Legal Area Classification: A Comparative Study of Text Classifiers on Singapore Supreme Court Judgments

Jerrold Soh Tsin Howe[*], **Lim How Khang** , **and Ian Ernst Chai**[**]

[*]Singapore Management University, School of Law
[**]Attorney-General's Chambers, Singapore
jerroldsoh@smu.edu.sg, howkhang.lim@gmail.com
ian_ernst_chai@agc.gov.sg

Abstract

This paper conducts a comparative study on the performance of various machine learning ("ML") approaches for classifying judgments into legal areas. Using a novel dataset of 6,227 Singapore Supreme Court judgments, we investigate how state-of-the-art NLP methods compare against traditional statistical models when applied to a legal corpus that comprised few but lengthy documents. All approaches tested, including topic model, word embedding, and language model-based classifiers, performed well with as little as a few hundred judgments. However, more work needs to be done to optimize state-of-the-art methods for the legal domain.

1 Introduction

Every legal case falls into one or more areas of law ("legal areas"). These areas are lawyers' shorthand for the subset of legal principles and rules governing the case. Thus lawyers often triage a new case by asking if it falls within tort, contract, or other legal areas. Answering this allows unresolved cases to be funneled to the right experts and for resolved precedents to be efficiently retrieved. Legal database providers routinely provide area-based search functionality; courts often publish judgments labelled by legal area.

The law therefore yields pockets of expert-labelled text. A system that classifies legal texts by area would be useful for enriching older, typically unlabelled judgments with metadata for more efficient search and retrieval. The system can also suggest areas for further inquiry by predicting which areas a new text falls within.

Despite its potential, this problem, which we refer to and later define as "legal area classification", remains relatively unexplored. One explanation is the relative scarcity of labelled documents in the law (typically in the low thousands), at least by deep learning standards. This problem is acute in smaller jurisdictions like Singapore, where the number of labelled cases is limited by the few cases that actually reach the courts. Another explanation is that legal texts are typically longer than the customer reviews, tweets, and other documents typical in NLP research.

Against this backdrop, this paper uses a novel dataset of Singapore Supreme Court judgments to comparatively study the performance of various text classification approaches for legal area classification. Our specific research question is as follows: how do recent state-of-the-art models compare against traditional statistical models when applied to legal corpora that, typically, comprise few but lengthy documents?

We find that there are challenges when it comes to adapting state-of-the-art deep learning classifiers for tasks in the legal domain. Traditional topic models still outperform the more recent neural-based classifiers on certain metrics, suggesting that emerging research (fit specially to tasks with numerous short documents) may not carry well into the legal domain unless more work is done to optimize them for legal NLP tasks. However, that shallow models perform well suggests that enough exploitable information exists in legal texts for deep learning approaches better-tailored to the legal domain to perform as well if not better.

2 Related Work

Papers closest to ours are those that likewise examine legal area classification. Goncalves and Quaresma (2005) used bag-of-words ("BOW") features learned using TF-IDF to train linear support vector machines ("linSVMs") to classify decisions of the Portuguese Attorney General's Office into 10 legal areas. Boella et al. (2012) used

Proceedings of the Natural Legal Language Processing Workshop 2019, pages 67–77
Minneapolis, Minesotta, June 7, 2019.

TF-IDF features enriched by a semi-automatically linked legal ontology and linSVMs to classify Italian legislation into 15 civil law areas. Sulea et al. (2017) classified French Supreme Court judgments into 8 civil law areas, again using BOW features learned using Latent Semantic Analysis ("LSA") (Deerwester et al., 1990) and linSVMs.

On legal text classification more generally, Aletras et al. (2016); Liu and Chen (2017); Sulea et al. (2017) used BOW features extracted from judgments and linSVMs for predicting case outcomes. Talley and O'Kane (2012) use BOW features and a linSVM to classify contract clauses.

NLP has also been used for legal information extraction. Venkatesh (2013) used Latent Dirichlet Allocation (Blei et al., 2003) ("LDA") to cluster Indian court judgments. Falakmasir and Ashley (2017) used vector space models to extract legal factors motivating case outcomes from American trade secret misappropriation judgments.

There is also growing scholarship on legal text analysis. Typically, topic models are used to extract N-gram clusters from legal corpora, such as Constitutions, statutes, and Parliamentary records, then assessed for legal significance (Young, 2013; Carter et al., 2016). More recently, Ash and Chen (2019) used document embeddings trained on United States Supreme Court judgments to encode and study spatial and temporal patterns across federal judges and appellate courts.

