eDiscovery [47,48]. Hogan *et al*. proposed a human-aided computer cognition framework for eDiscovery [49]. Savelka *et al.* also explored the transfer of ML models between different legal domains, jurisdictions [50] and even languages [51].

In this study, we employ a generative pre-trained transformer (GPT) language model, specifically OpenAI's GPT-4 [52], to annotate factors of suspicion in DIAS cases in zero-shot settings. The zero-shot performance of the GPT models has been evaluated by numerous studies in the context of legal texts. Yu *et al.* [53] analysed the capabilities of the models on the COLIEE entailment task based on the Japanese Bar exam. Similarly, Katz *et al.* [54] applied GPT-4 to the Uniform Bar Examination, and Bommarito *et al.* [55] applied it to the Uniform CPA Examination developed by the American Institute of Certified Public Accountants. Sarkar *et al.* [56] evaluated multiple techniques, including LLMs (BERT), in zero/few-shot classification of legal texts. GPT models have already been applied to analyse legal cases—for example, to: annotate sentences' roles in Board of Veterans' Appeals (BVA) cases, such as finding, evidence, legal rule, citation or reasoning [57]; predict Supreme Court Justice decisions [58]; determine how well a case passage explains a statutory term [59]; or generate interpretations of a term based on such passages [60,61]. Other studies by Blair-Stanek *et al.*, Nguyen *et al.* and Janatian *et al.* were focused on the capabilities of the GPT models to conduct legal reasoning [62–64], to model US Supreme Court cases [58], to give legal information to laypeople [65], and to support online dispute resolution [66]. Chalkidis examined the GPT model on the LexGLUE benchmark [67].

3. Pilot study

In the pilot study [3,4], we hypothesized that a program can employ ML/NLP methods to learn automatically to identify the essential elements in DIAS cases that courts consider in assessing the legality of the stop; that is, the factors which courts use to determine if a police officer had reasonable suspicion that the automobile contained drugs.

Traditionally, one must annotate training examples for an LLM to learn to classify features in case texts. In this section, we describe our procedure to fine-tune and use an LLM to automatically classify factor sentences in auto stop cases. This will highlight the importance of our factor-based models for prediction and explanation, as well as enable us to compare the GPT-based methodology of annotating factors described in §4. The advent of GPT-3.5 and GPT-4 are changing the annotation methods, as we illustrate in §4. Our pilot study predated these developments and relied on purely human annotation methods.

(a) Data

We begin with a description of our corpus. Based on the results of a series of 15 Westlaw keyword searches, we estimate there are a total of 37 000 DIAS cases. To analyse our hypothesis, we assembled a legal dataset of DIAS cases by retrieving court decisions from the Harvard Caselaw Access Project's (CAP) data corpus. We employed the CAP's search tool with keyword searches combining 'reasonable suspicion' or 'probable cause', and terms such as 'canine', 'drug dog', 'k-9', 'detain', 'detention', 'sniffing dog', 'drug interdiction', 'car', 'vehicle', 'truck' or 'automobile'. From the cases returned by our searches, we then selected the first[1] 211 relevant cases that addressed a motion to suppress evidence. Of these, the courts found reasonable suspicion in 63% of cases and found no reasonable suspicion in 37% [3,4].

We annotated a training set of case decisions by identifying instances of semantic types of information mentioned in sentences that are relevant for analysing the hypothesis, such as the presence of a factor of suspicion or the outcome of the issue (i.e. suspicion found or not). Our set of annotation types (i.e. our 'type system') includes the 20 factors shown in table 1. Eighteen of these are factors of suspicion and two relate to legal conclusions. We constructed this list of factors based on our litigation experience and after reading and analysing hundreds of legal opinions. Interestingly, as we subsequently learned, a very similar list of these factors originated in a 1986 Drug Enforcement Agency (DEA) programme, Operation Pipeline, instructing police officers how to detain a motorist that the officer wished to investigate [68].

[1] The cases were read and retrieved in the same order as they were returned by the CAP ranking algorithm.

Table 1. Factor type system.

(1) *occupant appearance or behaviour*	(4) *vehicle*
1A Furtive Movement	4K Expensive Vehicle
1B Nervous Behavior or Appearance	4L Vehicle License Plate or Registration
1C Suspicious or Inconsistent Answers	4M Unusual Vehicle Ownership
(2) *occupant status*	(5) *vehicle status*
2D Motorist License	5N Indicia of Hard Travel
2E Driver Status	5O Masking Agent
2F Refused Consent	5P Vehicle Contents Suggest Drugs
2G Legal Indications of Drug Use	5Q Suspicious Communication Device
2H Motorist's Appearance Related to Drug Use	5R Suspicious Storage
(3) *travel plans*	(6) *other annotation labels*
3I Possible Drug Route	6S Suspicion Found
3J Unusual Travel Plans	6T Suspicion Not Found

Seven paid law students (ranging from rising second-year students, to students who had just graduated) annotated court opinions to identify sentences that describe factors in our corpus of 211 cases. Annotating a single case required students to read the entire case, identify what factors courts identified as relevant to the decision and then proceed to annotate the opinion. For each sentence, students had to decide whether or not to apply a label.

Generally, students worked first in a group annotation exercise led by a legal expert, then individually annotated a small set of cases with expert feedback and finally did so on their own. In the initial 10–15 h of training led by a legal expert, we introduced them to the legal problem of auto stop cases, the type system outlined in table 1, a factor glossary, which describes and provides examples of each factor type, and an annotation guideline containing detailed instructions. We also introduced the annotators to Gloss, an online annotation environment that supports annotating sentences by colour-coded highlighting [69]. During this time, a legal expert periodically assessed the quality of the law students' annotations. To ensure sufficiently high inter-annotator agreement for an LLM to learn, we continually improved the annotation guidelines and monitored the annotations to resolve any disagreements.

Two students annotated each case. Our tool enables side-by-side comparison of texts marked up by two annotators. Using this feature, we instructed the annotators to try to resolve any conflicts and to compare to expert annotations. Finally, an expert reviewed all annotations and decided how any disputed sentences should be labelled. When the performance of the annotators reached sufficient quality, the full annotation procedure commenced.

To measure inter-annotator agreement, we used Cohen's kappa [70], which quantifies the likelihood that the agreement distribution results by chance rather than from real differences in the data. In [3,4], we reported an overall pairwise agreement mean kappa of 0.57, indicating moderate agreement according to Landis & Koch [71]. Identifying factors is a complex task and yet annotators can agree on factor descriptions in legal opinions that are on average 224 sentences long. Given the large number of sentences and the subtlety of the task, moderate agreement seems reasonable. The result served as the gold standard for the training data and for evaluating performance on the test set.

Overall, with training sessions, review sessions and annotation time, it took seven annotators, working a maximum of 10 h a week, about two months to complete the annotation of around 200 cases. In terms of sheer annotation time, we estimate that annotators were able, on average, to annotate 18 cases a week.

