Recognising legal characteristics of the judgments of the European Court of Justice: Difficult but not impossible

Alessandro CONTINI [a] Sebastiano PICCOLO [a,1], Lucia LOPEZ ZURITA [a], and Urska SADL [a]

[a] *Copenhagen University, Faculty of Law, iCourts*

Abstract. Machine learning has improved significantly during the past decades, with computers performing remarkably in formerly difficult tasks. This article reports the preliminary results on the prediction of two legally relevant characteristics of judgments of the European Court of Justice, doctrinal outcome and deference. The variables require the knowledge of concepts and doctrines of European Union law and judicial decision-making. The analysis relies on 1704 manually labelled judgments and trains a set of classifiers based on word embedding, LSTM, and convolutional neural networks. The preliminary findings suggest that all classifiers exceed simple baselines, learning meaningful representations of the judgments. Yet, the overall performance is rather weak, suggesting that the usual limitations of machine learning, including small training data, significant class imbalance and the characteristics of the variables requiring external knowledge pose a significant challenge to machines learning the law.

The article also outlines directions for future research.

Keywords. Classification, European Court of Justice, Word embedding, LSTM, CNN

1. Introduction

Deep neural networks (DNN) and computer hardware have expanded the range of tasks that machines can accomplish better than humans [1], including the remarkable progress in computer vision [2], natural language processing [3] and game playing [4]. Artificial intelligence is also transforming the legal profession by providing tools for implementing computational legal reasoning, and by enabling deep analyses and argument extraction from legal texts [5,6].

In short, legal scholars are reaching out to computer scientists to understand law and institutions. Nonetheless, legal doctrine has remained a safe space of a law professor. Only lawyers can tell people which rights they can claim and whether it makes sense to litigate. The widespread belief remains that the knowledge of law is a distinctively human domain, involving a deep understanding of legal sources, situation sense, the ability to read legal texts between the lines and construct the systems of knowledge with the proverbial legal intuition [7].

[1] Corresponding Author: Sebastiano Piccolo, jvt612@ku.dk.

Or has it? The progress of machine learning and its wide application to the legal profession and the justice system raises the question whether machine learning can contribute to the legal analysis of judicial decisions. In other words, can deep learning models recognize the contribution of judgments (judicial decisions) to legal doctrine? Legal scholars define legal doctrine as a set of rights and duties, concepts and principles.

The article approaches the application of machine learning to the legal domain by considering two concepts of utmost importance for legal scholars: doctrinal contribution and deference. The aim is to develop an algorithm that can accomplish two tasks by taking as an input the text of a judgment:

1. Predicting if the Court, through said judgment, makes a strong contribution to the legal doctrine. This can happen, for instance, when the Court creates new concepts or principles.
2. Detecting if the Court, through said judgment, defers the final decision to the national court or the legislator; that is, whether it gives the national judge leeway as per the final decision/outcome of the case.

The first task requires legal training and an intimate knowledge of both the general principles and the specific areas of the European Union law (legal doctrine). It is a difficult task. The second task relies more on the text / wording of the judgment and should therefore be simpler than the first task.

The two tasks are approached as two separate binary classification problems. The classifiers are trained based on word embedding, LSTM, and CNN (convolutional neural network). In order to optimize a set of hyperparameters for the classifiers, we used ParEGO (Pareto Efficient Global Optimization) [8], an evolutionary algorithm that approximates the Pareto front, which offers a good balance between the goodness of the solution and computational time. We compared the performance of the classifiers by: *1)* training the classifier on the full judgment (typically between 3000 and 5000 words), and *2)* training the classifiers on paragraphs of the judgments where the classifier learns to predict whether a given paragraph is found in a judgment that makes a strong doctrinal contribution (respectively a deference to national courts). The latter attempts to accelerate the training stage, feeding the algorithm a much shorter text. The final prediction, on the judgment level, is obtained by aggregating the prediction of each paragraph through *1)* majority vote and *2)* single positive paragraph – i.e. if one paragraph in a judgment is labelled as strong doctrinal contribution or deference the judgment is labelled accordingly.