We contribute to this literature by (1) benchmarking new text classification techniques against legal area classification, and (2) more deeply exploring how document scarcity and length affect performance. Beyond BOW features and linSVMs, we use word embeddings and newly-developed language models. Our novel label set comprises 31 legal areas relevant to Singapore's *common* law system. Judgments of the Singapore Supreme Court have thus far not been exploited. We also draw an important but overlooked distinction between *cases* and *judgments*.

3 Problem Description

Legal areas generally refer to a subset of related legal principles and rules governing certain dispute types. There is no universal set of legal areas. Areas like tort and equity, well-known in English and American law, have no direct analogue in certain civil law systems. Societal change may create new areas of law like data protection. However, the set of legal areas in a given jurisdiction and time is well-defined. Denote this as L.

Lawyers typically attribute a given case c_i to a given legal area $l \in L$ if c_i's attributes v_{c_i} (*e.g.* a vector of its facts, parties involved, and procedural trail) raise legal issues that implicate some principle or rule in l. Cases may fall into more than one legal area but never none.

Cases should be distinguished from the *judgments* courts write when resolving them (denoted j_{c_i}). j_{c_i} may not state everything in v_{c_i} because judges need only discuss issues material to how the case should be resolved. Suppose a claimant mounts two claims on the same issue against a defendant in tort, and in trademark law. If the judge finds for the claimant in tort, he/she may not discuss trademark at all (though some may still do so). Thus, even though v_{c_i} raises trademark issues, j_{c_i} may not contain any N-grams discussing the same. It is possible that a v_{c_i} we would assign to l leads to a j_{c_i} we would not assign to l. The upshot is that judgments are incomplete sources of case information; classifying *judgments* is not the same as classifying *cases*.

This paper focuses on the former. We treat this as a supervised legal text multi-class and multi-label classification task. The goal is to learn $f^* : j_i \mapsto L_{j_i}$ where * denotes optimality.

4 Data

The corpus comprises 6,227 judgments of the Singapore Supreme Court written in English.[1] Each judgment comes in PDF format, with its legal areas labelled by the Court. The median judgment has 6,968 tokens and is significantly longer than the typical customer review or news article commonly found in datasets for benchmarking machine learning models on text classification.

The raw dataset yielded 51 different legal area labels. Some labels were subsets of larger legal areas and were manually merged into those. Label imbalance was present in the dataset so we limited the label set to the 30 most frequent areas. Remaining labels (252 in total) were then mapped to the label "others". Table 1 shows the final label distribution, truncated for brevity. Appendix A.2 presents the full label distribution and all label merging decisions.

[1]The judgments were issued between 3 January 2000 and 18 February 2019 and were downloaded from `http://www.singaporelawwatch.sg`, an official repository of Singapore Supreme Court judgments.

Label	Count
civil_procedure	1369
contract_law	861
criminal_procedure_and_sentencing	775
criminal_law	734
family_law	491
...	...
others	252
...	...
banking_law	75
restitution	60
agency_law	57
res_judicata	49
insurance_law	39
Total	**8853**

Table 1: Truncated Distribution of Final Labels

5 Models and Methods

Given label imbalance, we held out 10% of the corpus by stratified iterative sampling (Sechidis et al., 2011; Szymaski and Kajdanowicz, 2017). For each model type, we trained three separate classifiers on *the same* 10% (n=588), 50% (n=2795), and 100% (n=5599) of the remaining *training* set ("training subsets"), again split by stratified iteration, and tested them against *the same* 10% holdout. We studied four model types of increasing sophistication and recency. These are briefly explained here. Further implementation details may be found in the Appendix A.3.

5.1 Baseline Models

$base_{pdf}$ is a dummy classifier which predicts 1 for any label which expectation equals or exceeds $1/31$ (the total number of labels).

$count_m$ uses a keyword matching strategy that emulates how lawyers may approach the task. It predicts 1 for any label if its associated terms appear $\geq m$ non-unique times in j_i. m is a manually-set threshold. A label's set of associated terms is the union of (a) the set of its sub-labels in the training subset, and (b) the set of non-stopword unigrams in the label itself. We manually added potentially problematic unigrams like "law" to the stopwords list. Suppose the label "tort_law" appears twice in the training subset, first with sub-label "negligence", and later with sub-label "harassment". The associated terms set would be $\{tort, negligence, harassment\}$.

5.2 Topic Models

lsa_k is a one-vs-rest linSVM trained using k topics extracted by LSA. We used LSA and linSVMs as benchmarks because, despite their vintage, they remain a staple of the *legal* text classification literature (see Section 2 above). Indeed, LDA models were also tested but strictly underperformed LSA models in all experiment and were thus not reported. Feature vectorizers and classifiers from scikit-learn (Pedregosa et al., 2011) were retrained for each training subset with all default settings except sublinear term frequencies were used in the TF-IDF step as recommended by Scikit-Learn (2017).