(b) Automatic identification of factors using fine-tuned LLMs

We conducted two experiments: one to assess how well an LLM could classify case sentences as instances of factors of suspicion and a second to assess how well the factors could explain the outcomes of the cases.

(i) Classification experiment and results

We developed a text processing pipeline and fine-tuned an LLM that learns automatically to classify texts by factors from the human-annotated training set. We employed RoBERTa [38], a member of the BERT-based family of transformer language models. The model was fine-tuned on the training set's annotated sentences and factor labels. The sentences were classified by whether they described a factor or conclusion, or were un-annotated (i.e. had no type). Approximately 52 000 sentences were used to train and test the model, using a 60–20–20% split of training, validation and testing sentences. This split was chosen to leave sufficient data for validation to ensure robustness of the results. The experiment evaluated how well the model learned to classify sentences as instances of factors compared with the human annotators' gold-standard classifications. It was structured as a multi-class (i.e. 20-way) classification problem. Ultimately, we calculated an accuracy of 0.97 and a macro accuracy of 0.84, and where there were 10 or more training sentences, the F1 scores ranged from 0.70 to 0.92.

These results provide promising evidence of the feasibility of automatically identifying factors in auto stop cases. In the next section, we describe a way in which these automatically identified sentences can be used for empirical legal analysis.

If we can automatically extract factors relevant for decisions about reasonable suspicion, we can scale up the numbers of cases we can include in our models. In particular, fine-tuned LLMs that can reliably identify factors of suspicion could dramatically reduce annotation costs and increase the scale at which cases can be analysed. We describe our method for factor annotation with LLMs in §4.

(ii) Case outcome explanation experiment, results and example

In the pilot study's second experiment, we selected 10 ML models we believed could be used to explain outcomes in the legal domain (table 2). In legal contexts, *interpretable and explainable* predictions are essential for understanding and communicating the reasoning behind the outputs. For this reason, we adopted easy-to-understand features (i.e. factors) and interpretable models (e.g. decision trees). Applying deep learning models directly to case texts, models that could identify predictive features on their own, would be interesting to investigate, especially since they could alleviate the need for complex feature engineering. As Branting *et al.* [13] have shown, however, there is no guarantee that the models' generated features would be as intelligible to legal professionals. In this study, therefore, we focus on explaining the decisions in terms of factors that model real-world legal reasoning. We leave as future work the possibility of applying deep learning models directly to case texts.

Employing the gold-standard annotations mentioned in §3(b)(i), we transformed the annotations into dichotomous vectors representing which factors were or were not present in each case. We trained each model on 80% of the gold-standard cases, and applied it to predict the outcomes of the remaining 20%. To train the models, we used cross-validation with 10-folds, repeating the procedure three times in order to ensure robustness of our results. As noted in Gray *et al.* [4], 'Given the cross-validation training and testing procedure and the accuracy on the test set, likelihood is very low that the models overfit the data'. We also identified the most important factors for the prediction.[2]

[2] Model Parameters: Random Forest: $m_{\text{try}} = 11$, Neural Network: Hidden Layers = 5, Decay = 0.1, XGBoost: $\eta = 0.05$, Iters = 200, Depth = 5, Generalized Linear Model: Binomial Family, Elastic Net: $\alpha = 1$, $\lambda = 0.0258$, kNN: $k = 14$, Distance = Jaccard, Weighted kNN: Weight = Optimal, Decision Tree: Complexity Parameter = 0.07.

Table 2. Performance of interpretable ML models to predict the outcome of cases.

model name	Acc	P	R	F1	three most important variables
random forest	0.975	1.0	0.970	0.985	5R, 4M, 1C
neural network	0.975	1.0	0.971	0.985	4M, 3I, 5O
XGBoost	0.951	1.0	0.941	0.969	3I, 1C, 5R
generalized linear model	0.902	0.916	0.971	0.942	1C, 3I, 5O
elastic net	0.902	0.895	1.0	0.945	1C, 3I, 5O
kNN	0.83	0.90	0.85	0.88	n.a.
weighted kNN	0.829	0.886	0.912	0.899	n.a.
decision tree	0.85	0.87	0.97	0.917	1C, 4M, 3I
most frequent label baseline	0.64	0.80	0.70	0.75	n.a.
random label baseline	0.46	0.81	0.38	0.52	n.a.

As shown in table 2, all the models tested outperformed two baseline models on accuracy, precision, recall and F1. One baseline predicted the outcome according to the most frequent label in the corpus (i.e. suspicion found). The other predicted outcomes at random. The accuracy of the former's baseline was 64% and the latter's accuracy was 46%. The best performing models were tied, with the neural network and the random forest models both predicting with 97% accuracy. The results suggest that our list of factors of suspicion, as shown in table 1, is effective for explaining case outcomes in the DIAS domain. Our sample size is still small, however; we will increase the number of cases in future work.

As noted, decision trees are interpretable and useful in the legal domain for a number of reasons, including the decision tree's similarity to reasoning in judicial opinions [14]. The decision tree model in table 2 can also provide a number of empirical insights into decision-making in DIAS cases. For example, at the root node of the decision tree in figure 1, the model first asks whether factor 4M is present in the input case. Factor 4M, Unusual Vehicle Ownership, has to do with whether the car is rented or otherwise owned by a third party. If 4M is present, the left branch is followed and one reaches a terminal node. The model predicts that reasonable suspicion is found with a probability of 0.78. The presence of factor 4M resolves 43% of the training data. If 4M is not present, the right branch is followed and the model asks if factor 1C, Suspicious or Inconsistent Answers, is present. If yes, the left branch leads to a terminal node and a prediction of reasonable suspicion found with a probability 67%. If factor 4M is not present, the presence of factor 1C, Suspicious or Inconsistent Answers, resolves another 28% of cases. Similar observations apply with respect to factors 3I, Possible Drug Route, and 2H, Motorist's Appearance Related to Drug Use.

The decision tree provides significant information about the auto stop legal domain, at least based on the 211 cases in our corpus. The insight gleaned from this model is that with knowledge of the presence or the absence of just four factors of suspicion, the model can explain all the case outcomes with 80% accuracy. This raises important questions. Driving a rental car is an innocent activity and the basis of an entire industry. Normatively, one may wonder if it should be so important a feature in auto stop scenarios, especially given the risks associated with auto stops?

The models in table 2 also enable systematic ways to explain the outcome of a case. Given a case of interest, a current fact situation, the kNN model can identify the most similar cases. One can use a similarity metric to select the cases that maximize factors shared with the focal case and minimize unshared ones.[3] Comparing the current fact situation with the most similar cases may

[3] In pioneering work, Mackaay & Robillard [72] applied a kNN model to Canadian tax cases.

