Construction and Evaluation of a High-Quality Corpus for Legal Intelligence Using Semiautomated Approaches

Haihua Chen, *Member, IEEE*, Lavinia F. Pieptea, and Junhua Ding

***Abstract*—A high-quality corpus is essential for building an effective legal intelligence system. The quality of a corpus includes both the quality of original data and the quality of its corresponding labeling. The major quality dimensions of a legal corpus include comprehensiveness, freshness, and correctness. However, building a comprehensive, correct, and fresh legal corpus is a grand challenge. In this article, we propose a semiautomated machine learning framework to address the challenge. We first created an initial corpus with 4937 instances that were manually labeled. Several strategies were implemented to assure its quality. The initial results showed that class imbalance and insufficiency of training data are the two major quality issues that negatively impacted the quality of the system that was built on the data. We experimented and compared three class-imbalance-handling techniques and found that the mixed-sampling method, which combines upsampling and downsampling, was the most effective way to address the issue. In order to address the insufficiency of training data, we experimented several machine learning methods for automated data augmentation including pseudolabeling, co-training, expectation-maximization, and generative adversarial network (GAN). The results showed that GAN with deep learning models achieved the best performance. Finally, ensemble learning of different classifiers was proposed and experimented with for the construction of a legal corpus, which achieves higher quality in comprehensiveness, freshness, and correctness compared to existing work. The semiautomated machine learning framework and the data quality evaluation method developed in this research can be used for data augmentation and quality evaluation of a large dataset as well as a reference for the selection of machine learning methods for data augmentation and generation. The machine learning models, the training data, and the legal corpus are published and publicly accessible at [Online]. Available: https://github.com/haihua0913/legalArgumentmining.**

***Index Terms*—BERT, data augmentation, data quality, deep learning, expectation-maximization (EM), generative adversarial network (GAN), legal argument, legal artificial intelligence (legal AI), machine learning corpus.**

Manuscript received February 1, 2022; accepted February 25, 2022. Date of publication March 25, 2022; date of current version June 2, 2022. This work was supported in part by the NSF under Grant 1852249 and in part by NSA under Grant H98230-20-1-0417. Associate Editor: R. Gao. *(Corresponding author: Junhua Ding.)*

Haihua Chen and Junhua Ding are with the Department of Information Science, University of North Texas, Denton, TX 76203 USA (e-mail: haihua.chen@unt.edu; junhua.ding@unt.edu).

Lavinia F. Pieptea is with the Department of Mathematics, University of North Texas, Denton, TX 76203 USA (e-mail: laviniaflorentinapieptea@my.unt.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TR.2022.3156126.

Digital Object Identifier 10.1109/TR.2022.3156126

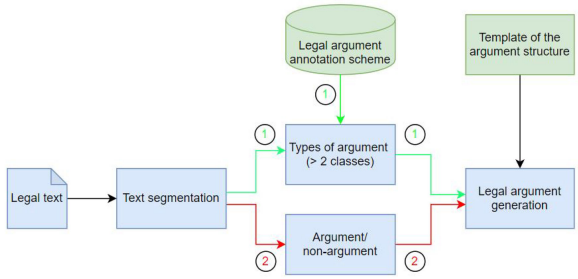


Fig. 1. Pipeline of legal argument mining.

I. Introduction

OVER the last few decades, legal artificial intelligence (legal AI) has developed rapidly with the practice of natural language processing (NLP) and machine learning in the legal domain. Legal AI tasks mainly include legal argument mining, legal judgment prediction, court view generation, legal entity recognition, legal question answering, and legal summarization [1]. As the core task of legal AI, legal argument mining aims to automatically extract units of argument or reasoning from natural language, legal documents usually court judgments with the goal of providing structured data for computational models of arguments and for reasoning engines [2]. As discussed by Palau and Moens [3], argument mining deserves more attention in the legal domain than in other areas since argumentation plays a central role in law practice. A legal argument recognition system can benefit legal professionals and provide a reliable reference to ordinary users.

Legal argument mining includes three steps, as shown in Fig. 1: 1) court judgment segmentation; 2) legal argument unit extraction; and 3) legal argument structure detection. Fig. 2 demonstrates the legal argument units of a court judgment. Extracting different legal argument types accurately is the fundamental, but most challenging task in legal argument mining.

Data volume and quality are the two crucial challenges for building an effective legal argument mining system [4]. However, creating a high-quality and sufficiently large legal corpus for machine learning is expensive. For example, it could take more than $2 000 000 to manually annotate some legal contracts by legal experts [5]. Alternative ways, such as the semiautomated method, should be developed. Semiautomated machine learning approaches that are built on a small amount of initial labeled data and large amount of unlabeled data

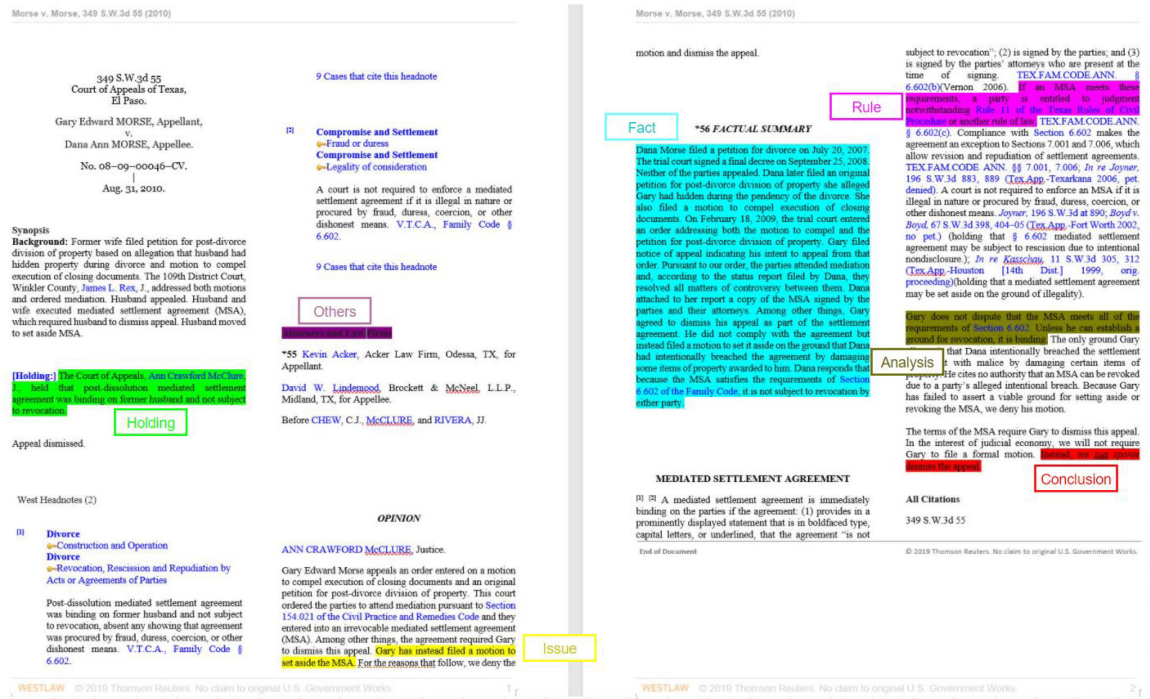


Fig. 2. Demonstration of different legal argument units of the court judgment "AA_edit_12 - Morse v Morse". Blue: fact, yellow: issue, green: holding, pink: rule, sage green: analysis, red: conclusion, and purple: others.

can be used to construct the needed legal corpus. Annotating high-quality initial data and designing effective automated algorithms for labeling unlabeled data are two determinants for the success of the semiautomated approaches.

As for the first determinant, basic steps need to be followed to ensure "science" of annotation [6]. Regarding the second determinant, semisupervised learning (SSL) is the most widely used approach to learn from both labeled and unlabeled data together [7]. Other frameworks, such as active learning (AL) [8], transfer learning (TL) [9], few-shot learning (FSL) [10], and generative adversarial networks (GANs) [11], have also become popular to learn from a limited number of labeled data. Notably, the expectation-maximization (EM) algorithm [12] has achieved the best performance on many tasks [13]–[15]. Recently, GAN-BERT, which extends the fine-tuning of BERT-like architectures with unlabeled data in a generative adversarial setting, has also been proven effective with a small amount of labeled data and a large amount of unlabeled data [16].

In this article, we first introduce an annotation scheme with six categories of legal arguments in U.S. legal cases. We conducted the human annotation experiment on 5066 sentences collected from Texas criminal cases and applied two strategies to ensure the quality of the annotation. However, the traditional interagreement of the labeling is not be sufficiently reliable for the quality evaluation since its correlation to the data quality cannot be directly interpreted [17]. Calculating the precision and recall of the annotated dataset for data quality evaluation is also infeasible, because creating the ground truth requires a lot of human effort and is costly in the legal domain. Therefore, we design a group of experiments to understand the impact of data quality to the performance of legal AI and the way for improving the data quality. The experimental results demonstrate the following:

1) Data with low annotation agreement does not significantly distort the model performance. It is still useful for training the machine learning models.
2) Data insufficiency and class imbalance in the data are two major quality issues of a legal corpus. Data augmentation and data resampling can be used to address the two issues, respectively.

The next research question we will discuss is the data augmentation of labeled data since manually labeling a high-quality legal corpus is unrealistic. SSL [13], co-training [18], EM [19], TL [20], and GAN-BERT [16] can be used for improving the machine learning performance when labeled data are insufficient and sufficient amount of unlabeled data are available. However, those techniques are not always effective and their effectiveness has rarely been rigorously validated in the legal domain. Based on the labeled legal corpus we constructed, we explore and compare multiple techniques (i.e., SSL, co-training, EM, TL, and GAN-BERT) for data augmentation. Our experimental results show that GAN-BERT achieves the highest performance, improving the best supervised machine learning models by 3% regarding the F1-score.

The third research question is on the effect of different learning parameters, such as amount of data and unlabeled data, and confidence score on the performance of the data augmentation. Each data augmentation technique has advantages and disadvantages under different parameter settings. Therefore, this research designed a series of experiments to explore and identify the best practices for the application of data augmentation techniques.

The main contributions of this research are summarized as follows:

1) We propose an annotation scheme for better classifying legal arguments and design a human annotation procedure to assure the interagreement of the annotation. The annotation scheme includes six types of legal arguments and its results were evaluated on 5060 sentences extracted from Texas criminal cases. The quality of the annotated legal corpus was further evaluated on its comprehensiveness, freshness, and accuracy. The annotation scheme and the annotated corpus provide needed resources for other researchers to conduct research in legal intelligence. The corpus is publicly available on request.
2) We develop an approach for evaluating the quality of a legal corpus and design a series of experiments to quantitatively evaluate it. The evaluation results provide the guidance for data quality improvement.
3) We compare the effectiveness of using pseudolabeling, cotraining, EM, and GAN in data augmentation for building a legal corpus. The experimental results show that GAN with BERT achieves the best performance. We also finetune the parameters for different data augmentation techniques and draw the best practice of implementing these algorithms for automated labeling. The results provide a reference for other researchers to choose the machine learning method for data augmentation.

The rest of this article is organized as follows. Section II presents the related work regarding legal argument mining, legal text classification algorithms, data quality assurance for machine learning, and approaches for data augmentation. Section III introduces a legal argument classification scheme and the annotation experiments for creating a corpus for legal intelligence. In Section IV, we conduct a systematic evaluation of the data quality of the corpus constructed earlier. Based on the evaluation results, Section V presents experiments with several widely used machine learning methods, including pseudolabeling, cotraining, EM, and GAN, for automated data augmentation. Section VI describes the application and the general procedure of the data augmentation techniques. Finally, Section VII concludes this article.