The preliminary findings indicate that predicting a strong doctrinal contribution is more difficult than detecting deference to national courts, as expected. The best performing model on predicting deference has F1=0.463, while the best performing model on predicting doctrinal outcome has F1=0.376. In general, the classifiers based on LSTM perform better than those based on CNN. The strategy of predicting single paragraphs and aggregating their scores is suboptimal, at best. The performance is around 25% worse than the performance of a classifier trained on the full judgment. This finding echoes the observation from Habernal et al. [6] that, in order to label arguments, legal experts rely on the context beyond the single paragraph used as input for their algorithm.

The performance of our algorithms is generally weak. This is likely due to a combination of factors such as: the small training dataset, the class imbalance, and the fact that the selected variables require extensive knowledge of complex legal concepts and legal doctrine.

2. Data and Methods

2.1. Dataset

The dataset includes 1704 Judgements issued by the Court of Justice of the European Union between 1954 and 2020. All judgements are in English and freely available from the official portal of the European Union Eur-lex. Content-wise, the judgments concern the free movement of goods and the free movement of persons. Both areas are highly salient in the European internal market and central to the process of European economic integration [9,10]. The Court fashioned the fundamental principles of European Union law and developed its central doctrines in judgments of these two areas. Moreover, the judgments speak of the Court's willingness to shape EU law, a subject of heated and prolonged debate in legal scholarship [11,12]. Specially relevant within this debate was the discussion of whether the Court sought to intervene in national legal systems, or if it left the key decisions to the courts and the legislators of the Member States. Each judgement was labelled by a legal expert (a human coder) specifying whether the judgment contributed to doctrine (Doctrinal Outcome or DOCOUT) and whether the Court deferred the final decision to the national courts or legislators (Deference or DEF).

The data is divided in two datasets: the first contains the full judgements and their relative predicted label. The second contains single paragraphs composing the judgments, each one with assigned label predicted for their judgement. Compared to similar researches the dataset is small: Wei et al. [13] sampled from a dataset composed by millions of judegments; Xiao et al. [14], in 2018, used a dataset composed by 2.6 million criminal cases published by the Supreme People's Court of China. Small training data is commonly known to result in poor performance. Simple models tend to under-fit the small training dataset, whereas complex models are likely to over-fit the training data. Deep learning models require a lot of training data because of the huge number of parameters needed to be tuned by a learning algorithm. Compared to traditional Machine Learning methods, the performance in Deep Learning increases with the amount of data [15]. However, it is extremely costly to produce hand coded training sets. It requires expert legal knowledge and time to process long court outputs (a typical judgment of the Court is often 3000 - 5000 words long and legally complex).

2.2. Variables

Doctrinal Outcome, the first variable of interest, relates to the Court's law-making activity in the narrow / legal doctrinal sense. Doctrine is defined as a set of rules and principles, which determine the scope and the content of rights and duties. The coding will be difficult to reproduce by non-lawyers because it requires legal training and the familiarity with the general principles of European Union law and the specific area of law, like the free movement of persons or goods. Thus, the coding relies on the opinion of legal experts and lawyers. There are two possible outcomes: weak (=0) and strong (=1). The Court can entrench, strengthen or expand its doctrines, create new concepts or develop principles (DOCOUT=1). By contrast, it can moderate its strong doctrinal positions or restate and apply established doctrines, concepts and principles, without further extending their scope (DOCOUT=0). **Deference** indicates whether the Court defers the final decision to the national court or the legislator; that is, whether it gives the national judge

Table 1. Number of negative samples for each positive one in the train and test sets

Label	Dataset	Neg/Pos ratio
DOCOUT	Full Judgments	5.642
	Paragraphs	5.382
DEF	Full Judgments	5.587
	Paragraphs	4.747

(a) Training set

Label	Dataset	Neg/Pos ratio
DOCOUT	Full Judgments	3.740
	Paragraphs	4.361
DEF	Full Judgments	4.953
	Paragraphs	3.755

(b) Test set

leeway as per the final decision/outcome of the case. The following language is indicative of the existence of deference (DEF=1): 'it is for the referring court to decide / establish / determine / examine', 'the national court must provide or decide'. When there are no references to the national courts, the outcome is DEF=0. Compared to the doctrinal outcome variable, which requires a degree of interpretation from the human coder, the hand coding of deference relies more on the language of the judgment.