5.3 Word Embedding Feature Models

Word vectors pre-trained on large corpora have been shown to capture syntactic and semantic word properties (Mikolov et al., 2013; Pennington et al., 2014). We leverage on this by initializing word vectors using pre-trained GloVe vectors of length 300.[2] Judgment vectors were then composed in three ways: $glove_{avg}$ average-pools each word vector in j_i (i.e. average-pooling); $glove_{max}$ uses max-pooling (Shen et al., 2018); $glove_{cnn}$ feeds the word vectors through a shallow convolutional neural network ("CNN") (Kim, 2014). We chose to implement a shallow CNN model for $glove_{cnn}$ because it has been shown that deep CNN models do not necessarily perform better on text classification tasks (Le et al., 2018). To derive label predictions, judgment vectors were then fed through a multi-layer perceptron followed by a sigmoid function.

5.4 Pre-trained Language Models

Recent work has also shown that language representation models pre-trained on large unlabelled corpora and fine-tuned onto specific tasks significantly outperform models trained only on task-specific data. This method of transfer learning is particularly useful in legal NLP, given the lack of labelled data in the legal domain. We thus evaluated Devlin et al. (2018)'s state-of-the-art BERT model using published pre-trained weights from $bert_{base}$ (12-layers; 110M parameters) and $bert_{large}$ (24-layers; 340M parameters).[3] However, as BERT's self-attention transformer archi-

[2] http://nlp.stanford.edu/data/glove.6B.zip

[3] https://github.com/google-research/bert

tecture (Vaswani et al., 2017) only accepts up to 512 Wordpiece tokens (Wu et al., 2016) as input, we used only the first 512 tokens of each j_i to fine-tune both models.[4] We considered splitting the judgment into shorter segments and passing each segment through the BERT model but doing so would require extensive modification to the original fine-tuning method; hence we left this for future experimentation. In this light, we also benchmarked Howard and Ruder (2018)'s ULMFiT model which accepts longer inputs due to its stacked-LSTM architecture.

6 Results

Given our multi-label setting, we evaluated the models on and report micro- and macro-averaged F1 scores (Table 2), precision (Table 3), and recall (Table 4). Micro-averaging calculates the metric globally while macro-averaging first calculates the metric *within each label* before averaging across labels. Thus, micro-averaged metrics equally-weight each *sample* and better indicate a model's performance on common labels whereas macro-averaged metrics equally-weight each *label* and better indicate performance on rare labels.

6.1 F1 Score

Subset	10%	50%	100%
$bert_{large}$	45.1 [57.9]	56.7 [63.8]	60.7 [66.3]
$bert_{base}$	43.1 [53.6]	52.0 [57.6]	56.2 [63.9]
$ulmfit$	**45.7** [62.8]	45.9 [63.0]	49.2 [64.3]
$glove_{cnn}$	40.7 [62.2]	58.7 [67.1]	63.1 [70.8]
$glove_{avg}$	36.7 [49.7]	**59.1** [64.3]	61.5 [65.6]
$glove_{max}$	29.2 [47.4]	47.8 [59.9]	52.5 [63.2]
lsa_{250}	37.9 [**63.5**]	55.2 [**70.8**]	**63.2** [**73.3**]
lsa_{100}	30.6 [58.5]	51.8 [68.5]	57.1 [70.8]
$count_{25}$	32.6 [36.1]	31.8 [30.6]	27.7 [28.1]
$base_{pdf}$	5.2 [17.3]	5.5 [16.6]	5.5 [16.6]

Table 2: Macro [Micro] F1 Scores Across Experiments

Across the three data subsets, all ML models consistently outperformed the statistical and keyword-matching baselines $base_{pdf}$ and $count_{25}$ respectively. Notably, even with limited training data (in the 10% subset), most ML approaches surpassed $count_{25}$ which, to recall, emulates how lawyers may use keyword searches for legal area classification. Deep transfer learning approaches in particular performed well in this data-constrained setting, with $bert_{large}$, $bert_{base}$, and $ulmfit$ producing the best three macro-F1s. $ulmfit$ also achieved the second best micro-F1.

As more training data became available at the 50% and 100% subsets, the ML classifiers' advantage over the baseline models widened to around 30 percentage points on average. Word-embedding models in particular showed significant improvements. $glove_{avg}$ and $glove_{cnn}$ outperformed most of the other models (with F1 scores of 63.1 and 61.5 respectively). Within the embedding models, $glove_{cnn}$ generally outperformed $glove_{avg}$ while $glove_{max}$ performed significantly worse than both and thus appears to be an unsuitable pooling strategy for this task.

Most surprisingly, lsa_{250} emerged as the best performing model on both micro- and macro-averaged F1 scores for the 100% subset. The model also produced the highest micro-averaged F1 score across all three data subsets, suggesting that common labels were handled well. lsa_{250}'s strong performance was fuelled primarily by high *precision* rather than recall, as discussed below.

