The 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)
Taipei, Taiwan, November 21-22, 2022. The Association for Computational Linguistics and Chinese Language Processing

Legal Case Winning Party Prediction With Domain Specific Auxiliary Models

Sahan Jayasinghe, Lakith Rambukkanage, Ashan Silva, Nisansa de Silva, Amal Shehan Perera

Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka
{sahanjayasinghe.17, rambukkanage.17, ashansilva.17, nisansaDds, shehan}@cse.mrt.ac.lk

Abstract

Sifting through hundreds of old case documents to obtain information pertinent to the case in hand has been a major part of the legal profession for centuries. However, with the expansion of court systems and the compounding nature of case law, this task has become more and more intractable with time and resource constraints. Thus automation by Natural Language Processing presents itself as a viable solution. In this paper, we discuss a novel approach for predicting the winning party of a current court case by training an analytical model on a corpus of prior court cases which is then run on the prepared text on the current court case. This will allow legal professionals to efficiently and precisely prepare their cases to maximize the chance of victory. The model is built with and experimented using legal domain specific sub-models to provide more visibility to the final model, along with other variations. We show that our model with critical sentence annotation with a transformer encoder using RoBERTa based sentence embedding is able to obtain an accuracy of 75.75%, outperforming other models.

Keywords: Natural Language Processing, Legal Domain, Case Law, Transformer Encoders

1 Introduction

Natural Language Processing (NLP) is undergoing rapid development and has proven to be practically useful across many text rich domains. With the proper utilization of tools and technologies, effective methodologies can be derived to tackle various problems that are repetitive, cognitively demanding and time consuming otherwise. Legal domain is such a text rich domain with a growing need for task automation. Legal domain corpora consists of statutes, regulations, constitutions and case law documents among many others which have to be repeatedly and constantly sifted through by legal professionals to obtain information pertinent to their current case. This research is primarily carried on Case Law documents where a model is train on a corpus of existing case law documents so that a prediction of the winning party in a current case law document can be obtained.

1.1 Case Law

In the legal domain, when confronted with a new case, where statues, regulations and constitutions cannot be used to straightforwardly arrive at a case decision, the courts refer to Case Law. Case Law is the practice of using the information and verdicts of previous cases as arguments for the case in hand where the older cases bear some semblance in one aspect or another to the contemporary case. (Cornell Law School, 2020a).

Since case law documents have a predictive, or rather a prescriptive value, in the domain itself, they are valuable resources for predictive tasks in both research and practical applications. As time goes on and more and more cases are closed, cases available to refer grow in abundance on a daily basis. For human legal professionals, this is a negative as it makes their task of remembering and referring to these cases increasingly hard. But on the perspective of deep learning models, this growth is a blessing rather than a hindrance as more and more data is gathered, the reliability and accuracy of the models increase. In this study, we have used case law documents to train our models.

1.2 Legal Party

In all legal cases two main parties are present (Cornell Law School, 2020b). One party corresponds to the party filing the case who is referred to as *petitioner* or *plaintiff*. In criminal cases they may also be referred to as the *prosecutor* which is a government entity. On the other hand, we have the party responding to the case which is referred to as the *defendant* or *respondent*. In criminal cases, this party may also be referred to as the *accused*. These parties may consist of individuals, groups of people, or organizations. Also there may be third parties in a case who are unaffected by the case decision. It is important to note that, in the case of an appeal, the party appealed will become the petitioner in the new case (Cornell Law School, 2020c). For the benefit of readability, for the rest of this paper, we will refer to the two parties as *petitioner* and *defendant*.

1.3 NLP in the Legal Domain

Recently many researchers have conducted legal domain specific researches. Among these, researches on legal domain specific embedding (Sugathadasa et al., 2017, 2018; Jayawardana et al., 2017a), legal ontology (Jayawardana et al., 2017a,c,b), sentiment analysis (Gamage et al., 2018; Ratnayaka et al., 2020), and discourse analysis (Ratnayaka et al., 2018, 2019b,a) can be observed. Also, granular objectives such as party identification (Samarawickrama et al., 2020; de Almeida et al., 2020; Samarawickrama et al., 2021), Party Based Sentiment Analysis (Rajapaksha et al., 2020; Mudalige et al., 2020; Rajapaksha et al., 2021), and critical sentence identification (Jayasinghe et al., 2021) have been explored among these researches. However, there is still the need and opportunity for these models to be used for higher level derivations that are more human readable or practically useful.