II. Related Works

In this article, we focus on developing a high-quality machine learning dataset for legal argument mining. Several areas are closely relevant to this research, which are legal argument identification, text classification algorithms, data quality evaluation, and data quality assurance. We also explore different approaches for augmenting training datasets.

A. Legal Argument Mining

Argument mining is the automatic identification and extraction of the structure of inference and reasoning expressed as arguments presented in natural language [2]. Argument mining mainly initiates in the legal domain, but soon becomes an essential task in other domains, such as education, web-based content, political debates, and speeches [21]. With the development of NLP and deep learning, legal argument mining has recently received increasing attention. Different legal argument mining approaches have been proposed to detect premises, claims, and argumentation schemes in judgments to ease the work of judges and law scholars in identifying similarities and differences among different judgments, the arguments proposed therein, and the outcome of the cases [21]. Fig. 1 shows the steps of legal argument mining.

Given a legal text corpus, usually from the European Court of Human Rights (ECHR) judgments [22], [23], U.S. Court of Federal reported cases [24], Chinese legal documents [25], and Japanese judgment documents [26], text segmentation is first used to extract the fragments of text (section level or sentence level) from the original legal document [2]. Second, a legal argument annotation scheme should be defined to classify the text fragments into different types of arguments. Research in this area divides this step into two substeps: 1) argument/nonargument identification [3], [27]; and 2) argument type identification. The former is a binary classification task, while the latter is a multiclassification task. However, most of the existing studies fall into the binary classification task, then directly move to the legal argumentation process based on a predefined argument structure [3], [26], [28], [29], as routine 2 shown in Fig. 1. Although Ashley [30] defined six categories of a legal argument: 1) rule and legal concepts; 2) standard of proof; 3) support/attack relation; 4) authority; 5) attribution information; and 6) plausibility, the reliability and practical value were reduced due to lacking an annotation experiment to evaluate the scheme. Le *et al.* [31] also proposed three categories of legal argument labeling: 1) fact description; 2) court view; and 3) penalty result. Nevertheless, the scheme cannot be considered comprehensive.

The structure of a legal argumentation is complicated, rather than a simple accumulation and combination of arguments, premises, and conclusions. For example, a fact is usually followed by another fact or an issue, while a court analysis or legal rule might follow an issue. In this situation, a fine-grained and reliable argument annotation scheme benefits the construction of high-quality legal argument dataset for legal argumentation and is useful for legal text summarization, legal information retrieval, and legal judgment prediction. It formulates one of the purposes of this research.

B. Algorithms for Legal Text Classification

Legal argument identification is usually regarded as a legal text classification task. Traditional machine learning [32] and deep learning models [33] have been applied for the task. Compared to text classification in the general domain, legal text classification is more challenging.

1) Creating a high-quality legal dataset with sufficient amount of training samples is costly.
2) Language models pretrained in general domain texts can hardly work well due to the specific words or intelligible slang by domain experts contained in legal text.

There are efforts on exploring machine learning models for legal text classification. Moens *et al.* [27] used the multinomial naïve Bayes (NB) classifier and the maximum entropy model

for detecting arguments in legal text according to the rhetorical types and built visualizations for convenient access and search. In another study, an argument-based machine learning, which makes use of the justifications of decisions, was applied to extract rules for explaining the data in a field of law [34]. Hachey and Grover [35] presented legal text classification according to a rhetorical scheme indicating a sentence's contribution to the overall argumentative structure of the legal judgments using four machine learning algorithms, including C4.5, NB, winnow, and support vector machine (SVM) . However, semisupervised classification algorithms have outperformed the supervised classification algorithms on text classification tasks when labeled samples are insufficient and a large amount of unlabeled data are available [36]. Therefore, semisupervised algorithms, such as an EM algorithm, have also been applied for catchphrase classification in legal text documents [36]. Meanwhile, machine learning tools, such as the IBM Watson system, have also been used to extract argumentation-relevant information from legal decision documents and build new arguments based on the extracted information [37].

Deep learning models perform better than traditional machine learning models in multiple NLP tasks, such as recommendation [38], mining software repositories [39], and text classification [32]. They are also popular among the legal AI community [1]. The deep learning models proposed for legal text classification can be roughly summarized into five categories: 1) CNN-based; 2) RNN-based; 3) GNN-based; 4) hybrid; and 5) transformer-based models. For example, Undavia [40] compared different word embeddings combined with CNN and RNN for document classification of legal court opinions and concluded that CNN with Word2vec achieved the highest performance. Neural models, such as BIGRU with self-attention (BiGRU-Att), hierarchical attention network, label-wise attention network, BERT, and HIER-BERT, have been proposed for judgment prediction on the ECHR dataset [41]. With the applications of knowledge graphs in domain-specific NLP tasks, GNN models are developed for learning over knowledge graphs. Aligned with the research direction, Li *et al.* [42] combined legal ontology with graph LSTM for text classification of Chinese legal documents. More recently, transformer models, such as BERT, Roberta, DistilBERT, and XLNet, which were fine-tuned for large-scale legal text classification, achieved state-of-the-art performance on the JRC-Acquis and EURLEX57 K datasets [43].

C. Techniques for Data Quality Evaluation

The approaches for data quality evaluation can be divided into two categories: 1) quantitative methods; and 2) qualitative methods. Statistical analysis, experimental study, and empirical evaluation were commonly used quantitative methods. Techniques for statistical analysis include descriptive statistics, plot chart, bubble scatter chart, confidence intervals, correlation relationship, the Chi-square test, and the Mann–Whitney test [44]. For example, continuous data were usually analyzed by the value of percentage, particularly regarding the data about completeness and accuracy, to ascertain whether they reached the quality standards [44]. A set of factors are identified, and a group of experiments are usually designed to validate the data quality for the experimental study. For example, the data collection process assessment activities were initiated by identification of the causes of poor data quality or were considered as a component of the evaluation of the effectiveness of the system [44]. Machine learning and deep learning have also been used to validate data quality. Qualitative methods, include review of publications and documentation, interview with key informants and field observations. Table I summarizes the recent data evaluation techniques.

As given in Table I, the dataset for machine learning and deep learning systems is fairly large, it is usually unrealistic to evaluate the quality using qualitative methods. Therefore, quantitative methods are used in this research. Different machine learning algorithms, such as TL, reinforcement learning, deep neural network, AL, and others, are selected for experimental study for different purposes and tasks. These studies also demonstrate that data quality can be quantitatively evaluated, and the evaluation results can guide practitioners to develop more reliable and higher performance machine learning systems.

D. Data Quality Assurance and Improvement for Machine Learning Systems

Since data quality largely determines the performance, fairness, robustness, safety, and scalability of ML and AI systems, which are built on the data [20], [53], data quality assurance is an essential requirement to ensure the quality of the systems [54].

Experiments have been conducted to explore the effects of annotation quality on model performance [55], [56]. Hsueh *et al.* [57] identified three criteria to select high-quality annotations: 1) noise level; 2) sentiment ambiguity; and 3) lexical uncertainty. Empirical study showed that the three criteria can be used to improve annotation data quality [57]. Taking the genome annotation work as an example, Huang *et al.* [58] ranked 17 data quality dimensions and 17 data-quality skills based on their importance in annotation data quality assurance. Intuitively, annotation quality can be controlled during the task assigning stage, yet existing studies have rarely discussed this. Recently, an efficient annotation framework that combines a stochastic transitivity model and an effective sampling strategy was used to infer high-quality labels with a low effort from crowdsourced pairwise judgments [59]. The approach outperformed existing annotation procedures when compiling the *Webis Argument Quality Corpus* [59].

Generally, a robust ML system has the capability to handle noisy data. Systematic experiments showed that label noise in the minority class is not as harmful to a classifier as noise in the majority class in the imbalanced datasets [60]. But the question is about what percentages of label noise can affect the ML performance. Investigation on two ML-based network intrusion detection systems that were implemented with C4.5 and NB algorithms on data of varying label errors proved that both algorithms are quite robust when subjected to training data with an increasing amount of label errors, C4.5 maintained a high level of accuracy, which was close to 90%, but its accuracy dropped

TABLE I
SUMMARY OF THE RECENT DATA EVALUATION TECHNIQUES: DATA QUALITY EVALUATION DIMENSION(S), EVALUATION METHOD, DATA TYPE, DATA QUALITY EVALUATION TECHNIQUES, AND FINDINGS

Dimension(s)	Evaluation method	Data type	Techniques/ algorithms	Findings	Reference
Duplication	Quantitative: Experimental study	Text	Transfer learning	A rigorous evaluation of data quality is necessary for guiding the quality improvement of machine learning	Chen et al., 2021 [10]
Data valuation	Quantitative: Experimental study	Tabular, image, text	Reinforcement learning	The proposed meta learning framework can rank the data values for the training dataset efficiently and effectively	Yoon et al, 2020 [47]
Relevance	Quantitative: Empirical evaluation	Image	Deep network embedding	Relevance can be evaluated from different perspectives, such as the quantity of relevant data and the degree of semantic similarity	Liu et al, 2020 [48]
Data bias	Quantitative: Experimental study	Text	Active learning	The proposed generic formula for Data Quality Index (DQI) can help dataset creators create datasets free of unwanted biases	Mishra et al., 2020 [49]
Accuracy	Quantitative: Experimental study	Knowledge graph	Cluster sampling with unequal probability theory	The proposed framework provides quality accuracy evaluation with strong statistical guarantee while minimizing human efforts on both static and evolving KG	Gao et al., 2019 [50]
Anomaly in data	Quantitative: Empirical evaluation	Multiple data types	Fuzz-testing	Data validation can benefit ML from several aspects: early detection of errors, model-quality wins from using better data, savings in engineering hours to debug problems, and a shift towards data-centric workflows in model development	Breck et al., 2019 [51]
Accuracy, completeness, and accessibility	Quantitative: Statistical analysis	Webpages	–	The quality, completeness and accessibility of online health information regarding fibromyalgia was poor	Basavakumar et al., 2019 [52]
Completeness	Quantitative: Empirical evaluation	Knowledge graph	Profiling Wikidata (ProWD)	ProWD is effective for detecting anomalies, analyzing knowledge imbalances, and checking data compliance	Wisesa et al., 2019 [53]
Completeness and Consistency	Quantitative and qualitative methods	Knowledge graph	Traditional machine learning	Both completeness and consistency can be evaluated using data driven approaches	Rashid et al., 2019 [54]
Data volume (Insufficient training data)	Quantitative: Experimental study	Text	Active semi-supervised learning	Active semi-supervised learning can be used to create meaningful data representations and simultaneously reduce the burden and cost of human annotations	Lourentzou, 2019 [7]

abruptly when there were 45% of label errors. In contrast, the accuracy of NB decreased gradually from 92.44% on clean data to 75.71% with 50% of errors [61]. By contrast, unsupervised learning, such as K-means achieves high accuracy even when the rate of mislabeled data is high since the labels do not participate in the training [61]. Therefore, unsupervised learning should be considered when a dataset is of poor quality.

Class imbalance in datasets is another common data quality problem for real-world classification tasks [62]. The class imbalance issue usually causes a standard classifier biased towards the common classes and performs poorly on rare classes [62]. There are two approaches to handle the unbalanced data: 1) resampling techniques and; 2) algorithmic ensemble techniques. Resampling techniques, which resample the original data to produce the balanced classes, are the most widely used strategies [63]. There are five ways to resample data: 1) downsampling; 2) random oversampling; 3) cluster-based oversampling; 4) the synthetic minority oversampling technique (SMOTE); and 5) the modified SMOTE. A comparative analysis of these techniques has been performed on handling the issue of imbalanced data [62], [63].