6.2 Precision

Subset	10%	50%	100%
$bert_{large}$	54.7 [65.8]	57.1 [59.7]	63.6 [64.3]
$bert_{base}$	41.4 [45.1]	48.1 [50.0]	61.4 [67.2]
$ulmfit$	49.3 [63.7]	46.6 [61.4]	48.7 [63.2]
$glove_{cnn}$	50.7 [69.8]	63.4 [68.5]	66.7 [72.9]
$glove_{avg}$	**62.5** [68.0]	67.0 [68.1]	64.8 [68.2]
$glove_{max}$	51.3 [65.1]	47.3 [56.6]	59.2 [68.6]
lsa_{250}	56.7 [76.1]	70.0 [81.1]	**83.4** [81.7]
lsa_{100}	52.3 [**77.2**]	**73.8** [**81.9**]	73.9 [**83.7**]
$count_{25}$	30.2 [26.4]	26.4 [19.8]	23.0 [17.8]
$base_{pdf}$	2.9 [10.0]	3.1 [9.5]	3.1 [9.5]

Table 3: Macro [Micro] Precision Across Experiments

As with F1 score, ML models outperformed baselines by large margins on precision. LSA models performed remarkably well here: except in the 10% subset, where $glove_{cnn}$ recorded the highest macro-precision, top results for both precision measures belonged to either lsa_{100} or lsa_{250}. Notably, on the 100% subset, lsa_{250} managed over 80% on micro- and macro-precision.

[4] Alternative strategies for selecting the 512 tokens trialed performed consistently worse and are not reported.

Subset	10%	50%	100%
$bert_{large}$	43.2 [51.7]	**59.0** [**68.5**]	61.6 [68.5]
$bert_{base}$	**50.0** [**66.1**]	58.7 [67.9]	54.2 [60.9]
$ulmfit$	46.1 [61.8]	48.4 [64.8]	52.6 [65.4]
$glove_{cnn}$	37.4 [56.0]	58.2 [65.8]	**62.3** [**68.8**]
$glove_{avg}$	28.7 [39.2]	56.5 [60.9]	62.1 [63.1]
$glove_{max}$	23.2 [37.2]	49.9 [63.6]	49.0 [58.5]
lsa_{250}	32.7 [54.4]	50.2 [62.8]	57.8 [66.5]
lsa_{100}	25.7 [47.0]	45.3 [58.9]	51.6 [61.4]
$count_{25}$	48.1 [56.9]	57.5 [66.3]	59.9 [66.9]
$base_{pdf}$	29.0 [64.5]	32.3 [67.7]	32.3 [67.7]

Table 4: Macro [Micro] Recall Across Experiments

6.3 Recall

LSA's impressive results, however, stop short at recall. A striking observation from Table 4 is that LSA and most other ML models did *worse* than $count_{25}$ on *both* micro- and macro-recall *across all data subsets*. Thus, a keyword-search strategy seems to be a simple yet strong baseline for identifying and retrieving judgments by legal area, particularly when recall is paramount and an ontology of area-related terms is available. To some extent this reflects realities in legal practice, where false negatives (missing relevant precedents) have greater potential to undermine legal argument than false positives (discovering irrelevant precedents).

Instead of LSA, the strongest performers here were the BERT models which produced the best micro- and macro-recall on the 10% and 50% subsets and $glove_{cnn}$ for the 100% training subset.

7 Discussion

We initially expected pre-trained language models, being the state-of-the-art on many non-legal NLP tasks, to perform best here as well. That an LSA-based linSVM would outperform both word-embedding and language models by many measures surprised us. How LSA achieved this is explored in Appendix A.4 which presents a sample of the (quite informative) topics extracted.

One caveat to interpreting our results: we focused on comparing the models' *out-of-box* performance, rather than comparing the models at their best (i.e. after extensive cross-validation and tuning). Specifically, the BERT models' inability to be fine-tuned on longer input texts meant that they competed at a disadvantage, having been shown only selected judgment portions. Despite this, BERT models proved competitive on smaller training subsets. Likewise, while $ulmfit$ performed well on the 10% subset (suggesting that it benefited from encoder pre-training), the model struggled to leverage additional training data and recorded only modest improvements on the larger training subsets.

Thus, our answer to the research question stated in Section 1 is a nuanced one: while state-of-the-art models do not clearly outperform traditional statistical models when applied *out-of-box* to legal corpora, they show promise for dealing with data constraints particularly if further adapted and fine-tuned to accommodate longer texts. This should inform future research.