1.4 Winning Party Prediction

Legal professionals, among other preparations, go through case law documents in order to prepare for ongoing court cases. The use of case law documents during preparation and during the court case, gives the intuition that these documents contain a prescriptive values and can be used as a data source for predictions of court case decisions. Also in United States courts, all the facts that are to be brought up in the case is known in advance by both parties. With this, legal professional can prepare a document with arguments they are going to use and arguments their opposing party may use which is similar to a case law document. If this document can be given a benchmark, that is to predict if the case can be won by the given arguments and facts, it would be a valuable insight for legal professionals. They can revise their facts and arguments with inclusions, exclusions and introductions of new facts to increase their likelihood winning the case. Dorf (1994) observes by pointing to Holmes (1920) that this practice of trying to predict the outcome of a court case at hand predates any attempt at automation.

In this research we discuss a novel approach to predict the winning party of a court case using case law documents from the United States Supreme Court. The past work that have been carried out is discussed in Section 2. The formulation of our methodology is discussed in Section 3 and the experiments carried out and the achieved results are discussed in Section 4.

2 Related Work

In the work by Shaikha et al. (2020), they have categorized the past approaches to predict the outcome of a legal case into three categories. Three approaches are distinguished by the use of 1) political or social science based, 2) linguistics based or 3) legal domain based features as the descriptors for the machine learning algorithms they use. 19 features have been formalized with respect to the legal domain, that has the potential to impact the decision of a criminal court case. It is important to note that feature extraction is manually done by going through court cases, and therefore it requires experts to identify the features. After feature extraction and preprocessing, researchers have conducted classification under 8 different algorithms such as Regression Trees, Bagging and Random Forests, Support Vector Machines and K-nearest neighbours. Classification and Regression Trees have been found to be the best performing.

In the research by Waltl et al. (2017), they have conducted their research fundamentally on German tax law cases. The research is conducted on features extracted using mostly regular expressions and manual annotations. A Naive Bayes classifier have been chosen as the best performing machine learning model. They have achieved 0.57 precision, 0.58 F1 score and 0.60 recall for positive outcomes.

Research done by Aletras et al. (2016) on predicting the decision of the European court of human rights, is identified as the first systematic approach to predicting winning parties by using NLP, as per the authors. They have modeled the problem as a binary classification problem, while using Support Vector Machines and N-grams and topics as features for the model.

Liu and Chen (2017) also proposes a classification approach for identifying the winning party of a court case. The process consists of two phases. In the 1st phase, an Article Classification model extracts top k articles that are cited in the case document. In the 2nd phase, the Judgement Classification model tries to predict the judgement of the court case. They have considered domain specific aspects such as punishment, cited statutes and features derived using NLP such as sentiment, as features for their model.

A tree based approach which uses new feature engineering techniques is proposed in the research conducted by Katz et al. (2014). The dataset used in this research consist of cases from the United States Supreme Court. Researchers have considered the impact from political biases for the decisions as well. They have used data ranging over multiple presidential terms to generalize the model more. Features already present with there chosen dataset have been used and some has been introduced by them. With the 7700 cases used, they have succeeded in getting 69.7% accuracy and individual judge votes with 70.9% accuracy.

Lage-Freitas et al. (2019) have proposed a machine learning approach to develop a system that predicts Brazilian court decisions. Researchers have suggested for it to be used as a supporting tool or a benchmark for legal professionals. The approach to calculate both the decision class and the unanimity of decisions have been designed. They have achieved good accuracy for some of the many model variations.

3 Methodology

In this section, the approach used for dataset preparation and the methodology for deriving the architecture used in this research are be discussed.