Today, insufficiency of training data has become a common data quality issue since deep learning models require large-scale data, especially labeled data for training. A standard solution is focusing on creating annotations or circumventing the need for labels by automatically labeling datasets or utilizing unlabeled data [4]. Techniques, such as SSL [7], AL [8], TL [9], FSL [10], and GANs [11], can be used in the process [64]. However, traditional bootstrapping approaches often negatively affect the NLP performance due to the addition of falsely labeled instances. To improve the quality of automatic labeling, Lourentzou [4] introduced a calibration of semisupervised AL, where the confidence of the classifier was weighted by an auxiliary neural model. The strategy could remove incorrectly labeled instances and dynamically adjust the number of proxy labels including in each iteration [4].

E. Approaches for Augmentation of Labeled Data

Different learning strategies, such as SSL, AL, TL, and FSL, have been proposed for producing labeled data [65].

1) Semisupervised Learning: SSL algorithms utilize unlabeled data to improve classification performance [13]. A plethora of learning methods exist for SSL [7], but two major classes are most widely used: 1) co-training and; 2) pseudolabeling. Co-training is an extension of self-training to multiple classifiers, which are iteratively retrained on each other's most confident predictions [7]. In pseudolabeling, a classifier is trained on the labeled data and updated with the most confident predictions of the previous classifiers on an unlabeled data [7]. Tariq *et al.* [18] proposed a co-training technique by incorporating the discriminative power of the widely used classifiers, namely random forest (RF), SVM, and NB, for mental illnesses classification.

Li and Yang [66] applied pseudolabeling with an NB classifier, called the PL-DNB model. Specifically, they employed the EM algorithm to train PL-DNB in a semisupervised manner so that all the documents with and without pseudolabels were used for creating a classifier. Second, they iteratively updated the pseudolabels with highly acceptable confidence [66]. Alternatively, Lee *et al.* [67] combined pseudolabeling with deep neural networks. The EM framework is usually embedded in SSL algorithms for learning from labeled and unlabeled documents. An another effective semisupervised method is implemented with GAN [68], where a "generator" is trained to produce samples by resembling some data distribution. The training process "adversarially" depends on a "discriminator," which is instead trained to distinguish samples of the generator from the real instances [16]. Semisupervised GAN uses labeled data to train the discriminator, while the unlabeled examples, as well as the ones automatically generated, improve its inner representations [68]. Recently, neural SSL for text classification under large-scale pretrained language models has emerged [16], [69], [70].

2) Active Learning: AL aims to find the most efficient way to query labels and learn a classifier with minimal human supervision [4]. It interactively assigns certain specific data points to users for annotation by identifying the best data to annotate next [4]. Several strategies can be used to select the best candidate by uncertainty sampling, density-weighted uncertainty sampling, diversity, QUIRE, and Bayesian methods [4]. The unlabeled data are from either an extensive pre-existing collection or streaming data. AL has been combined with traditional machine learning models and neural models for text classification [4], [71].

3) Transfer Learning: TL leverages knowledge from a source domain to improve the learning performance or minimize the number of labeled examples required in a target domain [9]. TL has been applied for both computer vision and NLP for automatically labeling data [72], [73]. TL shows tremendous capabilities for NLP tasks in specific domains and, is especially promising for low resource languages. For instance, a simple model with only a pretrained BERT-based model, a linear layer, softmax, and Viterbi decoding achieves state-of-the-art performance on semantic role labeling (SRL) in Portuguese [72]. In addition, cross-lingual TL using multilingual pretrained models and TL from dependency parsing can also improve the performance of SRL [72]. In the legal domain, Chen *et al.* [73] applied the weight-sharing mechanism of TL to utilize the data with high frequency to model the projection between fact and law articles.

4) Few-Shot Learning: FSL rapidly generalizes prior knowledge to new tasks containing only a few samples with supervised information [10]. FSL has drawn much recent attention since it can reduce data-gathering efforts and computational cost [10]. As argued by Yin *et al.* [74], few-shot textual entailment could be a promising attempt for universal NLP when we cannot guarantee the accessibility of rich annotations. There are following three methods in FSL for augmenting training datasets.

1) Transforming samples from the training set.
2) Transforming samples from a weak labeled or unlabeled dataset.
3) Transforming samples from similar datasets.

For instance, Schick *et al.* [75] proposed an approach for identifying words that can serve as proxies for labels given small amounts of training data for few-shot text classification. This approach relieved the need for expert knowledge. Large language models are also regarded as few-shot learners [76], since both of them can transform samples from similar datasets. Existing studies typically treated the number of training samples in the range of ten to 100 as an FSL task [76], yet it remains an open research question in terms of specific tasks.

III. High-Quality Legal Argument Corpus Annotation

A. Data Acquisition and Preparation

The initial data in this research were collected from the Harvard Law Library case law corpus [77], which includes 360 years of United States case law. The corpus contains 6 708 785 unique cases, including legal decisions from all state and federal courts in total from 582 reporters. The metadata and the full texts of all the cases are freely available[1]. This corpus was first released in 2018, and it has become ever more popular among the legal AI community [78], [79]. Compared to the unstructured data from other resources, the case law corpus is semistructured and of higher quality. In this research, we created a subset from the Harvard Law Library case law corpus, which includes all published criminal cases from the year 1840 to 2018 in the state of Texas with 27 712 cases in total. All the metadata and full texts were stored in JSON format. We extracted the full-text fields for legal argument corpus construction.

Since we conducted the annotation at the sentence level, the first step was to split each legal case into sentences and remove invalid sentences. We used LexNLP, an open-source Python package focused on NLP and machine learning for legal and regulatory text [80], to perform the task. However, some sentences were incorrectly split, and others were incomplete sentences, such as section titles. These sentences are called invalid sentences, and can reduce both annotation efficiency and performance. We manually annotated 1008 sentences, and trained a binary classifier using machine learning algorithms to remove the invalid sentences. We identified four features to train the classifier: 1) number of words/number of symbols; 2) number of letters/number of symbols; 3) average word length; and 4) length of the sentences. The test accuracy on logistic regression (LR), decision tree (DT), SVM, NB, K-nearest neighbors (KNN), RF, and XGBoost (XGB) is 0.9158, 0.8267, 0.8514, 0.8069, 0.8762, 0.8713, and 0.8713, respectively. Therefore, we chose LR for validating sentence classification. Finally, we produced 542 763 validated sentences, which served as the pool for annotation and data augmentation.

B. Annotation Scheme for Legal Argument

We create an annotation scheme with six categories, as given in Table II, for legal argument: 1) fact; 2) issue; 3) rule/law/holding; 4) analysis; 5) conclusion/opinion/answer; and 6) others. The first five categories are adopted from [81], which are the essential components in writing a legal report. We

[1][Online] Available: https://case.law/

TABLE II
PROPOSED ANNOTATION SCHEME FOR LEGAL ARGUMENT. A MORE DETAILED DESCRIPTION OF THE ANNOTATION SCHEME AND MORE EXAMPLES FOR EACH LABEL CAN BE FOUND AT THE PROJECT SITE IN GITHUB

Label	Description	Example	Annotation Guideline
Fact	Any fact that is pertinent to the case. This includes testimony, statements of record, case history, and anything else that is a fact that helps establish the foundation of the case for the court to build its analysis and judgment on.	*Now you say that the premises were controlled by Report Walton on that date and you know that?*	This is a very broad category. Anything that "sets the stage," as it were, should be labeled as a fact. Factual sentences do not include any synthesis or reasoning (that would fall under analysis); rather, they simply state events or matters of record.
Issue	Any issue or question that the court must decide. This includes the overall issue of the case as well as any sub-issues that are raised in the case.	*Appellant's sole contention on appeal is that the evidence is insufficient to sustain the conviction.*	This tends to be a pretty narrow category. Any sentence related to a question for the court to resolve should be labeled as an issue. Note that a lot of the issue sentences could be read as a fact, as any issue is inherently part of the factual history of the court case.
Rule/ law/ holding	Any statement of or reference to a rule, law, or holding. This includes sentences referencing a rule, law, or holding and then using it to reason through some point.	*Reliance is handed upon Toombs v. 21, 317 S.W. 2d 737, as authority for reversal of cases where the state's testimony was adduced by only one witness.*	Generally speaking, any time a reference is made to a rule, law, or holding, that sentence should be given this label. These sentences could be read as a fact, since whenever a law is quoted, for example, it is a fact that the law states that quotation.
Analysis	Any sentence that synthesizes information to further the court's reasoning. This includes sentences that refer to the facts of the case and then use them to push forward towards a resolution.	*Had the new Rules of Appellate Procedure been in effect during the pendency of this appeal, sanctions against the responsible attorneys would have been appropriate.*	Analysis sentences tend to move the court through the case from the facts towards the conclusion, so there is often a logical progression stringing analysis sentences together. A sentence that references a rule, law, or holding and then analyzes it in the context of the current case should be labeled Rule/Law/Holding, not as an analysis sentence.
Conclusion/ opinion/ answer	Any sentence that effectively resolves an issue facing the court.	*The trial court erred in submitting to the jury the issue of rape by threats.*	Conclusions tend to be short and straight to the point, either agreeing or disagreeing with some argument. As there are sub-issues, there are also sub-conclusions that conclude a specific sub-issue being discussed by the court.
Others	(1) Any sentence or phrase that does not fit the other labels in terms of its content. This includes section headings and others. (2) Sentences that were not split correctly.	*Ex.1: We have carefully reviewed both of these cases. Ex.2: 21, 317 S.W. 2d 737, as authority for reversal of cases where the state's testimony was adduced by only one witness.*	Sentences that do not add any information to the case and thus do not fall into any of the other labels. Section headings are in this category as they are neither complete sentences nor useful when building the logic of the court case.

TABLE III
SUMMARY OF THE DATASET

Category	Kappas>0.5	Kappas<0.5	Total
Fact (F)	1777	805	2582
Issue (I)	224	113	337
Rule/Law/Holding (R)	224	114	338
Analysis (A)	516	253	769
Conclusion (C)	173	172	345
Others (O)	558	131	689
Total	3472	1588	5060

create a new label ("other") to annotate sentences that do not belong to any of the categories. "Facts, issue, rules, analysis, and conclusion" together are abbreviated as FIRAC and are used to organize an argument. It is commonly used in the legal field [81]. Some of the components have also been reused in other studies [30], [31], [82].

According to the White's definition [81], facts are unemotional, nonjudgmental, and objective observations of the state of reality. Parties can agree or disagree with the facts. If the parties do not agree on what the facts are, then an issue of fact is raised [81]. The issue is the question being resolved, that is, what the parties disagree about. There are three types of issues in the legal area: 1) value/overall/legal consequences issues; 2) factual issues; and 3) legal issues [81]. Rule/law/holding is a summary of the law used by the author of the argument in support of the conclusion [81]. The analysis contains the reasons in support of the conclusion/answer/opinion. The analysis section must contain the elements of the law and the facts supporting each element of the law or a statement that no facts exist to support a particular element [81]. The conclusion/opinion/answer is the answer to the issue raised in the scenario [81]. A more detailed explanation of each category with examples and the annotation guideline can be found in Table II.

C. Human Annotation Experiment

1) Annotation Procedure: We use doccano[2] for the human annotation on legal arguments. Doccano is an open-source text annotation tool that provides annotation features for text classification, sequence labeling, and sequence-to-sequence tasks. We recruit six undergraduates and three graduates who are proficient in English for the annotation. To improve the annotation quality, we require the annotators to read legal case reports and other materials to be familiar with the rules of writing a legal brief. Meanwhile, we invite a law librarian to train them in the appropriate legal knowledge for the annotation and draft the guideline. We then run the preannotation on 30 cases to optimize the guideline.