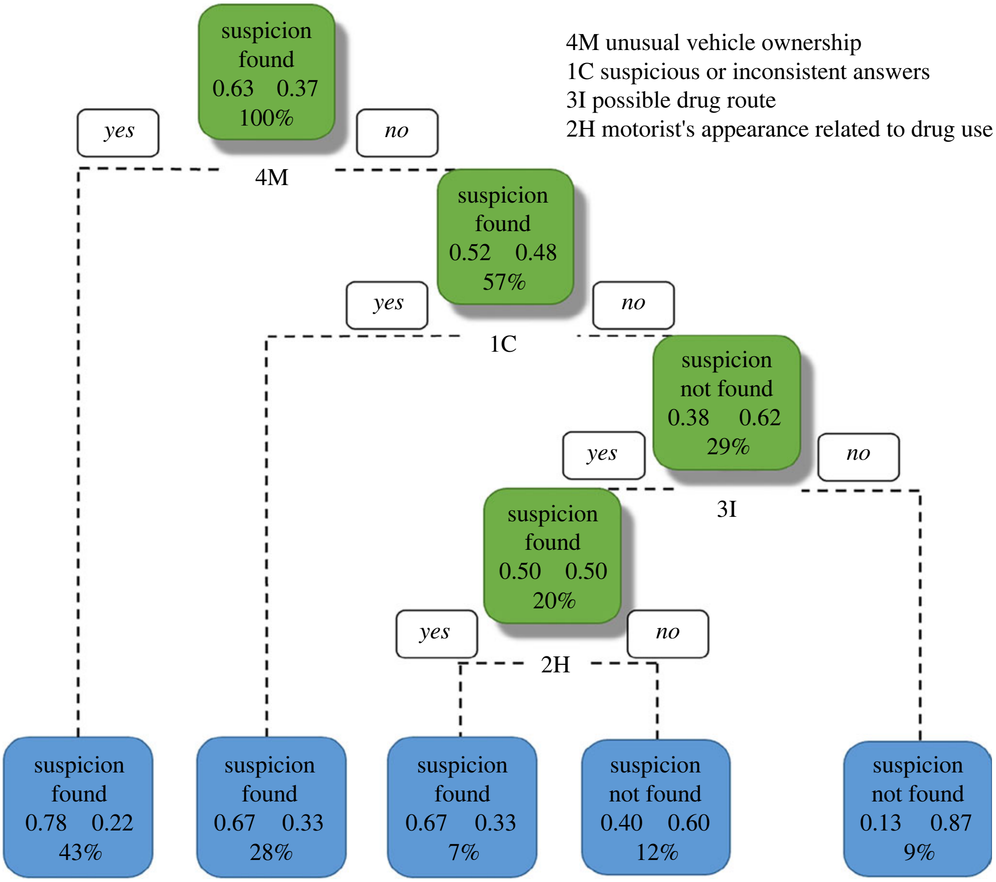
4M unusual vehicle ownership
1C suspicious or inconsistent answers
3I possible drug route
2H motorist's appearance related to drug use
suspicion found 0.63 0.37 100%
yes
no
4M
suspicion found 0.52 0.48 57%
yes
no
1C
suspicion not found 0.38 0.62 29%
yes
no
3I
suspicion found 0.50 0.50 20%
yes
no
2H
suspicion found 0.78 0.22 43%
suspicion found 0.67 0.33 28%
suspicion found 0.67 0.33 7%
suspicion not found 0.40 0.60 12%
suspicion not found 0.13 0.87 9%

Figure 1. Decision tree [4].

- *current fact situation:*
  - 1B Nervous Behavior or Appearance
  - 1C Suspicious or Inconsistent Answers
  - 4M Unusual Vehicle Ownership
  - 5N Indicia of Hard Travel
  - 5O Masking Agent
- rank cases by Jaccard dissimilarity:
  - select cases that maximize factors shared with CFS and minimize unshared ones.

- case name: USA versus *Anguiano*
  rank = 1:
  score: 0.2
  similar factors: ‘1B’, ‘4M’, ‘1C’, ‘5N’
  dissimilar factors: 5O–
  outcome: **suspicion found**
- case name: *State* versus *Myles*
  rank = 2:
  score: 0.3
  similar factors: ‘1B’, ‘4M’, ‘1C’
  dissimilar factors: ‘3J*’, ‘5N–’, ‘5O–’
  outcome: **suspicion found**

Figure 2. Explaining with neighbouring cases.

lead to a better understanding of which factors influence a fact situation’s outcome. In figure 2, the two most similar cases, USA versus Anguiano and State versus Myles, both happen to have findings of reasonable suspicion. Nearby cases might well have the opposite outcome, however, which might reduce one’s confidence in a prediction and lead one to investigate the factors whose presence or absence could make a difference in outcomes.

While such explanations will be especially useful in AI and Law tools to assist legal practitioners (e.g. judges and counsel) to evaluate particular cases, they also enable researchers in empirical legal studies to deepen their understanding of outcome predictions. For example, they discriminate weaker predictions that are subject to apparent counterarguments from those that are not.

In sum, our pilot study demonstrates the utility of applying fine-tuned LLMs to identify factors in case decisions and to compute factors' weights in order to explain case outcomes. These are two ways in which our work can reduce the complexity of empirical legal analysis. At the same time, however, it highlights the costs of manually annotating training sets of the ever larger numbers of cases needed to sharpen the empirical legal analysis. Given the cost of manual annotation, analysing even larger numbers of cases will only be possible to the extent that text analytics can reliably identify factors automatically at scale. In the next section, we demonstrate how GPT language models can simplify manual annotation and reduce costs even more by partially automating annotation of legally relevant factors.

4. Factor annotation with LLMs

GPT-FMs can reduce the complexity and cost of empirical legal analysis even more by partially automating annotation of legally relevant factors. This does not eliminate the need for humans to be involved in annotation, but it simplifies the human's role and makes annotation more efficient. Instead of finding and labelling the sentences that are instances of factors of suspicion, the human needs only to confirm GPT's sentence selections and labels.

(a) Prompting GPT-FMs for annotation

We employed `gpt-3.5-turbo-16k` (GPT-3.5), `gpt-4` (GPT-4) and `gpt-4-1106-preview` (GPT-4-Turbo) to annotate sentences in terms of factors. We focused on shorter cases to see how well GPT-FMs perform without the potential noise generated by long cases and to reduce the cost of the experiment. To facilitate this, we randomly selected 35 cases from our corpus of 211 cases. (See table 3 for a summary of the corpus's case sentence statistics.) Thirty of those cases were each less than or equal to 120 sentences long (roughly from the 25th quartile and below). To assess whether our results hold for longer, more complex cases, we also randomly selected five cases that were 280 sentences long or more (roughly from the 75th quartile and above).