2.3. Processing

The datasets obtained were randomly divided in a training set and a test set with 85% and 15% of the data. The training set was furthermore randomly split into 4 cross validation folders. For purposes of reproducibility and validation (cross-checking), the random split was fixed with a seed. The same seed is used to obtain all the following results and scores. Importantly, both labels presented a high imbalance in terms of negative and positive examples, as shown in Table 1. Moreover, the uneven (different) number of paragraphs in each judgment creates different imbalance ratios according to the dataset. This variability motivated the inclusion of the weight of a correct classified positive example as a parameter to be fine tuned. Class weights (assigning different weight to the two classes) are the simplest method to counter class imbalance. The number of samples in the classes is considered while computing the class weights. More significant weight to the minority class in the dataset places more emphasis on that class.

2.4. Baseline and Metrics

A majority class classifier over the Test set serves as a baseline for the identification of the possibility to successfully learn from the available judgments. Table 2 shows the performance of the "dummy classifier", which classifies every data object as the majority class, in this case 0. The dummy classifier is equivalent to a random guess in terms of ROC-AUC and F1 score. Comparable studies rely only on accuracy [13]. Table 2 show that the accuracy is almost the same as Wei et al. [13], meaning that the measure is insufficient to asses the performance of the classifier with an imbalanced dataset. By contrast, F1 score, widely used in NLP tasks [16], is more useful as it consist in the harmonic mean of precision and recall. ROC-AUC is also a good metric for imbalanced datasets [17].

2.5. Text Preprocessing

Text pre-processing methods including data transformation and filtering can significantly enhance the performance of classifiers. [18] The judgments were thus pre-processed, in-

Table 2. The majority class classifier on the Test Set.

Label	Dataset	F1	ROC-AUC	ACC
DOCOUT	Full Judgments	0	0.5	0.790
	Paragraphs	0	0.5	0.813
DEF	Full Judgments	0	0.5	0.719
	Paragraphs	0	0.5	0.790

cluding lowercasing and removal (filtering) of punctuation and numbers. Word embedding was used to codify each word in a vector of numbers, restricting the vocabulary to 30000 words. Word embedding trained on multiword-grouped corpora performed well, and was hence applied to simple tokenized datasets. [19] Usually, the removal of stopwords reduces the amount of irrelevant text; however, in our case, it only appears to slightly lower the performance of our models. As such, we did not remove stopwords.

2.6. Models

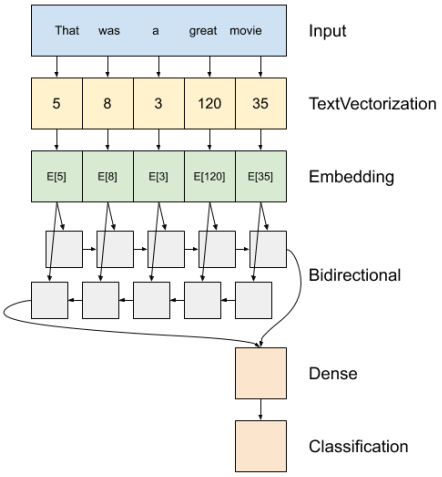
Neural networks (NNs) [20] are based on artificial neurons, connected in many layers to create a network. The input layer includes as many neurons as there are features and the output layer as many neurons as there are output possibilities. Other hidden layers may be included between the two. This article implemented and trained two different models: a Convolutional Neural Network (CNN) and a Bidirectional Long short-term memory Recurrent Neural Network (LSTM) . CNN and LSTM, have been shown to be excellent methods in text classification [21].