3.1 Dataset Preparation

As observed by Kreutzer et al. (2022), the quality of the data sets used often play a vital role in research. This research was conducted on a dataset extracted from the case law website[1] ranging from the year 2000 to year 2010 and belonging to the criminal category. The extracted cases were pre-processed by removing paragraphs at the beginning and the end. These paragraphs include the introductory paragraphs where the background of the case is summarized and the last paragraphs where the decision is stated. Afterwards several preprocessing steps were applied to the remaining paragraphs to remove citations and other notations, as they do not add any semantic meaning to the case. In our data pair, these cleaned and remaining paragraphs constitute the input. Since the decision of each case was found in the aforementioned removed paragraphs with a retrievable convention in almost all the cases, the decision of the court cases were extracted automatically. In our data pair, this extracted verdict constitutes the expected output.

Stanza NLP Library (Qi et al., 2020) was used to split a court case document into a list of sentences as for the representation purposes discussed in Section 3.2. Since Stanza is a general purpose NLP library (not specifically trained on legal context), there could be sentences divided by the periods in between abbreviations (some of which are specific jargon of the legal domain) and the periods within brackets. So, further pre-processing steps were needed to be taken to make the sentence splitting process accurate.

- Removed text within rounded brackets.

[1] https://caselaw.findlaw.com/

- Replaced abbreviations specific to legal domain with their long form. As shown in the following examples:
  - Fed.R.Crim.P.→ Federal Rule of Criminal Procedure
  - Fed.R.Evid.→ Federal Rule of Evidence
- Removed square brackets around letters or words. (Ex: [T]he, Extend[ed], [petitioner])
- Removed numbering from topic sentences (Ex: II., A., 3.)

A case document in US Supreme Court is generally structured as follows:

- Background information of the Case (represented Jury, Date of Hearing)
- A description of the case scenario
  1. Involved parties and their members (petitioners and defendants)
  2. How the case is formed (cause for filing the case)
  3. Available Evidence
  4. Lower court decision (Where the case was initially called)
- Supreme Court hearing
  1. Charges against the petitioner (he/she is the defendant in the first hearing by lower court)
  2. Opinions of Jury
  3. Arguments brought forward by each party
- Footnotes

After case documents were labeled with the decision, notion of *winning* was defined with respect to the petitioner party. *Affirmation*, *dismissal* or *rejection* of a case by US Supreme Court results in petitioner losing the case. *Reversal* of the lower court decision results in the petitioner winning the case.

3.2 Model Architecture

The approach taken to predict the winning party of is discussed in this section. Each case document is represented as a sequence of sentences. The model takes the corresponding sentence vector sequences as input.

Dimensions containing additional information about a case sentence, such as the criticality of a sentence towards a party, can be annotated using *Critical Sentence Identification model* which is derived in the work by Jayasinghe et al. (2021). Given a case sentence, their system outputs probabilities for four classes which defines the criticality of the sentence within that court case.

1. Has a negative impact towards petitioner in a case where petitioner loses
2. Has a positive impact towards petitioner in a case where petitioner loses
3. Has a negative impact towards petitioner in a case where petitioner wins
4. Has a positive impact towards petitioner in a case where petitioner wins

A sentence is considered to be critical if it has a negative impact towards petitioner party in a case where petitioner loses. Also, a sentence which has a positive impact towards the petitioner party is considered critical in a case where petitioner wins. Sentences predicted with a high probability for other classes considered to be non-critical.

Probabilities for the four criticality classes provided by the *Critical Sentence Identification model* are appended to sentence vectors there by increasing the dimension. The impact of the addition is discussed in Experiments and Results section 4

The sentence vector sequence representing a court case document is then passed on to Document Encoder model which is configured by using Recurrent Neural Networks (RNN) or Transformer Encoder layers. The output of the Document Encoder model is used to obtain petitioner party winning probability via the classifier component. This classifier component is configured by using a Linear Neural Network. Linear neural network ends with a single-node layer which outputs the probability of petitioner party winning the case. The discussed overall workflow of the process is depicted in Fig. 1.