8 Conclusion and Future Work

This paper comparatively benchmarked traditional topic models against more recent, sophisticated, and computationally intensive techniques on the legal area classification task. We found that while data scarcity affects all ML classifiers, certain classifiers, especially pre-trained language models, could perform well with as few as 588 labelled judgments.

Our results also suggest that more work can be done to adapt state-of-the-art NLP models for the legal domain. Two areas seem promising: (1) creating *law-specific* datasets and baselines for training and benchmarking legal text classifiers, and (2) exploring representation learning techniques that leverage transfer learning methods but scale well on long texts. For the latter, possible directions here include exploring different CNN architectures and their hyperparameters, using contextualized word embeddings, and using feature extraction methods on pre-trained language models like BERT (as opposed to fine-tuning them) so that they can be used on longer text inputs. As Lord Denning said in *Packer v Packer* [1953] EWCA Civ J0511-3:

> "If we never do anything which has not been done before, we shall never get anywhere. The law will stand whilst the rest of the world goes on; and that will be bad for both."

Acknowledgments

We thank the anonymous reviewers for their helpful comments and the Singapore Academy of Law for permitting us to scrape and use this corpus.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2.

Elliott Ash and Daniel L. Chen. 2019. Case vectors: Spatial representations of the law using document embeddings. In Michael Livermore and Daniel Rockmore, editors, *Computational Analysis of Law*. Sante Fe Institute Press. (forthcoming).

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Guido Boella, Luigi Di Caro, Llio Humphreys, and Livio Robaldo. 2012. Using legal ontology to improve classification in the eunomos legal document and knowledge management system. In *Semantic Processing of Legal Texts (SPLeT-2012) Workshop Programme*, page 13.

David J Carter, James J. Brown, and Adel Rahmani. 2016. Reading the high court at a distance: Topic modelling the legal subject matter and judicial activity of the high court of australia, 1903-2015. *University of New South Wales Law Journal*, 39(4):13001354.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Mohammad Hassan Falakmasir and Kevin Ashley. 2017. Utilizing vector space models for identifying legal factors from text. In *Legal Knowledge and Information Systems: JURIX 2017*. IOS Press.

Teresa Goncalves and Paulo Quaresma. 2005. Evaluating preprocessing techniques in a text classification problem. In *Proceedings of the Conference of the Brazilian Computer Society*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Hoa T. Le, Christophe Cerisara, and Alexandre Denis. 2018. Do convolutional networks need to be deep for text classification ? In *AAAI Workshops*.

Zhenyu Liu and Huanhuan Chen. 2017. A predictive performance comparison of machine learning models for judicial cases. *2017 IEEE Symposium Series on Computational Intelligence*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Scikit-Learn. 2017. *Truncated Singular Value Decomposition and Latent Semantic Analysis*. Retrieved 28 Aug 2018 at http://scikit-learn.org/stable/modules/decomposition.html#lsa.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. *Machine Learning and Knowledge Discovery in Databases*, pages 145–158.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450. Association for Computational Linguistics.

Vikash Singh. 2017. Replace or Retrieve Keywords In Documents at Scale. *ArXiv e-prints*.

Octavia-Maria Sulea, Marcos Zampieri, Mihaela Vela, and Josef Van Genabith. 2017. Predicting the law area and decisions of french supreme court cases. *RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning*.

Piotr Szymaski and Tomasz Kajdanowicz. 2017. A network perspective on stratification of multi-label data. In *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, volume 74 of *Proceedings of Machine Learning Research*, pages 22–35, ECML-PKDD, Skopje, Macedonia. PMLR.

Eric Talley and Drew O'Kane. 2012. The measure of a mac: A machine-learning protocol for analyzing force majeure clauses in m&a agreements. *Journal of Institutional and Theoretical Economics JITE*, 168(1):181–201.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ravi Kumar Venkatesh. 2013. Legal documents clustering and summarization using hierarchical latent dirichlet allocation. *IAES International Journal of Artificial Intelligence*, 2(1).

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Daniel Taylor Young. 2013. How do you measure a constitutional moment? using algorithmic topic modeling to evaluate bruce ackerman's theory of constitutional change. *Yale Law Journal*, 122:1990.

A Appendices

A.1 Data Parsing

The original scraped dataset had 6,839 judgments in PDF format. The PDFs were parsed with a custom Python script using the `pdfplumber`[5] library. For the purposes of our experiments, we excluded the case information section found at the beginning of each PDF as we did not consider it to be part of the judgment (this section contains information such as Case Number, Decision Date, Coram, Counsel Names etc). The labels were extracted based on their location in the first page of the PDF, i.e. immediately after the case information section and before the author line. After this process, 611 judgments that were originally unlabelled and one incorrectly parsed judgment were dropped, leaving the final dataset of 6,227 judgments.