We trained our models by using the Adam optimization algorithm and Binary Cross Entropy as loss functions. We trained our models until convergence, using an early stopping criterion that monitored the F1 score. This is a good strategy to prevent over-fitting and to save some computational time. We also added dropouts after every hidden layer to further reduce the chances of over-fitting. Dropout is more effective than other standard computationally expensive regularizers [22]. Deep Neural Networks (DNN) with dropout often preform better than DNN without dropout. [15]

Our models exhibit the structure of the classifier in Figure 1. First there is a Text Vectorisation layer followed by an Embedding layer. The Embedding layer will learn a vectorial space where similar words, or words that appear in similar contexts, are at a closer distance than words that appear in different contexts. Next, there is a Bidirectional LSTM layer. This layer *reads* the text sequentially and is therefore able to detect sequential dependencies as well as *remember* past information and context. Finally, we have a variable number of Dense layers, using ReLu activation function and a final output layer implementing a Sigmoid activation function. The models based on CNN follows the same structure, with the difference that a CNN substitutes the bidirectional LSTM, and the CNN layer is followed by a max pooling layer.

The number of hidden Dense layers and their dropout rate, as well as the size of the embedding, the number of neurons, the learning rate and the weight of every positive example are computed through the ParEGO hyperparameter tuning algorithm [8]. Each hidden layer implements dropout as a means of regularization.

Figure 1. The structure of the used LSTM RNN with a single dense layer. The CNN follows the same structure, changing the Bidirectional layers with a Convolutional and a Max Pooling one. Image taken from [23]



2.7. Hyperparameter Tuning

As previously noted, deep learning methods have many parameters – such as the number of hidden layers, the number of neurons in each layer, as well as training parameters such as learning rate – that need to be selected. The tuning of these parameters can offer a substantial increase in the model performance [24]. However, an exhaustive search of the best combination of values for a set of parameters is prohibitive. As such, we used the Pareto Efficient Global Optimization Algorithm: ParEGO[8] to search the space of hyperparameters. Though a series of initial random trials, using random runs and a 70/15/15 split, we identified promising ranges of values for each hyperparameter. On these ranges, we performed the search with ParEGO, in order to identify the best combination of parameters. The algorithm was selected as it offers a good trade-off between time and performance in situations of expensive evaluations [8]. Table 3 presents the values, obtained by ParEGO, for each of the hyper-parameter we tuned.

Table 3. Parameters found by ParEGO for each variable to predict, dataset (whether we use the full judgments or the paragraphs), and network type. Values are rounded to the third most significant digit.

Variable	DOCOUT				DEF			
Dataset	Full Judgments		Paragraphs		Full Judgments		Paragraphs	
Net Type	LSTM	CNN	LSTM	CNN	LSTM	CNN	LSTM	CNN
Emb. dimension	36	136	204	197	30	113	38	183
Num. units	41	42	167	47	26	31	31	49
Learning rate	1.89e-4	3.9e-5	2.5e-5	4.44e-4	1.54e-4	3.9e-5	9e-3	1.47e-3
Layers Num	0	1	3	1	0	1	1	1
Pos. weight	2.775	1.13	5.02	1.21	1.09	1.129	2.662	1.165
Dropout Conv	-	0.133	-	0.046	-	0.202	-	0.208
Dropout Dense	0.079	0.203	0.2	0.226	0.14	0.219	0.0293	0.319
Conv. Kernel	-	11	-	8	-	8	-	6

Table 4. Performance of our models on classifying full judgments or single paragraphs for doctrinal outcome (DOCOUT) and deference (DEF)

			Cross validation		Test-set	
Label	Dataset	Net Type	F1	ROC	F1	ROC
DOCOUT	Full Judgments	LSTM	0.317	0.682	0.376	0.649
		CNN	0.311	0.663	0.316	0.66
	Paragraphs	LSTM	0.570	0.870	0.569	0.884
		CNN	0.550	0.849	0.539	0.853
DEF	Full Judgments	LSTM	0.372	0.715	0.463	0.639
		CNN	0.380	0.700	0.342	0.677
	Paragraphs	LSTM	0.574	0.801	0.605	0.840
		CNN	0.589	0.837	0.605	0.863

3. Results

Overall, the performance of our models are weak. The F1 and ROC are reported in Table 4 for each of the variables we predicted, the type of model considered, and the type of prediction – i.e. classifying the full judgment or classifying single paragraphs. We present results from both the cross validation and the retained test set. The comparison between the results on cross validation and the test set suggest that we are not overfitting the data. However, in the case of the LSTM on full judgments, for both DOCOUT and DEF, we note that performance on the test set are higher than those on the cross validation. This might indicate some underfitting.