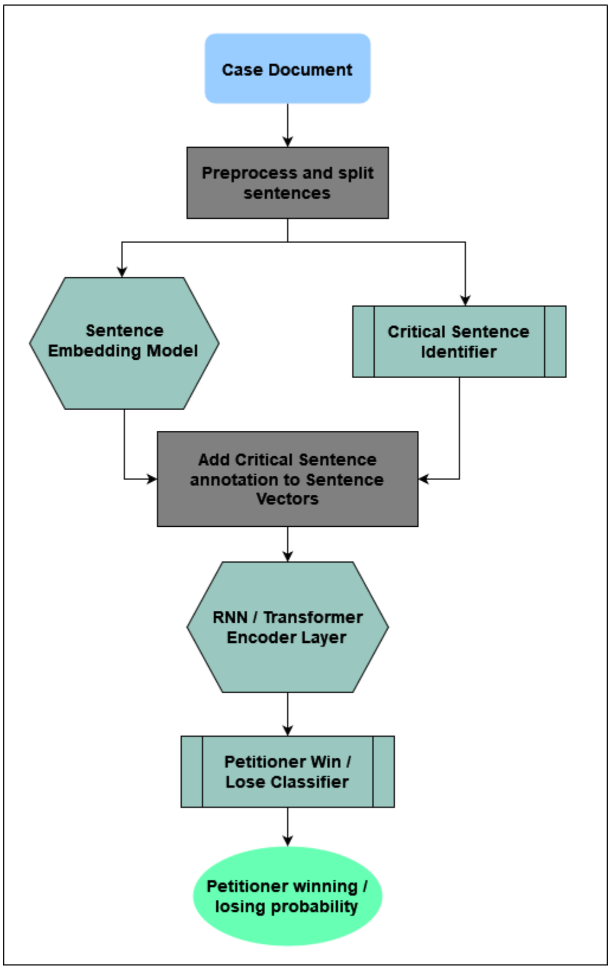

Figure 1: Winning Party Prediction Workflow

When the nature of a legal case is considered, often times the case is that the probability of Defendant party winning the case is equal to the probability of Petitioner party losing the case. There maybe cases for which it is not necessarily true, but we have followed that convention in this research.

The internal architecture for RNN based Wining Party Prediction model is displayed in Fig. 2 and for transformer encoder is displayed in Fig. 3.

In the RNN based model architecture (Fig. 2), Document Encoder consists of a single layer of either GRU (Chung et al., 2014) or LSTM (Hochreiter and Schmidhuber, 1997) where the final state vector is passed on to the classifier as the input. Classifier is built using a series of Dense Layers gradually down sized

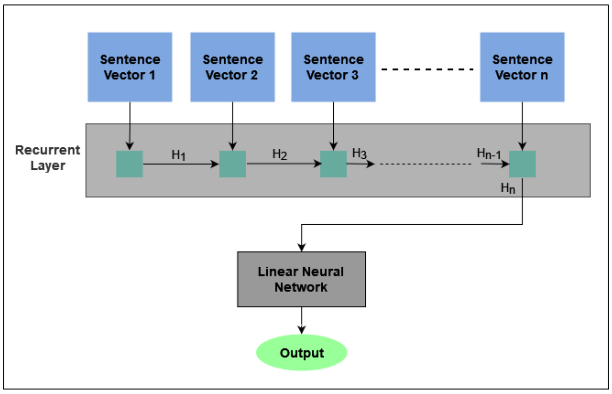

Figure 2: RNN based Model

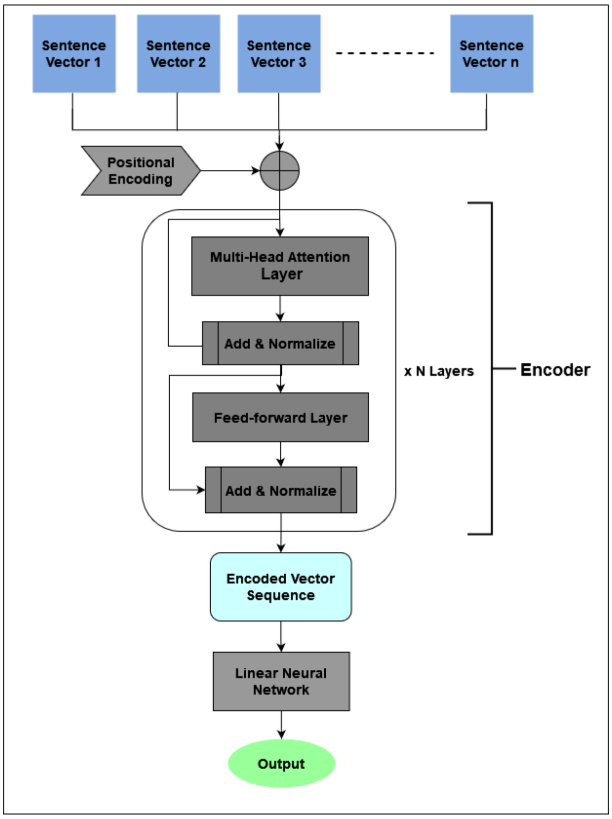

Figure 3: Transformer Encoder based Model

to a single node which is trained to predict the probability of winning of the petitioner party.