For the formal annotation, we equally separate the students into three groups. Each group is required to annotate around 500 sentences a week. Students in the same group annotate the same sentences independently. The majority vote is used to decide the final label of a sentence. If the label of a sentence cannot

[2][Online] Available: https://github.com/doccano/doccano

be confirmed based on the three annotators, another annotator will label the sentence. Finally, we produced 5066 annotated sentences in total.

2) Interagreement Evaluation: We measure agreement in kappa [83] with the following formula:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

where *P(A)* is the observed agreement of the annotators and *P(E)* is the expected agreement. Kappa ranges between −1 and 1. K=0 means the agreement is only as expected by chance. Generally, kappas of 0.8 are considered stable, and kappas of 0.69 are marginally stable, according to the strictest scheme applied in the field [83]. However, for some specific domains, such as biomedical, kappas of over 0.6 are considered trustworthy [6].

3) Quality Assurance of the Annotation: We implement two strategies to ensure the quality of the annotation. First, we detect and remove anomalous annotations. We consider the annotations of an annotator as an anomaly if the kappas with the other two annotators are much lower than the kappas between the two annotators. In this scenario, a fourth annotator will reannotate the same data, and the results will be evaluated to determine whether they can be used for the majority vote. Second, considering the difficulties of the legal domain and the challenge of the legal argument annotation task, we set the threshold of kappas as 0.5. If the paired kappas among the three annotators are below 0.5, we separate the data from the final results. The reason we reserve the low kappa annotations is that the kappa agreement has following limitations.

a) It is designed to take account of the possibility of guessing, but the assumptions it makes about rater independence and other factors are not well-supported, and thus, it may lower the estimate of agreement excessively.
b) It cannot be directly interpreted, and thus, it has become common for researchers to accept low kappa values (abbreviated as "kappas") in their inter-rater reliability studies [17].

4) Results: We obtained 3472 annotated sentences whose kappas are higher than 0.5 together with 1588 sentences whose kappas are below 0.5. The detailed statistics of samples in each category are given in Table III. We can see that the dataset suffers from the class imbalance issue. In the following section, we designed experiments to validate whether kappas are reliable to justify the annotation quality and evaluated how low kappas data will affect the machine learning performance. In addition, we explored the strategies to handle class imbalance issues in the legal argument identification.

IV. Legal Argument Identification and Data Quality Evaluation

A. Algorithms

In the initial legal argument identification experiments, we use tf-idf features-based machine learning models as baselines and compare them with the BERT-based deep learning model. In preexperiments, we find that SVM, RF, and LightGBM outperform the other machine learning models, including NB, KNN, DT, stochastic gradient boosting, and XGB; thus, we only report the results of the three models in this article. In this section, we introduced how the machine learning models and BERT are used for legal argument identification.

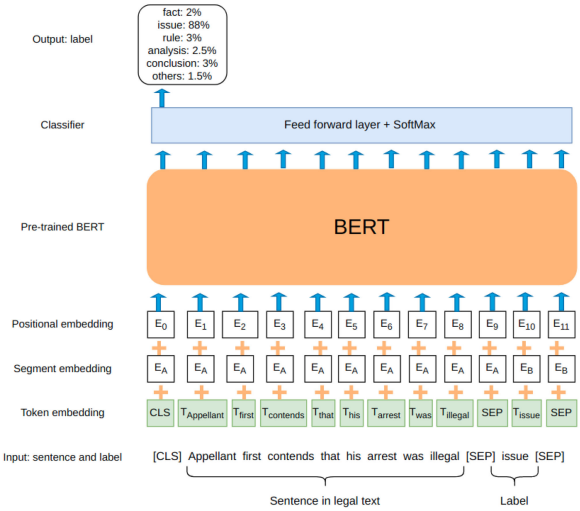


Fig. 3. BERT model for legal argument classification.

1) Support Vector Machine: SVM is a supervised machine learning algorithm, which separates the sentence vectors into different categories based on kernel methods [71]. The SVM classifier has been popular when the amount of training data is limited. The input of SVM is the sentence vector represented by tf-idf. We use a polynomial kernel with three degrees and 1.0 regularization to train the model.

2) Random Forest: According to a comparison study on 179 classifiers with 121 datasets, RF achieved the best performance [84]. RF is also a high performer in text classification since it mitigates the inherent challenges involved in textual data, such as high dimensionality, sparsity, and noisy feature space. In this article, we extract the tf-idf features and train trees on the random subsets of the features. The bagging algorithm is applied to produce random samples for the training.

3) LightGBM: LightGBM [85] has many advantages, including sparse optimization, parallel training, multiple loss functions, regularization, bagging, and early stopping. Meanwhile, instead of growing a tree level-wise—row by row—as most other DT-based algorithms do, LightGBM selects the leaf that will yield the most significant decrease in loss. It leverages the gradient-based one-side sampling technique to reduce the number of data instances and the exclusive feature bundling technique to reduce the number of features [85]. It has proven to be a highly efficient and effective text classification algorithm.

4) BERT: BERT is the state of the art for multiple NLP tasks, including text classification [86]. BERT effectively learns global semantic representation and significantly boosts NLP tasks. It generally uses unsupervised methods to mine semantic knowledge automatically, and then construct pretraining targets so that machines can learn to understand semantics [32]. The architecture of BERT for legal argument classification is shown in Fig. 3. Given a legal sentence together with its label as an input sequence, it will be converted using the pretrained BERT model

TABLE IV
DATASETS FOR TESTING THE EFFECT OF LOW KAPPAS DATA ON THE MACHINE LEARNING PERFORMANCE

Dataset	F	I	R	A	C	O	Total
Kappas>0.5	1777	224	224	516	173	558	3472
Train_mix_1	1769	218	240	520	179	550	3746
Train_mix_2	1770	218	240	520	179	549	3746
Train_mix_3	1770	218	240	520	179	549	3746
Train_mix_4	1769	218	240	520	179	550	3746
Train_mix_5	1769	218	240	520	179	550	3746
Train_mix_6	1769	218	240	520	179	549	3746
Train_mix_7	1769	218	240	520	179	549	3746
Train_mix_8	1769	218	240	520	179	549	3746
Train_mix_9	1769	218	240	520	179	549	3746
Train_mix_10	1769	218	240	520	179	550	3746
All_data	2253	269	332	677	225	660	4416
Test	267	34	34	77	26	83	521

(BERT-base-uncased, 12-layer, 768-hidden, 12-heads, and 110-M parameters). We then fine-tune the model, and add the softmax classifier to the top of BERT for legal text classification. The output is the probabilities of a legal sentence belonging to each category.

B. Evaluation Metrics

We use accuracy, precision, recall, and F1-score as metrics to evaluate the performance on each category since they are the most used evaluation metrics for text classification [32]. For the overall performance, we use weighted-average precision, recall, and F1-score. Each class's contribution to the average is weighted by its size, which is more appropriate for imbalanced data [87]. They are calculated with the following formulas:

$$\text{precision} = \frac{\sum_{i\in L} \text{precision}_i \times N_i}{\sum_{i\in L} N_i} \tag{1}$$

$$\text{recall} = \frac{\sum_{i\in L} \text{recall}_i \times N_i}{\sum_{i\in L} N_i} \tag{2}$$

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

where L is the label set and N is the total number of samples of each category. Due to space limitations, results of some metrics might not be listed. We report the F1-score in most of the experiments, since it takes into account both recall and precision and represents the accuracy.

C. Annotation Quality Evaluation

As mentioned above, it has become common for researchers to accept low kappa values in their inter-rater reliability studies [17]. Therefore, we generate 12 training datasets and one test dataset to evaluate how low kappas data will affect the machine learning performance. Ten of the 12 datasets are mixed by annotations from both high kappas and low kappas data, and the other two are the high kappas data and the combination of all the 11 datasets, respectively. The statistics of the datasets are given in Table IV. The models trained on the 12 datasets are evaluated on the same test dataset.

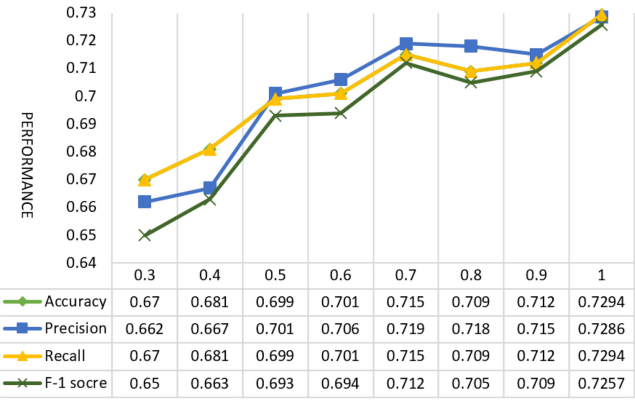


Fig. 4. Performance of the BERT model is evaluated by using different portions of data for the training. We randomly sample the portions of data and repeat ten-times running, and all the standard deviations regarding the accuracy, precision, recall, and F1-score are below 0.05.

Table V presents the experimental results of different datasets using SVM, RF, and LightGBM, while Table VI presents results using BERT. We draw the following observations.

1) LightGBM achieves the highest performance among all the traditional machine learning models, but the difference between SVM, RF, and LightGBM is not significant.
2) The BERT-based deep learning model outperforms other machine learning algorithms on legal argument identification.
3) Data with a low kappa value does not significantly reduce the model performance. On the contrary, both the traditional machine learning models and BERT benefit from combining all the annotated data with high kappa values and low kappa values.

The abovementioned experimental results demonstrate that the kappa agreement might not be a reliable measurement for annotation quality evaluation in some cases. Therefore, it is necessary to design new experiments to understanding the measurement of the data quality. In addition, a certain level of label errors (i.e., 30% or less label errors) is not harmful to a robust machine learning model, as proved by Lauria and Tayi [61], and it can be helpful for the model to learn effective features. Our experiments indicate that including all the annotated data can enhance the learning of both machine learning and deep learning models. Therefore, we will use the combined dataset for all the experiments in the rest of this article.

D. Data Sufficiency Evaluation

To validate whether the existing data are sufficient to train a supervised learning model, we use different data portions to train the model and observe whether the model performance increases with the increasing amount of data. The result is shown in Fig. 4.

In Fig. 4, the *X*-axis lists the portions of training data, while the *Y*-axis is the performance on the F1 score. The model performance increases as the data increases, indicating that the data are insufficient for training a good machine learning model. Data augmentation is needed to enhance the model performance (see Section V).