Using the GPT-FMs to perform a task requires that the user prompts the model with instructions on what they want the model to do. Our approach relies on a single prompting method: a categorization guideline approach. To develop the guideline prompt, we follow [59] and draft the prompt with almost an exact copy of the annotation guidelines provided to annotators in Gray *et al.* [3]. Table 4 illustrates an abridged example of a guideline prompt.[4]

As shown in item 1, we begin the prompt by instructing the model that we will be searching sentences for factors. Under item 2, we describe the legal problem and give an example fact pattern. In item 3, we describe and define each of the factors. Next, we provide the model with specific instructions to follow, as shown in item 4. Under item 5, the model is given example sentences and labels. Item 5 instructs the model to label each sentence and provides some example sentences and labels. Lastly, in item 6, we provide the model with sentences to label and formatting instructions.

Complex prompts often encounter unexpected problems and need to be developed incrementally. We employed five cases from the 30 randomly selected shorter cases to develop the final versions of the prompt. The original drafts of the prompts were based on the factor definitions and annotation guidelines provided to the human annotators in Gray *et al.* [3]. As suggested in Savelka *et al.* [59], we then augmented them as necessary over seven iterations of testing, editing prompts that had caused the model to assign labels erroneously. For guideline prompts, the full input given to the model included the base guideline prompt and up to 60 sentences. GPT-4 can handle a total of only 8192 tokens, while GPT-3.5 and GPT-4-Turbo can

[4] To view the full prompts used in the experiments, visit this github.

Table 3. Corpus case sentence statistics.

minimum number of sentences in case:	33
maximum number of sentences in case:	1090
average number of sentences in case:	224
median number of sentences in case:	185
number of sentences in 25th quartile:	123
number of sentences in 75th quartile:	288

Table 4. Abridged prompt example.

1. TASK In this task, we are attempting to label sentences to assess whether they contain important information. [81 characters…]
2. -START LEGAL PROBLEM EXPLANATION- We are interested in highway drug interdiction. This occurs when a motorist is stopped by police [2694 characters…] -END LEGAL PROBLEM EXPLANATION-
3. -START INSTRUCTIONS- You will assess sentences to determine whether they belong to any of the following categories…
Furtive Movement—Use this label if the driver or passenger in the vehicle makes a suspicious movement [5759 characters…]
4. You are also to follow these specific rules:
Typically, a sentence will describe a single factor, however, in some cases, a single sentence may include more than one factor [1738 characters…]
5. You should apply a label for each sentence.
Here are some examples:
1. Sentence: Officer Guthrie testified that while the above exchanges were taking place, he noticed that the driver, Arturo Tapia, seemed nervous, and that his hands were shaking.
Label: Physical Appearance of Nervousness [502 characters…]
-END INSTRUCTIONS-
6. -START LABELLING-
Label all of these *n* sentences:
[*n* Sentences]

handle more tokens. To keep our experiments comparable, we limited all experiments to within 8192 tokens.

In using GPT-FMs, one needs to set values for the parameters shown in table 5. Generally, the parameters have the following effects. ‘Model’ describes which model is to be used; in our case, this parameter could be `gpt-4`, `gpt-3.5-turbo-16k` or `gpt-4-1106-preview`. ‘Temperature’ controls the randomness of the model; 0 corresponds to no randomness. The higher the temperature, the more creative the output but it can also be less factual. As the temperature approaches 0, the model becomes more deterministic, which we deem important for achieving reproducible results. ‘Max tokens’ controls the maximum length (number of tokens) of the completion (i.e. the output). A token roughly corresponds to a word. Note that each model has a length limit on the prompt, and the completion counts towards that limit. While `gpt-4` allows for 8192 tokens, `gpt-3.5-turbo-16k` accepts up to 16 385 tokens and `gpt-4-1106-preview` accepts up to 128 000 tokens. ‘Top P’ is also related to randomness; when temperature is set to 0, this should be set to 1. ‘Frequency penalty’ adds a penalty for repetitions. ‘Presence penalty’

Table 5. Parameters used when prompting GPT-4.

MODEL	gpt-4
TEMPERATURE	0
MAX_TOKENS	3000
TOP_P	1
FREQUENCY_PENALTY	0
PRESENCE_PENALTY	0

Table 6. The averages of using each model in terms of tokens and cost.

	tokens			cost		
average	input	output	total	input	output	total
GPT-4	13 074	4757	17 832	0.39	0.28	0.68
GPT-3.5	13 074	4706	17 781	0.01	0.009	0.02
GPT-4-Turbo	13 074	4713	17 788	0.13	0.14	0.27

applies to tokens appearing multiple times in the output. Because we are repeatedly using the same labels, we set both frequency and presence penalties to 0.

Lastly, we break down each method from the view of the resources needed to prompt each of the models. The token counts were calculated with the `tiktoken` package available in Python.

The results are shown in table 6. Generally, we see that the average prompt tokens per model are the same—as they should be because we used the same prompts for each model. We see that the average number of tokens is the same, but that GPT-4 has a tendency to output slightly more tokens than GPT-3.5 or GPT-4-Turbo. Lastly, GPT-4 is far more expensive than the other two models, with GPT-3.5 being the least expensive. According to OpenAI, at the time of writing, the cost of using GPT-4 is `$0.03/1K tokens` for input and `$0.06/1K tokens` for output. Using GPT-3.5 is `$0.0010/1K tokens` for input and `$0.0020/1K tokens` for output; using GPT-4-Turbo is `$0.01/1K tokens` for input and `$0.03/1K tokens` for output.[5]

(b) Results and discussion

We assessed the performance of the GPT-FMs by comparing the predicted label to the gold-standard annotation that a legal expert familiar with the problem assigned to the contents of each sentence. We measured each model's ability across the standard measures of precision, recall and F1-score for each category. We also measured each model's overall performance in terms of how well it could:

- correctly label *each* individual sentence, i.e. accuracy,
- correctly identify the total number of different factors present in each case. That is, if a factor was present *at all* in the case, did the model identify it, i.e. the intersection between the model and the gold-standard annotations?[6]
- avoid falsely identifying factors that were not in the case.

In the next section, we present and summarize our results of the different capabilities of GPT-FMs to perform the classification of sentences.

[5] At the time of writing, costs can be found at this link: https://openai.com/pricing.

[6] We define the intersection as the shared factors between the model and gold-standard annotations. For example, the intersection between set_1 {'a', 'b', 'd'} and set_2 {'a', 'c', 'd'} is {'a', 'd'}.

Table 7. The performance of GPT-3.5, GPT-4 and GPT-4-Turbo on the classification task. The F1 metric is reported for each individual type because both precision and recall are important in the legal domain (the best F1 metric is bolded for each type). Overall metrics including accuracy are reported for each model.