The classifiers based on LSTM have in general better performance than the classifiers based on CNN. The performance on the prediction of deference are higher than the performance on the prediction of doctrinal outcome. This indicates that predicting deference is an easier task than predicting doctrinal outcome. Finally, predicting whether a paragraph belongs to a judgments that makes a strong doctrinal contribution (respectively defers the decision to a national court) is an easier task than predicting whether the full judgment makes a strong doctrinal contribution (respectively defers the decision to a national court).

However, this superiority is illusory. We used the best obtained classifiers on paragraphs to aggregate the predictions on the single paragraphs into predictions on the level of the judgments. We used two aggregation techniques: *1)* the classical majority vote, where a judgment is predicted to make a strong doctrinal contribution (respectively deference) if the majority of the paragraphs of the judgment is predicted to be part of a judgment that makes a strong doctrinal contribution (respectively deference); and *2)* the 'at least one positive' aggregation, where a judgment is predicted to make a strong doctrinal contribution (respectively deference) if at least one of the paragraphs of the judgment is predicted to be part of a judgment that makes a strong doctrinal contribution (respectively deference). The results, in terms of F1 score, are presented in Table 5. The first thing we note is that the overall performance is significantly worse than the performance obtained by directly classifying the full text. Second, we note that LSTM seems to fail completely in the majority vote, while having equal performance with the CNN on the 'one positive' aggregation. This happens because CNN classifies many more paragraphs than LSTM as positive thus having a very high recall at the expenses of precision.

Table 5. F1 scores of prediction obtained by aggregating the predictions of single paragraphs for doctrinal outcome and deference

Label	Net Type	Majority Voting	One positive
DOCOUT	LSTM	0	0.275
	CNN	0.21	0.279
DEF	LSTM	0.04	0.37
	CNN	0.369	0.369

4. Discussion

In this paper we considered two tasks: predicting if a judgment makes a strong doctrinal contribution and classifying whether the European Court of Justice defers the final decision to a national court. We faced a number of difficulties: our training dataset is small, there is a substantial class imbalance, and predicting doctrinal outcome is likely to require extensive knowledge of complex legal concepts and legal doctrine. The performance of our models, as noted before, are weak. However, we still learn many good and encouraging lessons from our findings, which prompt us to push our research forward.

First, our classifiers learn a meaningful representation of the judgments. This is evident from the fact that the performance are significantly better than the performance with a random classifier, or a dummy classifier. Furthermore, our models do not show any sign of overfitting; rather, our classifiers might actually underfit the training data. This is somehow encouraging, because it means that we still have space to improve our performance. Finally, LSTM appears to perform better than CNN on every task. In future research, we plan to push our results forward by increasing the size of the training data (which implies more hand coding of the data) as it is known that performance of deep learning models increases with the size of the training data [15], and by exploring more complex models: from multiple LSTMs in sequence, to encoder-decoder architectures, and more recent BERT-based models.

Second, predicting deference is easier than predicting doctrinal outcome (Table 4). This is expected, for many reasons. Deference has a lower class imbalance than doctrinal outcome. In fact, for the LSTM on the full judgments, the positive weight for deference selected by ParEGO (Table 3) is 1.09, as opposed to 2.775 for the doctrinal outcome. Additionally, from the legal perspective, it is more difficult to identify a strong or weak doctrinal outcome than a deferential outcome. The former is often implicit in the text, and often a matter of scholarly analysis rather than an information contained in the text of the judgment. [25].