[5] https://github.com/jsvine/pdfplumber

A.2 Label Mappings

Labels are a double-dash-delimited series of increasingly specific legal N-grams (e.g. "tort–negligence–duty of care–whether occupier owes lawful entrants a duty of care"), which denote increasingly specific and narrow areas of law. Multiple labels are expressed in multiple lines (one label per line). We checked the topic labels for consistency and typographical errors by inspecting a list of unique labels across the dataset. Erroneous labels and labels that were conceptual subsets of others were manually mapped to primary labels via the mapping presented in Table 5. Some subjectivity admittedly exists in the choice of mappings. However we were not aware of any standard ontology for used for legal area classification, particularly for Singapore law. To mitigate this, we based the primary label set on the Singapore Academy of Law Subject Tree which, for copyright reasons, we were unable to reproduce here.

It was only *after* this step that the top 30 labels were kept and the remaining mapped to "others". Figure 1 presents all 51 original labels and their frequencies.

A.3 Implementation Details on Models Used

All text preprocessing (tokenization, stopping, and lemmatization) was done using `spaCy` defaults (Honnibal and Montani, 2017).

A.3.1 Baseline Models

$count_m$ uses the FlashText algorithm for efficient exact phrase matching within long judgment texts (Singh, 2017). To populate the set of associated terms for each label, all sub-labels attributable to the label within the given training subset were first added as exact phrases. Next, the label itself was tokenized into unigrams. Each unigram was added individually to the set of associated terms unless it fell within a set of customized stopwords we created after inspecting all labels. The set is $\{and, law, of, non, others\}$.

Beyond $count_{25}$, we experimented with thresholds of 1, 5, 10, and 35 occurrences. F1 scores increased linearly as thresholds increased from 1

Primary Label	Alternative Labels
administrative_and_constitutional_law	administrative_law, adminstrative_law, constitutional_interpretation, constitutional_law, elections
admiralty_shipping_and_aviation_law	admiralty, admiralty_and_shipping, carriage_of_goods_by_air_and_land
agency_law	agency
arbitration	
banking_law	banking
biomedic_law_and_ethics	
building_and_construction_law	building_and_construction_contracts
civil_procedure	civil_procedue, application_for_summary_judgment, limitation_of_actions, procedure, discovery_of_documents
company_law	companies, companies-_meetings, companies_-_winding_up
competition_law	
conflict_of_laws	conflicts_of_laws, conflicts_of_law
contract_law	commercial_transactions, contract, contract_-_interpretation, contracts, transactions
criminal_law	contempt_of_court, offences, rape
criminal_procedure_and_sentencing	criminal_procedure, criminal_sentencing, sentencing, bail
credit_and_security	credit_and_securities, credit_&_security
damages	damage, damages_-_assessment, injunction, injunctions
evidence	evidence_law
employment_law	work_injury_compensation_act
equity_and_trusts	equity, estoppel, trusts, tracing
family_law	succession_and_wills, probate_&_administration, probate_and_administration
insolvency_law	insolvency
insurance_law	insurance
intellectual_property_law	intellectual_property, copyright, copyright_infringement, designs, trade_marks_and_trade_names, trade_marks, trademarks, patents_and_inventions
international_law	
non_land_property_law	personal_property, property_law, choses_in_action
land_law	landlord_and_tenant, land, planning_law
legal_profession	legal_professional
muslim_law	
partnership_law	partnership, partnerships
restitution	
revenue_and_tax_law	tax, revenue_law, tax_law
tort_law	tort, abuse_of_process
words_and_phrases	statutory_interpretation
res_judicata	
immigration	
courts_and_jurisdiction	
road_traffic	
debt_and_recovery	
bailment	
charities	
unincorporated_associations_and_trade_unions	unincorporated_associations
professions	
bills_of_exchange_and_other_negotiable_instruments	
gifts	
mental_disorders_and_treatment	
deeds_and_other_instruments	
financial_and_securities_markets	
sheriffs_and_bailiffs	
betting	_gaming_and_lotteries, gaming_and_lotteries
sale_of_goods	
time	