	GPT-3.5			GPT-4			GPT-4-Turbo		
individual type	P	R	F1	P	R	F1	P	R	F1
2E	0.00	0.00	0.00	0.30	1.00	**0.50**	0.00	0.00	0.00
4K	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1A	0.57	0.26	0.36	0.73	0.86	**0.79**	0.69	0.74	0.71
5N	0.33	0.75	0.46	0.20	0.50	0.29	1.00	0.50	**0.67**
2G	0.49	0.40	0.44	0.75	0.92	**0.82**	0.60	0.84	0.70
5O	0.75	0.68	0.71	0.90	1.00	**0.95**	0.79	0.79	0.79
2D	0.30	0.46	0.36	0.26	0.88	0.40	0.56	0.43	**0.49**
2H	0.42	0.26	0.32	0.60	0.23	0.33	0.91	0.59	**0.71**
1B	0.70	0.63	0.67	0.88	0.80	**0.84**	0.82	0.66	0.73
NF	0.90	0.87	0.89	0.98	0.89	**0.94**	0.93	0.94	**0.94**
3I	0.50	0.09	0.15	0.77	0.89	**0.83**	0.58	0.67	0.62
2F	0.65	0.70	0.68	0.66	1.00	0.80	0.74	0.89	**0.81**
6S	0.33	0.62	0.43	0.53	0.88	**0.66**	0.60	0.67	0.63
6T	0.23	0.33	0.27	0.33	0.77	**0.46**	0.38	0.48	0.42
5Q	0.50	0.55	0.52	0.44	1.00	**0.62**	0.50	0.43	0.46
5R	0.22	0.06	0.10	0.63	0.73	**0.68**	0.65	0.35	0.46
1C	0.28	0.35	0.31	0.51	0.85	**0.63**	0.63	0.41	0.49
3J	0.38	0.57	0.46	0.57	0.77	**0.65**	0.50	0.59	0.54
4M	0.34	0.20	0.25	0.60	0.65	**0.62**	0.55	0.30	0.39
5P	0.11	0.36	0.17	0.25	0.63	0.36	0.33	0.56	**0.42**
4L	0.23	0.25	0.24	0.12	0.50	0.20	0.36	0.33	**0.35**
accuracy:	0.79			0.91			0.89		
intersection:	0.81			0.97			0.87		
false factors:	2.9			2.6			1.5		

(i) Quality of GPT-FM annotations

As shown in table 7, the overall accuracy of GPT-3.5, GPT-4 and GPT-4-Turbo using the guideline prompts is encouraging. Generally, we see that GPT-4 and GPT-4-Turbo greatly out perform GPT-3.5. The performance between GPT-4 and GPT-4-Turbo is mixed, with the latter sometimes producing larger F1-scores for individual categories. The accuracy between GPT-4 and GPT-4-Turbo is similar, with the former performing slightly better. GPT-4 is better able to identify the different factors present in each case (the intersection between gold standard and GPT-4). However, GPT-4-Turbo is better at avoiding identifying false factors, i.e. false positives. Thus, it seems as though GPT-4-Turbo produces fewer false factors but is less able to find all individual factors identified by gold-standard annotations. The reverse is true for GPT-4; it can find nearly all factors identified by gold-standard annotations but produces more false factors than GPT-4-Turbo (table 8).

Table 8. Overall comparisons between all cases, short cases and long cases.

	all cases			short cases			long cases		
GPT model	3.5	4	4-Turbo	3.5	4	4-Turbo	3.5	4	4-Turbo
accuracy	0.79	0.91	0.89	0.77	0.91	0.88	0.86	0.89	0.93
intersection	0.81	0.97	0.87	0.80	0.97	0.87	0.86	0.99	0.86
false factors	2.9	2.6	1.5	2.7	2.4	1.3	4.0	4.6	2.6

In our dataset, since the vast majority of sentences (exactly 42 363 or over 90%) describe no factor, a model could feasibly perform well by simply predicting ‘no factor’ for every single sentence. As shown in table 7, this did not happen. We can determine how well the model predicts *relevant* factors in each case. We assess the number of different factors properly identified per case and the number of false factors wrongly identified per case. Given the dichotomous representation we used in the pilot study to predict and explain outcomes, if a factor is identified even once in a case, the factor will be counted as present in the case. GPT-4 correctly identified the different factors present in each case 91% of the time and identified 2.6 false factors per case. GPT-4-Turbo identified the different factors in a case 89% of the time and 1.5 false factors per case. We believe these results provide promising evidence that GPT-4 and GPT-4-Turbo are good starting points for accurate factor annotation.

Because these experiments dealt with only a subset of the cases analysed in Gray *et al.* [3,4] as reported in the pilot study in §3(b), we cannot directly compare GPT-4’s accuracy in factor annotation with that of the fine-tuned LLMs. We can, however, identify some qualitative differences between GPT-FMs and the fine-tuned LLMs. By making use of the context of a sentence, the GPT-FMs enable certain annotations that were not possible in [3,4]. In those experiments, annotators were instructed only to annotate sentences that made *explicit* reference to a factor. The reason was that the LLM models only considered one sentence at a time. Thus, the context window was limited. As a result, the following three sentences would not be labelled:

1. While writing the ticket, Deputy Kinik explained to Cowart–Darling that he was a K-9 officer and asked if there was anything in the car that could cause a drug detection dog to alert.
2. Cowart–Darling shook his head up and down in an affirmative gesture.
3. When the deputy asked about the nature of the illegal substance, Cowart–Darling denied having anything illegal in the vehicle.

Cowart–Darling contradicted himself about whether the vehicle contained contraband. GPT-4 properly labelled sentences two and three as ‘1C Suspicious or Inconsistent Answers’ as clearly there is an inconsistency between these two sentences. This illustrates how the GPT-FMs’ subtle reasoning can improve classification in contrast to other LLMs that cannot leverage contextual information as well.

The GPT-FMs are not infallible, however. The models seem to be insensitive to some instructions in the prompt, which can lead to false positives (i.e. incorrect factors). For example, the factor ‘5P Vehicle Contents Suggest Drugs’ applies where the officer sees indications that the vehicle contains drugs or paraphernalia (e.g. a scale and baggies). Given our instruction in the prompt, however, the model tended to label ‘Vehicle Contents Suggest Drugs’ whenever such items were mentioned. This is evident in table 7, where all models report higher recall but low precision for this factor. Other factors, upon investigation, show similar tendencies, including ‘2D Motorist License’ and ‘4L Vehicle License Plate or Registration’. The model tends to predict these labels whenever certain terms are mentioned, even in neutral contexts, such as ‘[The officer] asked for [the motorist’s] driver’s license and proof of insurance’.

The table also shows that, in terms of differences between shorter (≤25th quartile) and longer (≥ 75th quartile) cases, the difference in accuracy is minimal with respect to GPT-4 and GPT-4-Turbo. The performance of GPT-4 dips slightly, but GPT-4-Turbo performed better on longer cases. As to the intersection of the factors, GPT-4 performs slightly better and GPT-4-Turbo is about the same. GPT-4 and GPT-4-Turbo generally hallucinate more. In terms of accuracy and the intersection of factors, GPT-3.5 performed best on longer cases.