Transformer Encoder based model architecture (Fig. 3) is built using the encoder component of the original Transformer implementation (Vaswani et al., 2017). Document Encoder takes the sequence of sentence vectors as the input and adds the positional encoding to it. Positional encoding vector is calculated using the dimension of the input sentence vectors. Then the processed vector sequence is passed through a series of internal encoder layers. These encoder layers are duplicates of the same configuration and are built up of multi-head attention and position-wise feed forward layers. As per the definition of the Transformer Encoder by Vaswani et al. (2017), Multi-head attention layer is performing scaled-dot product on the input sequence. A normalization layer is used after multi-head attention layer and point-wise feed forward network to normalize the output vector of each layer. Global average pooling is used to reduce the 3-D output vector of the final encoder to a 2-D vector which is passed as input to the Classifier.

4 Experiments and Results

Experiments are performed by varying the Document Encoder model configurations and application of additional details to case sentences using the Critical Sentence Identification model (Jayasinghe et al., 2021). Document Encoder is experimented using different RNN configurations and Transformer Encoder configurations. To identify the number of layers best suitable for the transformer encoder, it was experimented with layers 6,3,2, and 1. As seen in the Fig 4, the best number of layers for the transformer encoder was found to be 1 in this case. RNN and Transformer Encoder components are used to encode the case documents. RNN models are experimented with both GRU and LSTM variations. Pre-trained *Sentence-BERT* by Reimers and Gurevych (2019), based on BERT (Devlin et al., 2018) and *DistilBERT* by Sanh et al. (2019),a distilled version of the RoBERTa-base (Liu et al., 2019), models are used for sentence embedding. Model building, training and evaluation are done using Tensorflow v2.8.

The following configurations were used for the Transformer Encoder:

- Number of Encoder layers = 1
- Number of Attention Heads = 8
- Vector Dimension = 768

Classifier model, which predicts the probability of petitioner winning takes the output from document encoder as the input and it is configured using a sequence of Dense Layers starting from 128 nodes.

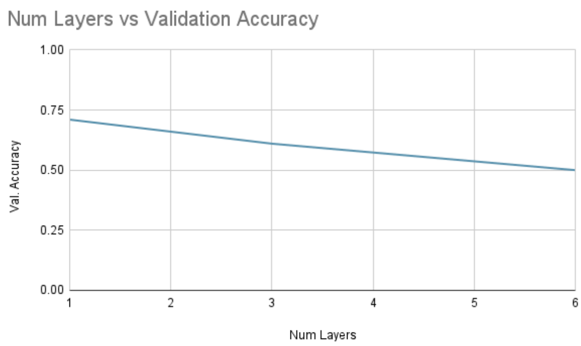

Figure 4: Number of Layers vs Validation Accuracy

Due to its suitability to handle datasets with imbalanced classes, Binary Focal Loss (Lin et al., 2017) is used to calculate the loss at each train step. At each training step, Focal Loss down-weights the loss for examples classified with higher accuracy of the dominant class and up-weights the loss for incorrectly classified examples of the minority class.

We summarize out findings in Table 1. It is curious to note that GRU with *Sentence-BERT* edges out the random baseline of 50% by only a narrow margin. This is an testament to the fact that the problem of *Winning Party Prediction* is non-trivial. The additional details provided by the critical sentence identification model (Jayasinghe et al., 2021), proved to be effective in predicting the winning party as per the results depicted in Table 1. This improvement is better visible in the case of GRUs than in the case of Transformers. Nevertheless, even with transformers, the improvement is relatively significant. *DistilBERT* (Sanh et al., 2019) embeddings have clearly outperformed pure *Sentence-BERT* (Reimers and Gurevych, 2019) configurations. The best performing configuration therefore is to use transformer encoders with *DistilBERT* sentence embeddings and the critical sentence annotation.