TABLE V
PERFORMANCE ON DIFFERENT TRAINING DATASETS USING MACHINE LEARNING MODELS, INCLUDING SVM, RF, AND LIGHTGBM REGARDING WEIGHTED ACCURACY, PRECISION, RECALL, AND F1-SCORE

	SVM				RF				LightGBM			
Dataset	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Kappas>0.5	0.6065	0.6228	0.6065	0.5371	0.6084	0.6118	0.6084	0.582	0.5931	0.5709	0.5931	0.5720
Train_mix_1	0.6238	0.6560	0.6238	0.5615	0.6180	0.6224	0.6180	0.5886	0.5969	0.5806	0.5969	0.5776
Train_mix_2	0.6123	0.6231	0.6123	0.5461	0.6200	0.6155	0.6200	0.6937	0.5835	0.5643	0.5835	0.5665
Train_mix_3	0.6104	0.6253	0.6104	0.5414	0.6104	0.6053	0.6104	0.5745	0.5893	0.5726	0.5893	0.5731
Train_mix_4	0.5988	0.6096	0.5988	0.5270	0.6180	0.6151	0.6180	0.5872	0.5988	0.5863	0.5988	0.5798
Train_mix_5	0.6065	0.6173	0.6065	0.5340	0.6123	0.6136	0.6123	0.5834	0.5931	0.5764	0.5931	0.5746
Train_mix_6	0.6104	0.6254	0.6104	0.5403	0.6219	0.6322	0.6219	0.5977	0.6161	0.6016	0.6161	0.5988
Train_mix_7	0.6104	0.6371	0.6104	0.5426	0.6065	0.6135	0.6065	0.5791	0.6065	0.5901	0.6065	0.5902
Train_mix_8	0.6084	0.6279	0.6084	0.5372	0.6027	0.6079	0.6027	0.5682	0.5950	0.5758	0.5950	0.5773
Train_mix_9	0.6104	0.6231	0.6104	0.5434	0.6142	0.6109	0.6142	0.5831	0.5950	0.5803	0.5950	0.5764
Train_mix_10	0.6046	0.6140	0.6046	0.5347	0.6219	0.6287	0.6219	0.5902	0.5988	0.5755	0.5988	0.5767
Average	0.6093	0.6256	0.6093	0.5404	0.6140	0.6160	0.6140	0.5843	0.5969	0.5794	0.5969	0.5784
All_data	0.6353	0.6541	0.6353	0.5772	0.6276	0.6196	0.6276	0.5971	0.6257	0.6141	0.6257	0.6006

TABLE VI
PERFORMANCE ON DIFFERENT TRAINING DATASETS USING THE DEEP LEARNING MODEL BERT

Dataset	Accuracy	Precision	Recall	F1-score
Kappas>0.5	0.7015	0.7021	0.7015	0.6936
Train_mix_1	0.7008	0.7070	0.7008	0.6964
Train_mix_2	0.7095	0.7179	0.7095	0.7075
Train_mix_3	0.7059	0.7101	0.7059	0.7019
Train_mix_4	0.7099	0.7161	0.7099	0.7071
Train_mix_5	0.7079	0.7141	0.7079	0.7034
Train_mix_6	0.7140	0.7175	0.7140	0.7099
Train_mix_7	0.7086	0.7169	0.7086	0.7066
Train_mix_8	0.7091	0.7156	0.7091	0.7054
Train_mix_9	0.7058	0.7159	0.7058	0.7038
Train_mix_10	0.7048	0.7140	0.7048	0.7009
Average	0.7071	0.7134	0.7071	0.7033
All_data	0.7294	0.7286	0.7294	0.7257

E. Handling Class Imbalance

Class imbalance is a common data quality issue in many datasets. It is not difficult to find from Table IV that the legal argument corpus developed in this research also suffers from the class imbalance issue. We resample the datasets based on oversampling and downsampling, aiming to explore the most effective strategy to solve the class imbalance problem. The parameters for different sampling methods are set as follows:

1) *Oversampling:* For oversampling, the SMOTE with k neighbors is set to 5 by default[3]. Data from each minor class are synthesized to the same data in the most significant class, which includes 2253 records.
2) *Downsampling:* This method reduces the amount of data from each major class to the same data in the class with the least training data. We randomly sample 225 records from each major class without replacement.
3) *Mixed-sampling (i.e., downsampling + oversampling + SMOTE):* For mixed-sampling, each class is considered, and we make sure that the ratio of the largest class to any other class is 6:4. The major classes are initially randomly downsampled to 1500 instances per class, and the minor classes are oversampled to 1000 cases per class. Then, SMOTE comes in and creates more synthetic data for each minor class to keep up with the major class data.

[3][Online] Available: https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/

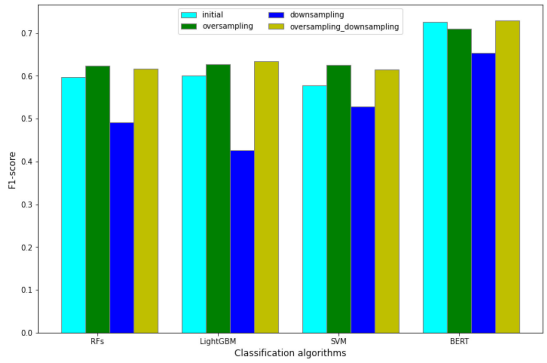


Fig. 5. Performance using different methods to handle class imbalance.

Fig. 5 displays the performance regarding oversampling and downsampling on the three classifiers, respectively. It shows that oversampling fits the dataset better for both machine learning and deep learning.

F. Discussion

From the results of data quality evaluation experiments, we summarize the insights as follows:

1) The commonly used kappa agreement is not sufficient to measure the annotated data quality. Customized experiments are needed to rigorously evaluate the data quality for specific machine learning applications.
2) The insufficient amount of data and class imbalance are the two major data quality issues for machine learning applications.
3) Many techniques can be applied for handling the data imbalance in text classification. When data are insufficient, oversampling is a better choice. However, when large categories exist, mixed-sampling is more effective.

V. DATA AUGMENTATION

A. Algorithms

1) Pseudolabeling: Pseudolabeling is an SSL algorithm, which is used when training data are insufficient. It is trained in a supervised fashion with labeled and unlabeled data simultaneously [67]. A pseudolabel will be assigned to the unlabeled data based on the maximum predicted probability in each class [67]. The whole process includes following five steps.

a) Construct a model using the training data.
b) Predict labels based on the confidence score for an unseen test dataset from unlabeled data.
c) Add confident predicted test observations to the training data.
d) Build a new model using the combined data.
e) Use the new model to predict the external test data. Different classifiers, confidence thresholds, and the number of unlabeled samples affect the model performance.

The pseudocode of our experiment setting is described in Algorithm 1.

Algorithm 1: Pseudolabeling.

Input: An initial set of labeled data LD, a set of unlabeled data UD, a separated test dataset
for model $=$ RF,SVM,LightGBM **do**
  for confidence-score $= 0.99, 0.98, \ldots, 0.90$ **do**
    Update the confidence score for selecting the pseudolabel
    for num-unlabeled $= 0, 5k, 10k \ldots, N$ **do**
      Update the number of unlabeled samples for pseudolabeling
      for $i = 1, 2, 3 \ldots, |U|$ **do**
        Assign d_i a pseudolabel with the above confidence score and add the sample to training data
      end for
    end for
    Optimize the amount of unlabeled data for pseudolabeling
  end for
  Optimize the threshold to select pseudolabel
end for
Output: A classier that takes an unlabeled document and predicts a class label

2) Co-Training: Co-training is an another type of SSL approach by incorporating the discriminative power of different classifiers, such as RF, SVM, and NB [18]. It takes advantage of the strength of different classifiers [88]. The procedure includes following six steps.

a) Construct three models (RF, SVM, and LightGBM) based on the training/test on an 80/20 ratio.
b) Select the best model to be trained on the whole labeled dataset.
c) Predict the class labels of the unlabeled dataset and select the most confident predictions.
d) Iterate 1)–3) until all unlabeled data items are labeled.
e) Build a new model using combined data that include the labeled dataset and the data that was just labeled with high confidence.
f) Use the new model to predict the external test data. The pseudocode of the experiment setting is described in Algorithm 2.

Algorithm 2: Co-training.

Input: An initial set of labeled data LD, a set of unlabeled data UD, a separated test dataset
Initialize a shared training set LD
Initialize a RF classifier
Initialize a SVM classifier
Initialize a LightGBM classifier
while UD have instances **do**
  Split LD in train/test splits using the hold out method (20%)
  Evaluate performance (accuracy) of RF, SVM, and LightGBM using the splits
  Select the best classifier as the candidate classifier
  Train the candidate classifier on LD
  Use the final classifier to predict U and select the most confident predictions, remove them from UD and add them to LD
end while
Output: A classier that takes an unlabeled document and predicts a class label

3) EM + LightGBM: EM has been frequently used for learning from labeled and unlabeled documents. The algorithm is usually implemented with following two steps.

a) First, train a classifier using the available labeled documents, and probabilistically label the unlabeled documents.
b) Second, train a new classifier using the labels for all the documents, and iterate the steps until the best classifier is built [19].

In this research, we apply EM with the LightGBM model as the initial classifier for the data augmentation. LightGBM instead of SVM is used since the former outperformed the latter in our previous experiment. The proposed algorithm is very similar to [89], which first extracts reliable negative examples, then uses LightGBM iteratively until it builds the best classifier. The pseudocode is presented in Algorithm 3.

4) GAN + BERT: GAN with BERT has been successfully applied for enhancing text classification when a small portion of labeled data and a large amount of unlabeled data are available [16]. The algorithm includes following two components.

a) Task-specific layers, for fine-tuning on the real-labeled data and unlabeled data.
b) SS-GAN layers, with the generated fake data to enable SSL.

Multilayer perceptron (MLP) is used to produce the vector or fake instances, and the discriminator receives the input of either vector from fake instances, labeled instances, or unlabeled instances. The discriminator aims to classify the real instance to one of the k classes (i.e., six classes in this research) and the fake instances to another $k + 1$ class. The pseudocode is described in Algorithm 4.

The detailed definitions of $L_{D_{\text{supervise}}}$, $L_{D_{\text{unsupervise}}}$, and L_G can be referred to [16].

Algorithm 3 EM + LightGBM

Input: An initial set of labeled data LD, a set of unlabeled data UD, a separated test dataset
E-step: Identify a set of reliable negative documents from the unlabeled set.
E-step-1: Construct positive feature set (PF) and negative feature set (NF) with 1-DNF
Extract word feature set $\{w_1, w_2, ..., w_n\}$, $w_i \in$ LD $\cup$ UD
Assume PF = { }, NF = { }
for i = 1 to n **do**
  if $\frac{\text{freq}(w_i,\text{LD})/|\text{LD}|}{\text{freq}(w_i,\text{UD})/|\text{UD}|}$ >threshold **then**
    PF = PF $\cup \{w_i\}$
  else
    NF = NF $\cup \{w_i\}$
  end if
end for
E-step-2: Find reliable negative samples
RN = UD
for each document in $d \in$ UD **do**
  if $\exists x_j$ freq(x_j, d) >0 and $x_j \in$ PF **then**
    RN = RN – {d}
  end if
end for
M-step: Build a set of classifiers by iteratively applying LightGBM and then selecting the best from the set.
Assign each document in RN the class label –1
k = 1
while UD have instances **do**
  Use LD and RN to train a LightGBM classifier S_k
  Classify Q (Q = UD – RN) using S_k
  Let the set of documents in Q that are classified as negative be W
  if W = { } **then**
    exit-loop
  else
    Q = Q – W
    RN = RN $\cup$ W;
    k = k +1
  end if
end while
Use the last LightGBM classifier S last to classify LD
if >8% positive are classified as negative **then**
  use S_1 as the final classifier
else
  use S_{last} as the final classifier
end if
Output: A classier that takes an unlabeled document and predicts a class label

Algorithm 4 GAN + BERT

Input: An initial set of labeled data LD, a set of unlabeled data UD, a separated test dataset
Sentence embeddings h_{CLS} for each text sentence in LD and UD by BERT
Generate a 100-D noise vector from $N(\mu, \sigma^2)$
Produce the output vector h_{fake} using MLP (Generator G)
Input h_{CLS} and h_{fake} to another MLP (discriminator D)
for iteration = 1, 2,..., n **do**
  Update D:
    $D = L_{D_{\text{supervise}}} + L_{D_{\text{unsupervise}}} + L_G$
  for epoch = 1, 2,..., p **do**
    for d in labeled data **do**
      Update $L_{D_{\text{supervise}}}$
    end for
    for d in unlabeled data **do**
      Update $L_{D_{\text{unsupervise}}}$
    end for
    for d in generated fake data **do**
      Update L_G
    end for
  end for
  Update label assignment
end for
Output: A classier that takes an unlabeled document and predicts a class label