(ii) Cost of GPT-4 annotations

On the whole, using GPT-FMs for annotation is much less expensive than paying law students to annotate cases. In the work reported in Gray *et al.* [3,4], described in the above pilot study, we paid 7 hired law students about $8500 dollars to annotate 211 cases (i.e. about $40.00 per case or more than $1400 for 35 cases.) In performing the annotation task, the human annotators read each case, selected the sentences that described factors and assigned the factor labels; a time-consuming effort that required months of work.

By contrast, it cost $23.72 to prompt GPT-4, $0.79 to prompt GPT-3.5 and $9.53 to prompt GPT-4-Turbo. We estimate that it took a total of about $45.00 to conduct the experiments. This includes the cost associated with crafting the prompts and debugging the programs. To decide on the gold standard, a single expert reviewed each labelled sentence and decided whether the label reflected a reasonable choice. The expert completed the task in just 8 h, including the time to double check the work. Even employing two humans to review labels assigned by GPT-FMs would not be nearly as costly as manual annotation.

In sum, employing GPT-FMs with guideline prompting for annotation appears to be more efficient and less expensive than traditional manual annotation of training instances. Using GPT-FMs as a first step in annotating cases dramatically reduces the work of humans, who need only confirm the annotations suggested by GPT-FMs; a much less time-consuming task. Given the positive, if preliminary, results of GPT-4 and GPT-4-Turbo reported in table 7, it seems plausible that using GPT-FMs can eliminate months of multi-person annotation tasks.

5. Limitations

This paper presents a framework and methodology for automatically annotating factors with which to predict and interpret outcomes of past legal cases. We focus on the use of interpretable and explainable methods to illuminate the case outcome predictions. Our work demonstrates the potential for predicting future cases and techniques that could be applied to similar tasks in other legal domains. Our results are subject to a number of limitations. First, we have so far employed a relatively limited number of cases. We will continue to add more. Second, our methodology struggles with annotating longer cases, identifying too many false positives. We plan to explore modifications to address this issue. Third, a single expert assessed the quality of the GPT-FMs' performance on this task. We plan to enlist multiple experts for future evaluations. Finally, as explained above, the model sometimes ignores explicit annotation instructions. We will seek language for prompts that better captures the descriptions of problematic factors.

6. Future work

We are currently exploring the use of GPT-3.5, GPT-4 and GPT-4-Turbo as an annotator's assistance tool with a greater number of cases. Moreover, we hope to make our annotation/classification experiments more efficient by implementing a dual classifier procedure. We will first use a binary 'coarse' classifier to distinguish between helpful and unhelpful sentences. Helpful sentences describe factors or contain information that the model could use as context in distinguishing whether a sentence describes a factor. Unhelpful sentences include those describing points of law or legal citations. By reducing the number of unhelpful sentences, we may scale up the GPT-FM annotation task, enabling us to annotate many more cases semi-automatically.

7. Conclusion

As explained in §2, factor analysis is widely used to analyse certain legal domains in terms of factual elements that influence case outcomes. It employs factors to reduce legal complexity: factors generalize the case facts that are important in the domain, enabling empirical legal researchers to analyse the cases statistically and determine the factors' relative importance. Traditionally, however, factor analysis has required expensive manual annotation of case texts by legal experts.

In this paper, we have provided a methodological framework that could reduce legal complexity and cost in domains where factor analysis is important. The pilot study of §3 shows how ML/NLP can learn to identify factors in cases automatically and explain case outcomes in terms of factors. It also emphasizes the need to analyse many more cases to improve the empirical legal analysis; a time-consuming and expensive task if manual case annotation is employed.

The method described in §4 for semi-automated factor annotation using GPT-FMs shows how to alleviate much of the burden of human annotation. While our demonstration focused on factors of suspicion in DIAS cases, in principle the framework could be adapted to other factor-based domains, such as trade-secret misappropriation. Our framework for more efficient and accurate identification of relevant factors could support empirical legal analysis of factors and their weights based on much larger numbers of cases than previously possible. Ultimately, the annotations supplied by GPT-FMs could serve as a starting point for annotators, reducing the cost and increasing the efficiency of annotation as compared with the resources needed to fine-tune an LLM such as BERT or RoBERTa. The GPT-FMs also are capable of leveraging valuable contextual information that is not available with traditional encoder LLMs.

Data accessibility. This article has no additional data.

Declaration of AI use. We have not used AI-assisted technologies in creating this article.

Authors' contributions. M.A.G.: conceptualization, data curation, formal analysis, investigation, methodology, software, writing—original draft, writing—review and editing; J.S.: conceptualization, formal analysis, methodology, supervision, writing—original draft, writing—review and editing; W.M.O.: conceptualization, data curation, formal analysis, investigation, writing—original draft, writing—review and editing; K.D.A.: conceptualization, formal analysis, funding acquisition, methodology, supervision, writing—original draft, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. We received funding from Pitt Momentum Funds Teaming and Scaling Grants, and from the Center for Text Analytic Methods in Legal Studies at the University of Pittsburgh.

Acknowledgements. We acknowledge support for this research from Pitt Momentum Funds Teaming and Scaling Grants, and from the Center for Text Analytic Methods in Legal Studies at the University of Pittsburgh (www.law.pitt.edu/center-text-analytic-methods-legal-studies).

References

1. Bambauer J. 2015 Hassle. *Mich. L. Rev.* **113**, 461.
2. LaFave W. 2004 The 'routine' traffic stop from start to finish: too much 'routine', not enough law. *Mich. L. Rev.* **102**, 1843. (doi:10.2307/4141969)
3. Gray M, Savelka J, Oliver W, Ashley K. 2022 Toward automatically identifying legally relevant factors. In *Legal knowledge and information systems*, pp. 53–62. Amsterdam, Netherlands: IOS Press.
4. Gray M, Savelka J, Oliver W, Ashley K. 2023 Automatic identification and empirical analysis of legally relevant factors. In *Int. Conf. of Artificial Intelligence and Law* (*ICAIL 2023*). New York, NY: ACM Press.
5. Hendrycks D, Burns C, Chen A, Ball S. 2021 Cuad: an expert-annotated NLP dataset for legal contract review. (https://arxiv.org/abs/2103.06268)