5 Conclusion and Future Work

Legal domain corpora carries its own complexities due to the domain nature. Therefore applying NLP in the legal domain requires domain specific approaches. In this study, we showed that our model with critical sentence annotation with a transformer encoder using RoBERTa based sentence embedding is able to

Model	Sentence Embedding	Critical Sentence Annotation	Accuracy	Macro F1
GRU	*Sentence-BERT*	N	56.32	53.14
	DistilBERT	N	65.71	57.14
	DistilBERT	Y	73.05	63.27
LSTM	*DistilBERT*	Y	72.04	65.52
GRU - Bidirectional	*DistilBERT*	Y	75.46	63.88
Transformer Encoder	*Sentence-BERT*	N	69.26	60.85
	DistilBERT	N	74.88	64.96
	DistilBERT	Y	**75.75**	**66.54**

Table 1: Winning Party Prediction Metrics

obtain an accuracy of 75.75%, outperforming other models. The need for domain-specific models can also be seen by the increase in accuracy when the critical sentence annotation is used. This system can be horizontally extended by adding more sub models to provide features to the final model. While the results obtained by *DistilBERT* (Sanh et al., 2019) sentence embeddings are impressive, extending the conclusions drawn by Sugathadasa et al. (2017) for word embeddings, it can be postulated that legal-domain specific sentence embeddings would potentially reap better results. Also as future work, the impact of having models trained with supervised approaches and unsupervised approaches should be experimented, as legal domain has a deficit of labeled data compared to its large corpora.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: a natural language processing perspective. *PeerJ Computer Science*, 2:e93.

Melonie de Almeida, Chamodi Samarawickrama, Nisansa de Silva, Gathika Ratnayaka, and Amal Shehan Perera. 2020. Legal Party Extraction from Legal Opinion Text with Sequence to Sequence Learning. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 143–148. IEEE.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cornell Law School. 2020a. Case law. https://www.law.cornell.edu/wex/case_law. Accessed: 2022-08-18.

Cornell Law School. 2020b. Legal party. https://www.law.cornell.edu/wex/party. Accessed: 2022-08-18.

Cornell Law School. 2020c. Petitioner. https://www.law.cornell.edu/wex/petitioner. Accessed: 2022-08-18.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Michael C Dorf. 1994. Prediction and the rule of law. *UCLA L. Rev.*, 42:651.

Viraj Gamage, Menuka Warushavithana, Nisansa de Silva, Amal Shehan Perera, Gathika Ratnayaka, and Thejan Rupasinghe. 2018. Fast Approach to Build an Automatic Sentiment Annotator for Legal Domain using Transfer Learning. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 260–265.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Oliver Wendell Holmes. 1920. The path of the law. *Collected Legal Papers*, pages 167–173.

Sahan Jayasinghe, Lakith Rambukkanage, Ashan Silva, Nisansa de Silva, and Amal Shehan Perera. 2021. Critical sentence identification in legal cases using multi-class classification. In *2021*

IEEE 16th International Conference on Industrial and Information Systems (ICIIS), pages 146–151. IEEE.

V. Jayawardana, D. Lakmal, Nisansa de Silva, A. S. Perera, K. Sugathadasa, B. Ayesha, and M. Perera. 2017a. Word Vector Embeddings and Domain Specific Semantic based Semi-Supervised Ontology Instance Population. *International Journal on Advances in ICT for Emerging Regions*, 10(1):1.

Vindula Jayawardana, Dimuthu Lakmal, Nisansa de Silva, Amal Shehan Perera, Keet Sugathadasa, and Buddhi Ayesha. 2017b. Deriving a Representative Vector for Ontology Classes with Instance Word Vector Embeddings. In *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*, pages 79–84. IEEE.

Vindula Jayawardana, Dimuthu Lakmal, Nisansa de Silva, Amal Shehan Perera, Keet Sugathadasa, Buddhi Ayesha, and Madhavi Perera. 2017c. Semi-Supervised Instance Population of an Ontology using Word Vector Embedding. In *Advances in ICT for Emerging Regions (ICTer), 2017 Seventeenth International Conference on*, pages 1–7. IEEE.