Table 5: Primary-Alternative mappings for raw dataset labels

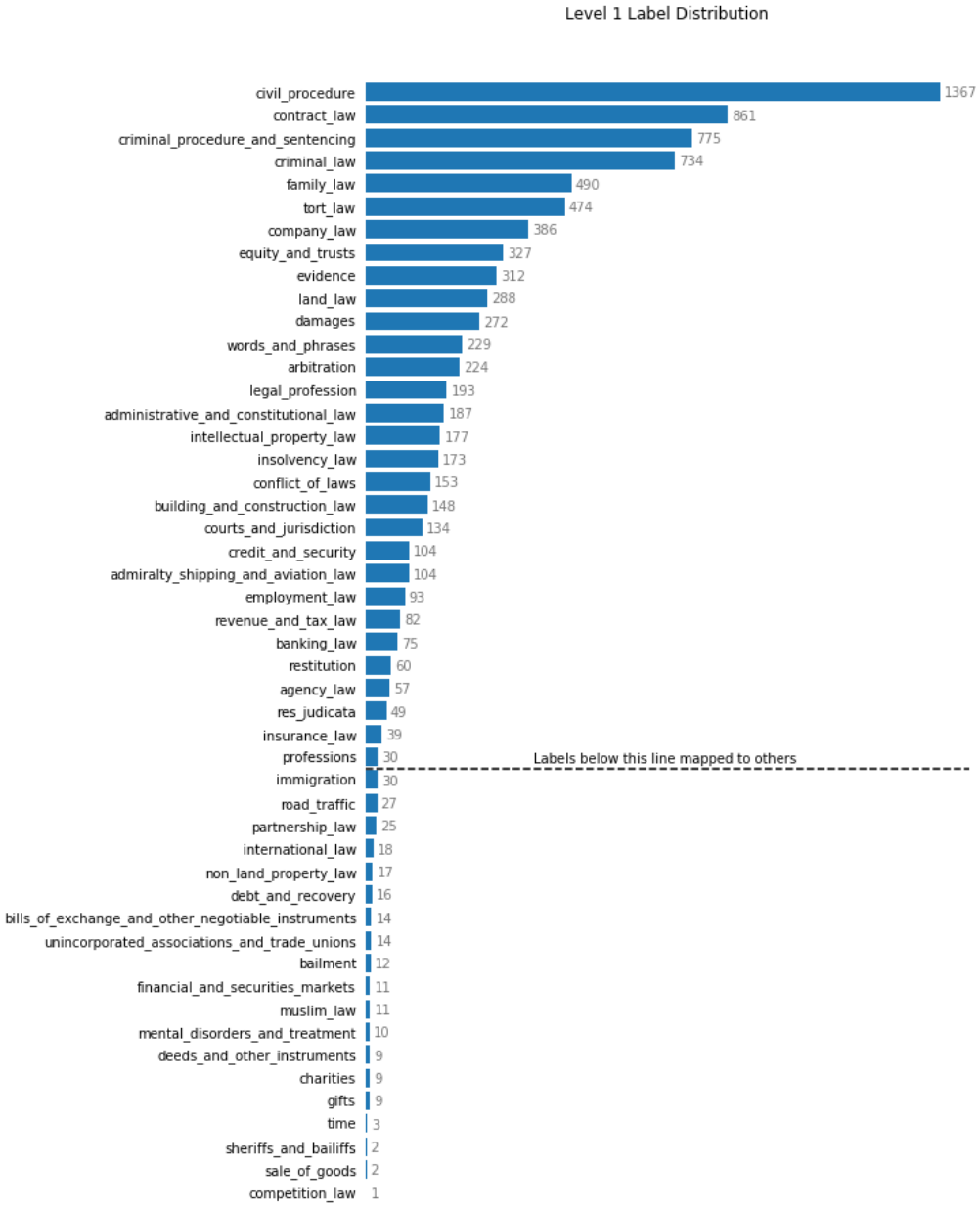

Figure 1: Distribution of Cleaned Labels and the Final 30 Labels Included

to 25, but only increased marginally from 25 to 35.

A.3.2 Topic Models

LSA was achieved using scikit-learn's TFIDFVectorizer and TruncatedSVD classes. Document-topic weights were then normalized with scikit-learn's Normalizer class before being fed to the classifier. Where relevant, the random state was set at 36. Note that judgments were preprocessed with spaCy as above before being fed into the LSA pipeline. Beyond 100 and 250 topics, an experiment using 50 topics only performed consistently worse.

The classifier used scikit-learn's OneVsRest and LinearSVC classes with all default settings. An alternative linSVM with balanced class-weights was tested but performed consistently worse by both macro and micro-f1 scores and was thus omitted for brevity.

A.3.3 Word Embedding Feature Models

For all the word embedding feature models, we used spaCy's tokenizer to obtain the word tokens. We fixed the maximum sequence length per judgment at 10K tokens and used a vocabulary of the top 60K most common words in the training corpus. Words that did not have a corresponding GloVe vector were initialized from a uniform distribution with range $[-0.5, 0.5]$. The models were implemented in TensorFlow[6] with the Keras API. To deal with class imbalance, we weighted the losses by passing class weights to the `class_weight` argument of `model.fit`.

For the CNN models, we based our implementation off the non-static version in Kim (2014) but used [3, 3, 3] x 600 filters, as we found that increasing the number of filters improved results.