6. Song D, Gao S, He B, Schilder F. 2022 On the effectiveness of pre-trained language models for legal natural language processing: an empirical study. *IEEE Access* **10**, 75 835–75 858. (doi:10.1109/ACCESS.2022.3190408)
7. Chalkidis I, Fergadiotis M, Malakasiotis P, Aletras N, Androutsopoulos I. 2020 LEGAL-BERT: the muppets straight out of law school. (https://arxiv.org/abs/2010.02559)
8. Chen H, Wu L, Chen J, Lu W, Ding J. 2022 A comparative study of automated legal text classification using random forests and deep learning. *Inf. Process. Manage.* **59**, 102798. (doi:10.1016/j.ipm.2021.102798)
9. Chen H, Pieptea LF, Ding J. 2022 Construction and evaluation of a high-quality corpus for legal intelligence using semiautomated approaches. *IEEE Trans. Reliab.* **71**, 657–673. (doi:10.1109/TR.2022.3156126)
10. Zhong H, Xiao C, Tu C, Zhang T, Liu Z, Sun M. 2020 How does NLP benefit legal system: a summary of legal artificial intelligence. (https://arxiv.org/abs/2004.12158)
11. Nguyen HT, Nguyen MP, Vuong THY, Bui MQ, Nguyen MC, Dang TB, Tran V, Nguyen LM, Satoh K. 2022 Transformer-based approaches for legal text processing: JNLP Team-COLIEE 2021. *Rev. Socionetwork Strateg.* **16**, 135–155. (doi:10.1007/s12626-022-00102-2)
12. Shao Y, Mao J, Liu Y, Ma W, Satoh K, Zhang M, Ma S. 2020 BERT-PLI: modeling paragraph-level interactions for legal case retrieval. In *Int. Joint Conf. on Artificial Intelligence Organization (IJCAI 2021), Yokohama, Japan*, pp. 3501–3507. IJCAI.
13. Branting LK, Pfeifer C, Brown B, Ferro L, Aberdeen J, Weiss B, Pfaff M, Liao B. 2021 Scalable and explainable legal prediction. *Artif. Intell. Law* **29**, 213–238. (doi:10.1007/s10506-020-09273-1)
14. Kastellec JP. 2010 The statistical analysis of judicial decisions and legal rules with classification trees. *J. Empir. Leg. Stud.* **7**, 202–230. (doi:10.1111/j.1740-1461.2010.01176.x)
15. Rempell S. 2022 Factors. *Buff. L. Rev.* **70**, 1755. (doi:10.2139/ssrn.4095435)
16. Beebe B. 2006 An empirical study of the multifactor tests for trademark infringement. *Calif. L. Rev.* **94**, 1581. (doi:10.2307/20439078)
17. Beebe B. 2007 An empirical study of US copyright fair use opinions, 1978–2005. *U. Pa. L. Rev.* **156**, 549.
18. Beebe B. 2020 An empirical study of US copyright fair use opinions updated, 1978–2019. *NYU J. Intell. Prop. Ent. L.* **10**, 1.
19. Shao HL, Leflar RB, Huang SC. 2022 Factors determining child custody in Taiwan after patriarchy's decline: decision tree analysis on family court decisions. *Asian J. Comp. Law* **18**, 272–288. (doi:10.1017/asjcl.2022.28)
20. Rissland EL, Friedman MT. 1995 Detecting change in legal concepts. In *Proc. of the 5th Int. Conf. on Artificial Intelligence and Law, College Park, MD*, pp. 127–136. New York, NY: Association for Computing Machinery.
21. Ashley KD. 1990 *Modeling legal arguments: reasoning with cases and hypotheticals*. New York, NY: MIT Press.
22. Bench-Capon T. 2017 HYPO's legacy: introduction to the virtual special issue. *Artif. Intell. Law* **25**, 205–250. (doi:10.1007/s10506-017-9201-1)
23. Grabmair M. 2016 Modeling purposive legal argumentation & case outcome prediction using argument schemes in the value judgment formalism. Language: English.
24. Chorley A, Bench-Capon T. 2005 An empirical investigation of reasoning with legal cases through theory construction and application. *AI Law* **13**, 323–371. (doi:10.1007/s10506-006-9016-y)
25. Chorley A, Bench-Capon T. 2005 AGATHA: using heuristic search to automate the construction of case law theories. *Artif. Intell. Law* **13**, 9–51. (doi:10.1007/s10506-006-9004-2)
26. Westermann H, Walker VR, Ashley KD, Benyekhlef K. 2019 Using factors to predict and analyze landlord-tenant decisions to increase access to justice. In *Proc. of the Seventeenth Int. Conf. on Artificial Intelligence and Law, Montreal, QC*, pp. 133–142. New York, NY: Association for Computing Machinery.
27. Ashley KD. 2017 *Artificial intelligence and legal analytics*. Cambridge, UK: Cambridge University Press.
28. Wyner A, Peters W. 2010 Towards annotating and extracting textual legal case factors. In *SPLeT-2010, Valletta, Malta*, pp. 36–45. Paris, France: European Language Resources Association (ELRA).

29. Ashley KD, Brüninghaus S. 2009 Automatically classifying case texts and predicting outcomes. *Artif. Intell. Law* **17**, 125–165. (doi:10.1007/s10506-009-9077-9)
30. Falakmasir M, Ashley K. 2017 Utilizing vector space models for identifying legal factors from text. In *JURIX 2017*, vol. 302, pp. 183–192. Amsterdam, Netherlands: IOS Press.
31. Gretok E, Langerman D, Oliver WM. 2020 Transformers for classifying fourth amendment elements and factors tests. In *Legal Knowledge and Information Systems*, pp. 63–72. Amsterdam, Netherlands: IOS Press.
32. Alcántara Francia OA, Nunez-del Prado M, Alatrista-Salas H. 2022 Survey of text mining techniques applied to judicial decisions prediction. *Appl. Sci.* **12**, 10200. (doi:10.3390/app122010200)
33. Luo B, Feng Y, Xu J, Zhang X, Zhao D. 2017 Learning to predict charges for criminal cases with legal basis. (https://arxiv.org/abs/1707.09168)
34. Xiao C*et al.* 2018 Cail2018: a large-scale legal dataset for judgment prediction. (https://arxiv.org/abs/1807.02478)
35. Zhong H, Guo Z, Tu C, Xiao C, Liu Z, Sun M. 2018 Legal judgment prediction via topological learning. In *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing, Brussels, Belgium*, pp. 3540–3549. Stroudsburg, PA: Association for Computational Linguistics. (doi:10.18653/v1/D18-1390)
36. Xu N, Wang P, Chen L, Pan L, Wang X, Zhao J. 2020 Distinguish confusing law articles for legal judgment prediction. (https://arxiv.org/abs/2004.02557)
37. Li J, Zhang G, Yu L, Meng T. 2019 Research and design on cognitive computing framework for predicting judicial decisions. *J. Signal Process. Syst.* **91**, 1159–1167. (doi:10.1007/s11265-018-1429-9)
38. Liu Y*et al.* 2019 Roberta: a robustly optimized bert pretraining approach. (https://arxiv.org/abs/1907.11692)
39. Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. 2016 Predicting judicial decisions of the European Court of Human Rights: language processing perspective. *PeerJ Comput. Sci.* **2**, e93. (doi:10.7717/peerj-cs.93)
40. Chalkidis I, Androutsopoulos I, Aletras N. 2019 Neural legal judgment prediction in English. (https://arxiv.org/abs/1906.02059)
41. Lipton ZC. 2016 The Mythos of model interpretability. *CoRR* (https://arxiv.org/abs/1606.03490)
42. Shaikh RA, Sahu TP, Anand V. 2020 Predicting outcomes of legal cases based on legal factors using classifiers. *Proc. Comput. Sci.* **167**, 2393–2402. (doi:10.1016/j.procs.2020.03.292)
43. Westermann H, Savelka J, Walker VR, Ashley KD, Benyekhlef K. 2019 Computer-assisted creation of boolean search rules for text classification in the legal domain. In *JURIX, Madrid, Spain*, pp. 123–132. Amsterdam, Netherlands: IOS Press.
44. Westermann H, Savelka J, Walker VR, Ashley KD, Benyekhlef K. 2020 Sentence embeddings and high-speed similarity search for fast computer assisted annotation of legal documents. In *Legal Knowledge and Information Systems: JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9–11, 2020*, vol. 334, p. 164. Amsterdam, Netherlands: IOS Press.
45. Šavelka J, Trivedi G, Ashley KD. 2015 Applying an interactive machine learning approach to statutory analysis. In *Legal Knowledge and Information Systems*, pp. 101–110. Amsterdam, Netherlands: IOS Press.
46. Waltl B, Muhr J, Glaser I, Bonczek G, Scepankova E, Matthes F. 2017 Classifying legal norms with active machine learning. In *JURIX, Luxembourg*, pp. 11–20. Amsterdam, Netherlands: IOS Press.
47. Cormack GV, Grossman MR. 2016 Scalability of continuous active learning for reliable high-recall text classification. In *Proc. of the 25th ACM Int. on Conf. on Information and Knowledge Management, Indianapolis, IN*, pp. 1039–1048. New York, NY: Association for Computing Machinery.
48. Cormack GV, Grossman MR. 2015 Autonomy and reliability of continuous active learning for technology-assisted review. (https://arxiv.org/abs/1504.06868)
49. Hogan C, Bauer R, Brassil D. 2009 Human-aided computer cognition for e-discovery. In *Proc. of the 12th Int. Conf. on Artificial Intelligence and Law, Barcelona, Spain*, pp. 194–201. New York, NY: Association for Computing Machinery.