Daniel Martin Katz, Michael J Bommarito II, and Josh Blackman. 2014. Predicting the behavior of the supreme court of the united states: A general approach. *arXiv preprint arXiv:1407.6333*.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

André Lage-Freitas, Héctor Allende-Cid, Orivaldo Santana, and Lívia de Oliveira-Lage. 2019. Predicting brazilian court decisions. *arXiv preprint arXiv:1905.10348*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Yihung Liu and Yen-Liang Chen. 2017. A two-phase sentiment analysis approach for judgement prediction. *Journal of Information Science*, 44.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Chanika Ruchini Mudalige, Dilini Karunarathna, Isanka Rajapaksha, Nisansa de Silva, Gathika Ratnayaka, Amal Shehan Perera, and Ramesh Pathirana. 2020. Sigmalaw-absa: Dataset for aspect-based sentiment analysis in legal opinion texts. *arXiv preprint arXiv:2011.06326*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Isanka Rajapaksha, Chanika Ruchini Mudalige, Dilini Karunarathna, Nisansa de Silva, Gathika Rathnayaka, and Amal Shehan Perera. 2020. Rule-based approach for party-based sentiment analysis in legal opinion texts. *arXiv preprint arXiv:2011.05675*.

Isanka Rajapaksha, Chanika Ruchini Mudalige, Dilini Karunarathna, Nisansa de Silva, Amal Shehan Perera, and Gathika Ratnayaka. 2021. Sigmalaw PBSA-A Deep Learning Model for Aspect-Based Sentiment Analysis for the Legal Domain. In *International Conference on Database and Expert Systems Applications*, pages 125–137. Springer.

G. Ratnayaka, T. Rupasinghe, Nisansa de Silva, M. Warushavithana, V. Gamage, M. Perera, and A. S. Perera. 2019a. Classifying Sentences in Court Case Transcripts using Discourse and Argumentative Properties. *ICTer*, 12(1).

Gathika Ratnayaka, Thejan Rupasinghe, Nisansa de Silva, Viraj Gamage, Menuka Warushavithana, and Amal Shehan Perera. 2019b. Shift-of-Perspective Identification Within Legal Cases. In *Proceedings of the 3rd Workshop on Automated Detection, Extraction and Analysis of Semantic Information in Legal Texts*.

Gathika Ratnayaka, Thejan Rupasinghe, Nisansa de Silva, Menuka Warushavithana, Viraj Gamage, and Amal Shehan Perera. 2018. Identifying Relationships Among Sentences in Court Case Transcripts Using Discourse Relations. In *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 13–20. IEEE.

Gathika Ratnayaka, Nisansa de Silva, Amal Shehan Perera, and Ramesh Pathirana. 2020. Effective approach to develop a sentiment annotator for legal domain in a low resource setting. *arXiv preprint arXiv:2011.00318*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Chamodi Samarawickrama, Melonie de Almeida, Amal Shehan Perera, Nisansa de Silva, and Gathika Ratnayaka. 2021. Identifying legal party members from legal opinion texts using natural language processing. Technical report, EasyChair.

Chamodi Samarawickrama, Melonie de Almeida, Nisansa de Silva, Gathika Ratnayaka, and Amal Shehan Perera. 2020. Party Identification of Legal Documents using Co-reference Resolution and Named Entity Recognition. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 494–499. IEEE.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Rafe Athar Shaikha, Tirath Prasad Sahua, and Veena Anand. 2020. Predicting Outcomes of Legal Cases based on Legal Factors using Classifiers. *Procedia Computer Science 167 (2020) 2393—2402*.

Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. 2017. Synergistic Union of Word2Vec and Lexicon for Domain Specific Semantic Similarity. *IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6.

Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. 2018. Legal Document Retrieval using Document Vector Embeddings and Deep Learning. In *Science and Information Conference*, pages 160–175. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Bernhard Waltl, Georg Bonczek, Elena Scepankova, Jörg Landthaler, and Florian Matthes. 2017. Predicting the outcome of appeal decisions in germany's tax law. In *Electronic Participation*, pages 89–99, Cham. Springer International Publishing.