B. Settings of Experiments

We train the models on a Windows 10 machine with one NVIDIA Tesla Titan V GPU, eight Intel(R) CPUs (i7-9700 @3.00 GHz), and 128 GB of RAM. The hyperparameter settings for all the algorithms are described as follows:

1) For all experiments, 4416 annotated sentences were split into 80 and 20% for training and validation, respectively, while a separated dataset with 521 samples was used for testing. We also collected additional 542 763 unlabeled sentences.
2) For pseudolabeling, we incrementally ran the amount of unlabeled data from 0 to 24 000 with a step increase of 1000. Meanwhile, we incrementally changed the confidence score from 0.10 to 0.90 with a step increase of 0.1, and find that the performance kept increasing when we increased the confidence score. We then changed the confidence score from 0.90 to 0.99 with a step increase of 0.01.
3) For co-training, we compared the performances using two classifiers, which include RF and LightGBM, and three classifiers, which include RF, LightGBM, and SVM.
4) For EM, we set the threshold as 8% positive that can be classified as negative to select the final classifier as suggested by Liu *et al.* [89]. We set the parameter "percent_thresh" as 5, and further experiment with different values to visualise the performance changes.
5) For co-training and EM, we looped the algorithm with the range of unlabeled data size from 0 to 1000 with a step size of 20 to find the best classifier.
6) The other parameters of pseudolabeling, co-training, and EM were set the same as the initial LightGBM model.
7) As for the BERT and BERT-based models, we set the batch size to 64, with a max sequence length of 64 and a learning rate of 2e-5 to ensure that the GPU memory is fully utilized. The dropout probability was always kept at 0.1. We used Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We empirically set the max number of the epoch to 16 for BERT and saved the best model on the validation set for testing.
8) The other parameters for GAN-BERT were set the same as BERT. However, we empirically set the max number of the epoch to 5 for GAN-BERT. In addition, we set the warmup_proportion as 0.1, as suggested in [16].

For all the experiments, we used the mixed-sampling method to handle the class imbalance issue if applicable, and conducted five-fold cross-validation.

C. Results

1) Overall Results: In order to validate the capability of different data augmentation techniques, we presented the

TABLE VII
PERFORMANCE OF DIFFERENT ALGORITHMS FOR DATA AUGMENTATION

Model	Accuracy	Precision	Recall	F1 score
LightGBM	0.6257	0.6141	0.6257	0.6006
BERT + tuning	0.7294	0.7286	0.7294	0.7257
LegalBERT + tuning	0.7313	0.7304	0.7313	0.7284
Pseudo-labeling	0.6334	0.6515	0.6334	0.6394
Co-training	0.6756	0.6785	0.6756	0.6703
EM+LightGBM	0.6564	0.6681	0.6564	0.6393
GAN+BERT	**0.7562**	**0.7539**	**0.7562**	**0.7525**

The significance of boldface numbers are the best results.

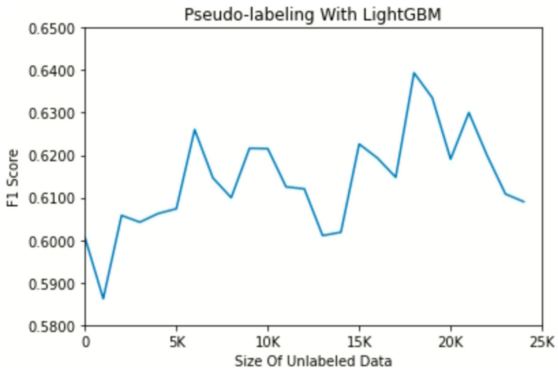


Fig. 6. Performance of using a different number of unlabeled data.

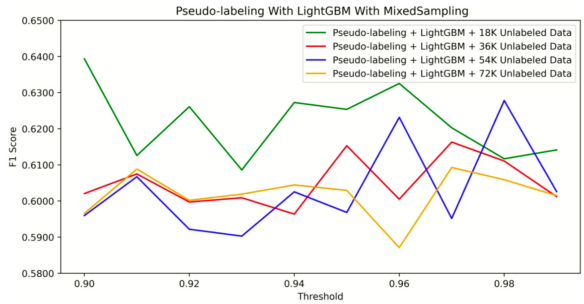


Fig. 7. Performance of using different confidence scores for pseudolabeling.

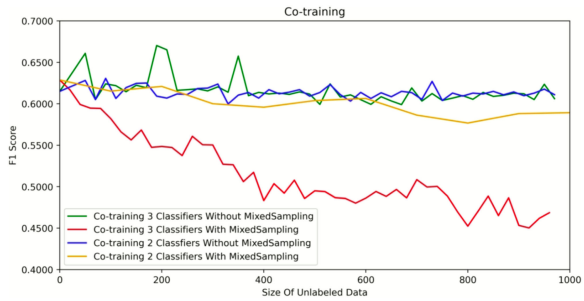


Fig. 8. Performance of co-training. Two classifiers (LightGBM and RF) and three classifiers (LightGBM, RF, and SVM) with or without mixed-sampling. Different amounts of unlabeled data were used.

performance of each algorithm regarding the accuracy, precision, recall, and F1 score in Table VII. GAN + BERT achieved the best performance (F1 score = 0.7525), followed by LegalBERT (F1 score = 0.7284) and the BERT model (F1 score = 0.7257). The results demonstrated that BERT could produce high-quality representations of the input text, and adopted unlabeled material can help the network in generalizing its representations for the legal argument classification task. Meanwhile, GAN + BERT achieved a 0.0268 in F1 score improvement than the general BERT model, indicating that the "fake" examples automatically generated in the GAN framework can enhance the inner representations of argument sentences for GAN-BERT.

The results also showed that deep learning-based models outperform traditional machine learning-based models in the short text classification in the legal domain. Pseudolabeling, co-training, and EM frameworks are beneficial for the traditional text classification algorithms, yet their improvement is limited. For example, LightGBM with pseudolabeling, co-training, and EM have improved LightGBM itself by 0.0388, 0.0697, and 0.0387 on F1 scores, respectively. The effectiveness of different data augmentation on performance improvement varies. In addition, data augmentation techniques are not always effective for performance improvement. Only carefully selecting the training strategy, the amount of unlabeled data and the appropriate parameters could produce desired results. We conducted a series of experimental results to find the best practices for applying different data augmentation techniques in Section V-C2.

2) Parameter Analysis: Fig. 6 shows the results of pseudolabeling with LightGBM using different portions of unlabeled data for the data augmentation. We incrementally add new unlabeled data for the pseudolabeling and check the changing of the performance. The confidence score is kept as 0.90, and the mixed-sampling method is applied in this process. The unlabeled data disrupts the model initially, while benefits the model when the unlabeled data increases. Finally, it achieves the best performance when 18 000 unlabeled data items are used. Fig. 7 shows that when a different number of unlabeled data is used for the data augmentation, the best confidence score varies. For example, when 18 000, 36 000, 54 000, and 72 000 unlabeled samples are used, the best confidence scores are 0.90, 0.95, 0.98, and 0.97, respectively. We can roughly conclude that a higher confidence score is helpful for data quality assurance when a larger amount of unlabeled data is used.

The results of co-training are shown in Fig. 8. Unlike pseudolabeling, mixed-sampling distorted the model performance, as seen from the yellow line and the red line. Under the mixed-sampling setting, unlabeled data do not contribute, but actually damage the model performance. Instead, when the original imbalanced dataset is used, co-training with three classifiers occasionally outperformed the one with two classifiers, as demonstrated in the green and purple lines. The performance improvement brought by the three classifiers depends on the number of unlabeled samples used in the training. For example, when the number of unlabeled samples is 50, 190, 210, and 350, co-training with the three classifiers achieves 0.6609, 0.6703, 0.6651, 0.6576 in F1 score, respectively. They are significantly higher than the one with two classifiers. Moreover, the co-training framework does not appreciate more unlabeled data.

Fig. 9 shows the results of the EM with LightGBM algorithms. Different from the previous two algorithms, the performance of EM + LightGBM was not significantly affected by the class imbalance issue. Even without handling the class imbalance, the algorithm achieved the best performance of 0.6393 in F1

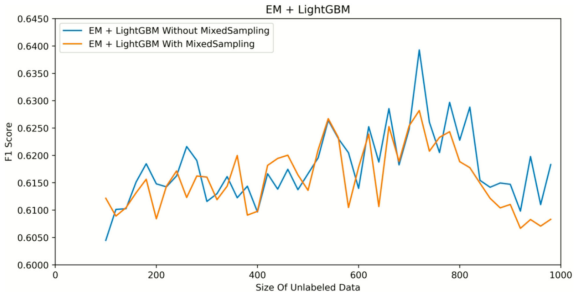


Fig. 9. Performance of EM + LightGBM with and without handling class imbalance.

TABLE VIII
PERFORMANCE OF GAN-BERT USING A DIFFERENT NUMBER OF UNLABELED DATA

#unlabeled data	label rate	F-1 score
437,184	1%	0.7390
216,384	2%	0.7305
142,784	3%	0.7326
105,984	4%	0.7116
83,904	5%	0.7272
69,184	6%	0.7371
58,670	7%	0.7295
50,784	8%	0.7266
44,651	9%	0.7257
39,744	10%	**0.7525**
17,664	20%	0.7361
10,304	30%	0.7477
6,624	40%	0.7139
4,416	50%	0.7436

score when 720 unlabeled samples were used. Using either too few or too many unlabeled samples for the training can harm the model performance.

Deep learning usually needs more training data than the traditional machine learning, which can be observed from Figs. 7–9 and Table VIII. We gradually increase the percentage of unlabeled data for GAN-BERT, and the corresponding results are given in Table VIII. The performance stays stable with around 1%–2% fluctuation when the labeled data are less than 10% to more than 20%. The best performance, 0.7525 in F1 score, is achieved when the labeled data ratio is 10%. The result aligns with the findings from [16], and it is confirmed when about 5000 labeled documents are used. Compared with BERT or LegalBERT, GAN-BERT needs fewer data (i.e., about 10%) for fine-tuning and can achieve higher performance, indicating that the classifier has benefited from both the GAN framework and the fine-tuning.

D. Discussion

From the data augmentation experimental results described earlier, we summarize the insights as follows:

1) Both the selection of new training data from automated labeling data and the selection of the amount of unlabeled data for the data augmentation affect the model performance. From the experiments, we find that pseudolabeling requires more unlabeled data for the data augmentation than the co-training and EM. The reason is that pseudolabeling uses a filter to select the most confident labeling samples for the retraining, while co-training uses all the unlabeled input data and EM uses most of the unlabeled input data for the training. In this case, data quality is essential for assuring the model performance.
2) Techniques for handling the class imbalance issue that works well for supervised learning are not always effective for SSL. For example, mixed-sampling decreases the performance of co-training and EM. The reason is that the noise data can be easily taken into the two models for retraining. When mixed-sampling, especially oversampling, is conducted, there is a high possibility that the noise data were also replicated, which reduces the data quality. Therefore, resampling is helpful for co-training and EM-based data augmentation algorithms. On the contrary, oversampling improves the data quality in pseudolabeling, because it takes advantage of the confidence score in selecting high-quality data.
3) The fine-tuning can be used to improve data quality by taking advantage of the information brought by the source dataset. However, the datasets for fine-tuning should be related to the target-specific task [20] (the legal text mining on Texas criminal cases in this research). It explains that LegalBERT outperforms BERT because LegalBERT was fine-tuned with all the unlabeled legal datasets while BERT was not. Our experiments also demonstrate that fine-tuning benefits more from larger datasets than smaller datasets.
4) GAN can enhance BERT when only few labeled data are available. Compared to BERT, GAN improves the model performance with even less labeled data while not introducing additional costs. Compared to LegalBERT, it requires less unlabeled data, thus, reduces the training time and resources.