50. Savelka J, Westermann H, Benyekhlef K. 2021 Cross-domain generalization and knowledge transfer in transformers trained on legal data. (https://arxiv.org/abs/2112.07870)
51. Savelka J*et al.* 2021 Lex Rosetta: transfer of predictive models across languages, jurisdictions, and legal domains. In *Proc. of the Eighteenth Int. Conf. on Artificial Intelligence and Law, São Paulo, Brazil*, pp. 129–138. New York, NY: Association for Computing Machinery.
52. OpenAI. 2023 GPT-4 technical report.
53. Yu F, Quartey L, Schilder F. 2022 Legal prompting: teaching a language model to think like a lawyer. (doi:10.48550/arxiv.2212.01326)
54. Katz DM, Bommarito MJ, Gao S, Arredondo P. 2023 Gpt-4 passes the bar exam. *Available at SSRN 4389233*.
55. Bommarito J, Bommarito M, Katz DM, Katz J. 2023 GPT as knowledge worker: a zero-shot evaluation of (AI)CPA capabilities. (doi:10.48550/arxiv.2301.04408)
56. Sarkar R, Ojha AK, Megaro J, Mariano J, Herard V, McCrae JP. 2021 Few-shot and zero-shot approaches to legal text classification: a case study in the financial sector. In *Proc. of the Natural Legal Language Processing Workshop 2021*, pp. 102–106. Punta Cana, Dominican Republic: Association for Computational Linguistics. (doi:10.18653/v1/2021.nllp-1.10)
57. Savelka J. 2023 Unlocking practical applications in legal domain: evaluation of GPT for zero-shot semantic annotation of legal texts. (https://arxiv.org/abs/2305.04417)
58. Hamilton S. 2023 Blind judgement: agent-based supreme court modelling with GPT. (https://arxiv.org/abs/2301.05327)
59. Savelka J, Ashley K, Gray M, Westermann H, Xu H. 2023 Can GPT-4 support analysis of textual data in tasks requiring highly specialized domain expertise?. In *Proc. of the Sixth Workshop on Automated Semantic Analysis of Information in Legal Text* (*ASAIL 2023*), *Braga, Portugal*. Aachen, Germany: M. Jeusfeld c/o Redaktion Sun SITE.
60. Savelka J, Ashley K, Gray M, Westermann H, Xu H. 2023 Explaining legal concepts with augmented large language models (GPT-4). In *Proc. of the Workshop on Artificial Intelligence for Legislation* (*AI4Legs 2023*), *Braga, Portugal*. Aachen, Germany: M. Jeusfeld c/o Redaktion Sun SITE.
61. Savelka J, Ashley KD. 2023 The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts. *Front. Artif. Intell.* **6**, 1279794. (doi:10.3389/frai.2023.1279794)
62. Blair-Stanek A, Holzenberger N, Van Durme B. 2023 Can GPT-3 perform statutory reasoning? (https://arxiv.org/abs/2302.06100)
63. Nguyen HT, Goebel R, Toni F, Stathis K, Satoh K. 2023 How well do SOTA legal reasoning models support abductive reasoning? (https://arxiv.org/abs/2304.06912)
64. Janatian S, Westermann H, Tan J, Savelka J, Benyekhlef K. 2023 From text to structure: using large language models to support the development of legal expert systems. (https://arxiv.org/abs/2311.04911)
65. Tan J, Westermann H, Benyekhlef K. 2023 ChatGPT as an artificial lawyer?. In *Artificial Intelligence for Access to Justice (AI4AJ 2023), Braga, Portugal*. Aachen, Germany: M. Jeusfeld c/o Redaktion Sun SITE.
66. Westermann H, Savelka J, Benyekhlef K. 2023 LLMediator: GPT-4 assisted online dispute resolution. In *Artificial Intelligence for Access to Justice* (*AI4AJ 2023*), *Braga, Portugal*. Aachen, Germany: M. Jeusfeld c/o Redaktion Sun SITE.
67. Chalkidis I. 2023 ChatGPT may Pass the Bar Exam soon, but has a Long Way to Go for the LexGLUE benchmark. (https://arxiv.org/abs/2304.12202)
68. Bambauer J. 2014 Hassle. *Mich. L. Rev.* **113**, 461.
69. Savelka J, Ashley KD. 2018 Segmenting US court decisions into functional and issue specific parts. In *JURIX, Groningen, Netherlands*, pp. 111–120. Amsterdam, Netherlands: IOS Press.
70. Cohen J. 1960 A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**, 37–46. (doi:10.1177/001316446002000104)
71. Landis JR, Koch GG. 1977 The measurement of observer agreement for categorical data. *Biometrics* **33**, 159–174. (doi:10.2307/2529310)
72. Mackaay E, Robillard P. 1974 Predicting judicial decisions: the nearest neighbour rule and visual representation of case patterns. *Datenverarbeitung im Recht* **3**, 302–331. (doi:10.1515/9783112320594-012)