A.3.4 BERT

To fine-tune BERT to our multi-label classification task, we used the PyTorch implementation of BERT by `HuggingFace`[7] and added a linear classification layer $W \in \mathbf{R}^{K \times H}$, where K is the number of classifier labels and H is the dimension of the pooled representation of the input sequence, followed by a sigmoid function. We fine-tuned all the BERT models using mixed-precision training and gradient accumulation (8 steps). To address data imbalance, we weighted the losses by passing positive weights for each class (capped at 30) to the `pos_weight` argument of `torch.nn.BCEWithLogitsLoss`.

A.3.5 ULMFiT

We first fine-tuned the pre-trained ULMFiT language model (WikiText-103) on our entire corpus of 6,227 judgments using a language model objective for 10 epochs before replacing the output layer with a classifier output layer and then further fine-tuned the model on labelled data with the classification objective using `fastai`'s recommended recipe[8] for text classification (we used gradual unfreezing and the one-cycle learning rate schedule to fine-tune the classifier until there was no more improvement on the validation score). We used mixed precision training and fixed the maximum sequence length at 5K tokens to allow the training data to fit in memory.

A.4 Topics Extracted by Topic Mining

Table 6 presents the top 10 tokens associated with the top 25 topics extracted by LSA on the 100% data subset. Notice that these topics are common to both lsa_{100} and lsa_{250} since the output of TFIDF and SVD do not vary with k. The only difference is that lsa_{100} uses only the first 100 topic vectors (i.e. the topic vectors corresponding to the 100 largest singular values computed by the decomposition) created by LSA whereas lsa_{250} uses the first 250. However, topics extracted from different data subsets would differ.

A quick perusal of the extract topics suggests many have would be highly informative of a case's legal area. Topics 2, 7, 21, 24, and 25 map nicely to criminal law, topics 3 and 5 to family law, and topics 18, and 20 to arbitration. Other individually-informative topics include topics 6 (road traffic), 8 (building and construction law), 9 (land law), 11 (legal profession), 16 (company law), and 22 (conflict of laws).

[6] https://github.com/tensorflow/tensorflow

[7] https://github.com/huggingface/pytorch-pretrained-BERT

[8] https://docs.fast.ai/text.html

Topic No.	Top 10 Tokens
1	plaintiff, court, defendant, case, party, claim, order, appeal, fact, time
2	offence, accuse, sentence, imprisonment, prosecution, offender, charge, drug, convict, conviction
3	matrimonial, husband, wife, marriage, child, maintenance, contribution, asset, cpf, divorce
4	application, court, appeal, order, district, matrimonial, respondent, proceeding, judge, file
5	matrimonial, marriage, child, maintenance, husband, divorce, parliament, division, context, broad
6	injury, accident, plaintiff, damage, defendant, dr, award, medical, work, pain
7	drug, cnb, diamorphine, packet, mda, bag, heroin, traffic, arbitration, plastic
8	contractor, contract, sentence, imprisonment, construction, clause, project, offender, cl, payment
9	property, land, purchaser, tenant, title, estate, decease, owner, road, lease
10	arbitration, victim, rape, sexual, arbitrator, arbitral, cane, clause, accuse, cl
11	disciplinary, profession, solicitor, committee, advocate, society, client, misconduct, lpa, professional
12	creditor, debt, bankruptcy, accident, debtor, wind, liquidator, injury, death, decease
13	plaintiff, defendant, proprietor, infringement, plaintiffs, defendants, cane, 2014, land, 2012
14	appellant, 2014, road, district, 2016, trial, defendant, property, judge, pp
15	drug, arbitration, profession, disciplinary, society, clause, vessel, death, advocate, diamorphine
16	shareholder, director, company, vehicle, share, resolution, traffic, management, vote, minority
17	creditor, solicitor, road, vehicle, profession, bankruptcy, disciplinary, drive, lane, driver
18	arbitration, adjudicator, decease, tribunal, adjudication, arbitral, vehicle, arbitrator, drive, mark
19	contractor, adjudicator, adjudication, decease, beneficiary, estate, employer, death, child, executor
20	arbitration, arbitrator, tribunal, award, arbitral, profession, contractor, disciplinary, architect, lpa
21	drug, respondent, appellant, diamorphine, gd, factor, cl, adjudicator, judge, creditor
22	2015, forum, 2014, 2016, 2013, foreign, 2012, appellant, conveniens, spiliada
23	stay, appellant, arbitration, estate, register, forum, district, beneficiary, owner, applicant
24	vessel, cargo, decease, murder, sale, ship, death, dr, kill, knife
25	sexual, rape, penis, vagina, complainant, stroke, intercourse, penetration, vessel, sex

Table 6: Top Tokens For Top 25 Topics Extracted by lsa_{250} on the 100% subset.