The results of our experiments provide clear instructions on the selection and implementation of different algorithms for data augmentation. Although the algorithms have been applied for data augmentation in previous research, our work is the first systematic investigation and comparative study of these algorithms. We demonstrate that these algorithms can achieve comparable performance in the legal domain.

VI. FURTHER DISCUSSION AND IMPLICATIONS

In Fig. 10, we propose a framework for extending the dataset based on the data augmentation techniques. The idea is to imitate human annotation with machine learning algorithms. The framework includes following three steps.

1) Test dataset creation.
2) Evaluation and selection of classifiers.
3) Automatically labeling more training data using the selected classifiers.

By following the framework, we can expand the size of the dataset and ensure the data quality of the expanded dataset in the meanwhile.

Based on the abovementioned framework developed, we first label a small portion of data with domain experts by following the process in Section III. This step aims to create the dataset

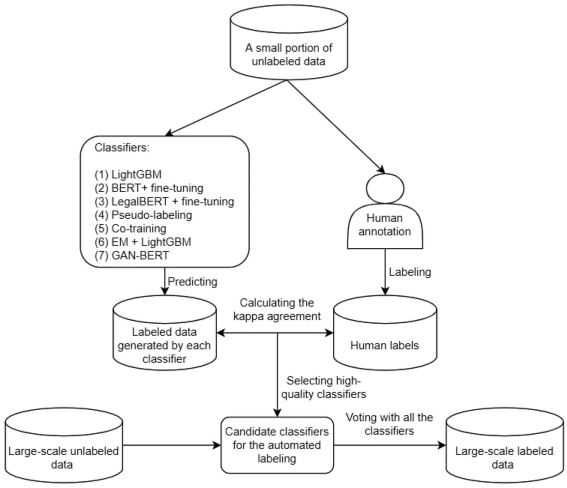


Fig. 10. Framework for extending the legal argument mining dataset based on data augmentation techniques.

for evaluating the quality of the automatic labeling of different classifiers.

The second step is to evaluate the annotation quality of each classifier and select the best classifiers for dataset expansion. Note that the small portion of data annotated by domain experts will also be automated labeled by each classifier developed in Sections IV and V. We calculate the kappa agreement between the labels, predicted by the classifier and generated by human experts. If the agreement score reaches a predefined threshold, we keep the classifier for final dataset extension. Otherwise, we remove the classifier from the candidate list.

The third step is to produce more high-quality training data with the classifiers selected in the second step. We apply the selected classifiers for labeling large-scale unlabeled data, then use majority vote to get the final label for each data record. In this way, we will create large-scale labeled data. Instead of using one classifier for the dataset extension, we use the majority vote from multiple classifiers to avoid bias and ensure the data quality.

In the future, we will use this strategy to build a high-quality large legal argument mining dataset and release the dataset to the public. The strategy can also be reused in other domains.

VII. Conclusion

Legal argument mining has becoming a prominent task in legal AI. However, it is very challenging to build a high-quality legal argument mining corpus either manually or automatically since the former is costly and the latter can hardly assure the data quality. To bridge the gap, we introduced semiautomated approaches for the legal argument mining corpus construction and augmentation in this article. We first proposed an annotation scheme for legal arguments, then conducted an annotation experiment to construct an initial corpus using the United States case law. Instead of relying on kappa agreement for the quality evaluation, we designed a series of experiments to quantitatively evaluate the data quality. Finally, we experimented with several widely used machine learning methods, including pseudolabeling, co-training, EM, and GAN for automated data augmentation. A group of guidelines were proposed for selecting and implementing different data augmentation algorithms, which are summarized as follows:

1) Compared to traditional machine learning models, such as SVM and LightGBM, a powerful model, such as BERT, can better capture semantic information, thereby can be the prime model for legal argument classification.
2) GAN-BERT outperformed other algorithms for data augmentation. We can choose the GAN-BERT model for data augmentation when a small portion of labeled data and a large portion of unlabeled data are available.
3) Handling the class imbalance issue with mixed-sampling can improve the performance of supervised learning. However, it may bring some noise data; thereby reducing the model performance during the data augmentation using co-training and EM.
4) The amount of unlabeled data for data augmentation should be carefully selected for training, otherwise the new labeled data may produce a negative impact on the model performance.

The data augmentation techniques we proposed and the practical guidelines we summarized from the experiments can automatically label more legal arguments. This article is intended to be the foundation of legal argument mining and generation.

In the future, we will investigate the combination of BERT with different data augmentation algorithms for better legal argument classification. Recently, several studies incorporated prior knowledge and manual-crafted features into BERT to guide its attention selection and achieved improved performance [90], [91]. For example, Xia *et al.* [91] proposed a knowledge-enhanced BERT, which injected knowledge into BERT's multihead attention mechanism. The model is able to consistently improve semantic textual matching performance over the original BERT model [91]. Enlightened by this idea, we will also explore the effectiveness of domain concepts in optimizing BERT's attention on legal text classification. Our ultimate goal is to build a high-quality legal intelligent system for automated legal argument mining and generation.

Acknowledgment

The authors would like to thank V. Wong, T. Blanchet, R. Li, W. Kostuch, . K. S. Tummala, S. Penupala, M. Jing, P. Mayank, and J. Ding for designing the annotation guideline and participating in the annotation experiment. The authors appreciate Dr. A. Palmer from the University of Colorado Boulder and Dr. J. Chen from the University of North Texas for giving them wonderful feedback on the algorithm design. The authors would also like to thank S. S. Vadla for conducting part of the experimental study in this research, and are grateful to all the anonymous reviewers for their precious comments and suggestions.

References

[1] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, "How does NLP benefit legal system: A summary of legal artificial intelligence," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5218–5230.
[2] J. Lawrence and C. Reed, "Argument mining: A survey," *Comput. Linguistics*, vol. 45, no. 4, pp. 765–818, 2019.
[3] R. M. Palau and M.-F. Moens, "Argumentation mining: The detection, classification and structure of arguments in text," in *Proc. 12th Int. Conf. Artif. Intell. Law*, 2009, pp. 98–107.

[4] I. Lourentzou, "Data quality in the deep learning ERA: Active semi-supervised learning and text normalization for natural language understanding," Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois Urbana-Champaign, Champaign, IL, USA, 2019.

[5] D. Hendrycks, C. Burns, A. Chen, and S. Ball, "CUAD: An expert-annotated NLP dataset for legal contract review," *CoRR*, vol. abs/2103.06268, 2021. Accessed: Dec.. 10, 2021. [Online]. Available: https://arxiv.org/abs/2103.06268

[6] E. Hovy and J. Lavid, "Towards a 'science'of corpus annotation: A new methodological challenge for corpus linguistics," *Int. J. Transl.*, vol. 22, no. 1, pp. 13–36, 2010.

[7] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.

[8] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 6382–6388.

[9] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[10] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.

[11] I. Goodfellow *et al.*, "Generative adversarial Nets," in *Proc. 27th Int. Con. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc.: Ser. B.*, vol. 39, no. 1, pp. 1–22, 1977.

[13] U. Brefeld and T. Scheffer, "Co-EM support vector learning," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 16.

[14] A. Chokka and K. S. Rani, "A cluster-based improved expectation maximization framework for identification of somatic gene clusters," in *Emerging Research in Data Engineering Systems and Computer Communications*. Berlin, Germany: Springer, 2020, pp. 521–534.

[15] K. Greff, S. van Steenkiste, and J. Schmidhuber, "Neural expectation maximization," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6694–6704.

[16] D. Croce, G. Castellucci, and R. Basili, "GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 2114–2119.

[17] M. L. McHugh, "Interrater reliability: The kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[18] S. Tariq *et al.*, "A novel co-training-based approach for the classification of mental illnesses using social media posts," *IEEE Access*, vol. 7, pp. 166165–166172, 2019.

[19] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, no. 2, pp. 103–134, 2000.

[20] H. Chen, J. Chen, and J. Ding, "Data evaluation and enhancement for quality improvement of machine learning," *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 831–847, Jun. 2021.

[21] E. Cabrio and S. Villata, "Five years of argument mining: A data-driven analysis," *Int. Joint Conf. Artif. Intell.*, vol. 18, pp. 5427–5433, 2018.

[22] R. Mochales and M.-F. Moens, "Argumentation mining," *Artif. Intell. Law*, vol. 19, no. 1, pp. 1–22, 2011.

[23] M. Teruel, C. Cardellino, F. Cardellino, L. A. Alemany, and S. Villata, "Increasing argument annotation reproducibility by using inter-annotator agreement to improve guidelines," in *Proc. 11th Int. Conf. Lang. Resour. Eval.*, 2018, pp. 4061–4064.

[24] M. Grabmair *et al.*, "Introducing LUIMA: An experiment in legal conceptual retrieval of vaccine injury decisions using a UIMA type system and tools," in *Proc. 15th Int. Conf. Artif. Intell. Law*, 2015, pp. 69–78.

[25] W. Yang, W. Jia, X. Zhou, and Y. Luo, "Legal judgment prediction via multi-perspective bi-feedback network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4085–4091.

[26] H. Yamada, S. Teufel, and T. Tokunaga, "Building a corpus of legal argumentation in japanese judgement documents: Towards structure-based summarisation," *Artif. Intell. Law*, vol. 27, no. 2, pp. 141–170, 2019.

[27] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, "Automatic detection of arguments in legal texts," in *Proc. 11th Int. Conf. Artif. Intell. Law*, 2007, pp. 225–230.

[28] J. Visser, J. Lawrence, C. Reed, J. Wagemans, and D. Walton, "Annotating argument schemes," *Argumentation*, vol. 35, no. 1, pp. 101–139, 2021.

[29] V. W. Feng and G. Hirst, "Classifying arguments by scheme," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2011, pp. 987–996.

[30] K. Ashley, "Applying argument extraction to improve legal information retrieval," in *Proc. ArgNLP*, 2014, pp. 1–9.

[31] Y. Le, C. He, M. Chen, Y. Wu, X. He, and B. Zhou, "Learning to predict charges for legal judgment via self-attentive capsule network," in *Proc. Eur. Conf. Artif. Intell.*, 2020, pp. 1802–1809.

[32] Q. Li *et al.*, "A survey on text classification: From shallow to deep learning," *CoRR*, vol. abs/2008.00364, 2021. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2008.00364

[33] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning-based text classification: A comprehensive review," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–40, 2021.

[34] M. Možina, J. Žabkar, T. Bench-Capon, and I. Bratko, "Argument based machine learning applied to law," *Artif. Intell. Law*, vol. 13, no. 1, pp. 53–73, 2005.

[35] B. Hachey and C. Grover, "Sequence modeling for sentence classification in a legal summarisation system," in *Proc. ACM Symp. Appl. Comput.*, 2005, pp. 292–296.

[36] I. S. Bajwa, F. Karim, M. A. Naeem, and R. Ul Amin, "A semi supervised approach for catchphrase classification in legal text documents," *J. Comput.*, vol. 12, no. 5, pp. 451–461, 2017.

[37] K. D. Ashley and V. R. Walker, "Toward constructing evidence-based legal arguments using legal decision documents and machine learning," in *Proc. 14th Int. Conf. Artif. Intell. Law*, 2013, pp. 176–180.

[38] C. Su and D. Huang, "Hybrid recommender system based on deep learning model," *Int. J. Performability Eng.*, vol. 16, no. 1, pp. 118–129, Jan. 2020.

[39] X. Bai, H. Zhou, and H. Yang, "An HVSM-based gru approach to predict cross-version software defects," *Int. J. Performability Eng.*, vol. 16, no. 6, pp. 979–990, Jun. 2020.

[40] S. Undavia, A. Meyers, and J. E. Ortega, "A comparative study of classifying legal documents with neural networks," in *Proc. IEEE Federated Conf. Comput. Sci. Inf. Syst.*, 2018, pp. 515–522.

[41] I. Chalkidis, I. Androutsopoulos, and N. Aletras, "Neural legal judgment prediction in english," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4317–4323.

[42] G. Li, Z. Wang, and Y. Ma, "Combining domain knowledge extraction with graph long short-term memory for learning classification of chinese legal documents," *IEEE Access*, vol. 7, pp. 139616–139627, 2019.

[43] Z. Shaheen, G. Wohlgenannt, and E. Filtz, "Large scale legal text classification using transformer models," *CoRR*, vol. abs/2010.12871, 2020. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2010.12871

[44] H. Chen, D. Hailey, N. Wang, and P. Yu, "A review of data quality assessment methods for public health information systems," *Int. J. Environ. Res. Public Health*, vol. 11, no. 5, pp. 5170–5207, 2014.

[45] J. Yoon, S. Arik, and T. Pfister, "Data valuation using reinforcement learning," in *Proc. Int. Conf. Mach. Learn*, 2020, pp. 10 842–10851.

[46] Y. Liu, Y. Wang, K. Zhou, Y. Yang, and Y. Liu, "Semantic-aware data quality assessment for image Big Data," *Future Gener. Comput. Syst.*, vol. 102, pp. 53–65, 2020.

[47] S. Mishra, A. Arunkumar, B. Sachdeva, C. Bryan, and C. Baral, "DQI: Measuring data quality in NLP," *CoRR*, vol. abs/2005.00816, 2020. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2005.00816

[48] J. Gao, X. Li, Y. E. Xu, B. Sisman, X. L. Dong, and J. Yang, "Efficient knowledge graph accuracy evaluation," *CoRR*, vol. abs/1907.09657, 2019. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/1907.09657

[49] E. Breck, N. Polyzotis, S. Roy, S. Whang, and M. Zinkevich, "Data validation for machine learning," in *Proc. 2nd SysML Conf.*, 2019.

[50] D. Basavakumar, M. Flegg, J. Eccles, and P. Ghezzi, "Accuracy, completeness and accessibility of online information on fibromyalgia," *Rheumatol. Int.*, vol. 39, no. 4, pp. 735–742, 2019.

[51] A. Wisesa, F. Darari, A. Krisnadhi, W. Nutt, and S. Razniewski, "Wikidata completeness profiling using proWD," in *Proc. 10th Int. Conf. Knowl. Capture*, 2019, pp. 123–130.

[52] M. R. A. Rashid, G. Rizzo, M. Torchiano, N. Mihindukulasooriya, O. Corcho, and R. García-Castro, "Completeness and consistency analysis for evolving knowledge bases," *J. Web Semantics*, vol. 54, pp. 48–71, 2019.

[53] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo, "'Everyone wants to do the model work, not the data work': Data cascades in high-stakes AI," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, pp. 1–15.

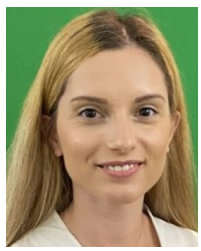
[54] J. Ding, X. Hu, and V. Gudivada, "A machine learning based framework for verification and validation of massive scale image data," *IEEE Trans. Big Data*, vol. 7, no. 2, pp. 451–467, Jun. 2021.

[55] K. Pustu-Iren *et al.*, "Investigating correlations of inter-coder agreement and machine annotation performance for historical video data," in *Proc. Int. Conf. Theory Pract. Digit. Libraries*, 2019, pp. 107–114.

[56] K. Alhazmi, W. Alsumari, I. Seppo, L. Podkuiko, and M. Simon, "Effects of annotation quality on model performance," in *Proc. IEEE Int. Conf. Artif. Intell. Inf. Commun.*, 2021, pp. 63–67.
[57] P.-Y. Hsueh, P. Melville, and V. Sindhwani, "Data quality from crowdsourcing: A study of annotation selection criteria," in *Proc. NAACL HLT Workshop Act. Learn. Natural Lang. Process.*, 2009, pp. 27–35.
[58] H. Huang, B. Stvilia, C. Jörgensen, and H. W. Bass, "Prioritization of data quality dimensions and skills requirements in genome annotation work," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 63, no. 1, pp. 195–207, 2012.
[59] L. Gienapp, B. Stein, M. Hagen, and M. Potthast, "Efficient pairwise annotation of argument quality," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5772–5781.
[60] J. Van Hulse, "Data quality in data mining and machine learning," Ph.D. dissertation, Dept. Comput. Sci. Eng., Florida Atlantic Univ., Boca Raton, FL, USA, 2007.
[61] E. J. Lauria and G. K. Tayi, "Statistical machine learning for network intrusion detection: A data quality perspective," *Int. J. Serv. Sci.*, vol. 1, no. 2, pp. 179–195, 2008.
[62] W.-H. Weng, J. Deaton, V. Natarajan, G. F. Elsayed, and Y. Liu, "Addressing the real-world class imbalance problem in dermatology," in *Proc. Mach. Learn. Health*, 2020, pp. 415–429.
[63] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, 2019.
[64] S. Y. Feng *et al.*, "A survey of data augmentation approaches for NLP," *CoRR*, vol. abs/2105.03075, 2021. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2105.03075
[65] C. Lanquillon, "Learning from labeled and unlabeled documents: A comparative study on semi-supervised text classification," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discov.*, 2000, pp. 490–497.
[66] X. Li and B. Yang, "A pseudo label based dataless naive bayes algorithm for text classification with seed words," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1908–1917.
[67] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Representation Learn.*, 2013, p. 896.
[68] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 2234–2242, 2016.
[69] Z. Sun, C. Fan, X. Sun, Y. Meng, F. Wu, and J. Li, "Neural semi-supervised learning for text classification under large-scale pretraining," *CoRR*, vol. abs/2011.08626, 2020. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2011.08626
[70] Y. Su *et al.*, "CSS-LM: A contrastive framework for semi-supervised fine-tuning of pre-trained language models," *CoRR*, vol. abs/2102.03752, 2021. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2102.03752
[71] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, 2001.
[72] P. Dube, B. Bhattacharjee, S. Huo, P. Watson, B. Belgodere, and J. R. Kender, "Automatic labeling of data for transfer learning," in *Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 122–129.
[73] Y.-S. Chen, S.-W. Chiang, and T.-Y. Juang, "A few-shot transfer learning approach using text-label embedding with legal attributes for law article prediction," *Appl. Intell.*, vol. 52, no. 3, pp. 2884–2902, 2022.
[74] W. Yin, N. F. Rajani, D. Radev, R. Socher, and C. Xiong, "Universal natural language processing with limited annotations: Try few-shot textual entailment as a start," *CoRR*, vol. abs/2010.02584, 2020. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2010.02584
[75] T. Schick, H. Schmid, and H. Schütze, "Automatically identifying words that can serve as labels for few-shot text classification," *CoRR*, vol. abs/2010.13641, 2020. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2010.13641
[76] T. B. Brown *et al.*, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/2005.14165
[77] Caselaw Access Project, Accessed: Sep. 1, 2020. [Online]. Available: https://case.law/
[78] M. Ostendorff, E. Ash, T. Ruas, B. Gipp, J. Moreno-Schneider, and G. Rehm, "Evaluating document representations for content-based legal literature recommendations," in *Proc. 18th Int. Conf. Artif. Intell. Law*, 2021, pp. 109–118.
[79] L. Zheng, N. Guha, B. R. Anderson, P. Henderson, and D. E. Ho, "When does pretraining help? Assessing self-supervised learning for law and the caseHOLD dataset," in *Proc. 18th Int. Conf. Artif. Intell. Law*, 2021, pp. 159–168.
[80] M. J. Bommarito II, D. M. Katz, and E. M. Detterman, "LexNLP: Natural language processing and information extraction for legal and regulatory texts," in *Research Handbook on Big Data Law*. Cheltenham, U.K.: Edward Elgar Publishing, 2021.
[81] N. J. White, "Legal writing: Legal arguments, briefs, and outlines," 2021. Accessed: Dec. 10, 2021. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1924979
[82] K. Branting *et al.*, "Semi-supervised methods for explainable legal prediction," in *Proc. 17th Int. Conf. Artif. Intell. Law*, 2019, pp. 22–31.
[83] J. Carletta, "Assessing agreement on classification tasks: The kappa statistic," *Comput. Linguistics*, vol. 22, pp. 249–254, 1996.
[84] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.
[85] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 3146–3154, 2017.
[86] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. Accessed: Dec. 10, 2021. [Online]. Available: https://arxiv.org/abs/1810.04805
[87] M. Khalusova, "Machine learning model evaluation metrics part 2: Multi-classification," 2019, Accessed: Oct. 8, 2021. [Online]. Available: https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/
[88] C. Aridas and S. Kotsiantis, "Combining random forest and support vector machines for semi-supervised learning," in *Proc. 19th Panhellenic Conf. Informat.*, 2015, pp. 123–128.
[89] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in *Proc. 3rd IEEE Int. Conf. Data Mining*, 2003, pp. 179–186.
[90] L. Cai, Y. Song, T. Liu, and K. Zhang, "A hybrid bert model that incorporates label semantics via adjustive attention for multi-label text classification," *IEEE Access*, vol. 8, pp. 152183–152192, 2020.
[91] T. Xia, Y. Wang, Y. Tian, and Y. Chang, "Using prior knowledge to guide bert's attention in semantic textual matching tasks," in *Proc. Web Conf.*, 2021, pp. 2466–2475.

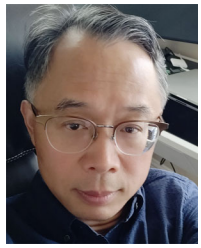
Haihua Chen (Member, IEEE) received the B.S. degree in information management from Central China Normal University, Wuhan, China, in 2014, the M.S. degree in information science from Wuhan University, Wuhan, China, in 2017, and the Ph.D. degree in information science from the Department of Information Science, University of North Texas (UNT), Denton, TX, USA, in 2022.

He is currently a Clinical Assistant Professor of Data Science with the Department of Information Science, UNT, where he is also the Laboratory Manager of Intelligent Information Access Lab. His research interests include data quality in machine learning, legal artificial intelligence, health informatics, and academic data mining.



Lavinia F. Pieptea received the B.S. degree in mathematics from the University of Bucharest, Bucharest, Romania, in 2019. She is currently working toward the graduation degree in mathematics with the Department of Mathematics, University of North Texas, Denton, TX, USA.

Her research interests include biomedical computation, information security, and privacy.



Junhua Ding received the B.S. degree from the China University of Geosciences, Wuhan, China, in 1994, the M.S. degree from Nanjing University, Nanjing, China, in 1997, and the Ph.D. degree from Florida International University, Miami, FL, USA, in 2004, all in computer science.

He is currently a Reinburg Endowed Professor of Data Science with the Department of Information Science, University of North Texas (UNT), Denton, TX, USA, where he is also the Director of Data Science Program. Before he joined UNT in 2018, he was a Faculty Member with the Department of Computer Science, East Carolina University. His research interests include data analytics, machine learning, computational law, data security and privacy, and software engineering.