Third, the idea of processing the whole text by paragraph, obtaining the predictions on paragraphs ad aggregating said predictions into a single one on the judgment level yields sub-optimal performance. In other tasks, such as sentiment analysis, prediction on short sentences yields usually superior results [26]. This is not the case for the legal domain. Judgments are complex texts where the context beyond a single paragraph is important and, as noted in [6], it is often used by legal experts to label arguments. As such, in order to classify legal texts, we need to cope with long sequences. Besides the already mentioned encoder-decoder architectures, other ideas worth of further investigation are *1)* feeding multiple paragraphs in parallel to the prediction algorithm, thus training a network with multiple inputs, and *2)* summarising/filtering the judgments to retain only the most salient parts of the text.

5. Conclusions

The article investigated whether machine learning could contribute to the legal analysis of judicial decisions by predicting two legally interesting and demanding outcomes: doctrinal outcome and deference. It trained a set of classifiers based on word embedding, LSTM, and CNN on a dataset of manually labelled judgments of the European Court of Justice, showing that the classifiers can learn a meaningful representation of the judgments. In other words, they achieve better predictions than random classifiers in terms of F1 score and ROC-AUC. Among the classifiers, LSTM performs better than CNN. Additionally, it is not easy to cope with long legal texts. The judgments were thus split and processed in paragraphs, predicting the outcomes for single paragraphs and aggregating the single scores into a prediction on the whole judgment level. This strategy yielded suboptimal performance. The findings suggest that NLP on long texts is still an open problem, which affects the legal domain.

Further analysis and experimentation would be required to fully understand the significance of these results. These include: developing more sophisticated models, incorporating more hand-coded judgements, and finding ways to deal with long text sequences. This work can be viewed as a starting point for studying the impact of text classification and the potential of deep learning models in very specific NLP fields, such as the legal domain. At the same time, it suggests that the legal experts remain the final authority when it comes to legal doctrine. The law professor is safe, for now.

References

[1] E. Brynjolfsson and T. Mitchell, "What can machine learning do? workforce implications," *Science*, vol. 358, no. 6370, pp. 1530–1534, 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[3] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[4] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[5] K. D. Ashley, *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press, 2017.

[6] I. Habernal, D. Faber, N. Recchia, S. Bretthauer, I. Gurevych, C. Burchard, *et al.*, "Mining legal arguments in court decisions," *arXiv preprint arXiv:2208.06178*, 2022.

[7] F. Schauer, *Thinking like a lawyer: a new introduction to legal reasoning*. Harvard University Press, 2009.

[8] J. Knowles, "Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[9] P. Craig and G. de Búrca, "Eu law: Text, cases, and materials.," *7th ed. New York: Oxford University Press,*, 2020.

[10] C. Barnard, "The substantive law of the eu: the four freedoms.," *Fourth edition. Oxford University Press*, 2019.

[11] B. De Witte, E. Muir, and M. Dawson, "Judicial activism at the european court of justice," *Edward Elgar Publishing*, 2013.

[12] H. Rasmussen, "On law and policy in the european court of justice: a comparative study in judicial policymaking," *M. Nijhoff*, 1986.

[13] F. Wei, H. Qin, S. Ye, and H. Zhao, "Empirical study of deep learning for text classification in legal document review," *2018 IEEE International Conference on Big Data (Big Data)*, pp. 3317–3320, 2018.

[14] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, and J. Xu, "Cail2018: A large-scale legal dataset for judgment prediction," 2018.

[15] M. Z. Alom, T. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. Nasrin, M. Hasan, B. Essen, A. Awwal, and V. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, p. 292, 03 2019.

[16] L. Derczynski, "Complementarity, F-score, and NLP evaluation," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, (Portorož, Slovenia), pp. 261–266, European Language Resources Association (ELRA), May 2016.

[17] K. Spackman, ". signal detection theory: Valuable tools for evaluating inductive learning," *Proc. Sixth Internat. Workshop on Machine Learning. Morgan Kaufman, San Mateo, CA, pp. 160–163*, 1989.

[18] E. Haddi, X. Liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," in *ITQM*, 2013.

[19] J. Camacho-Collados and M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis," 2017.

[20] C. M. Bishop, *Neural Networks for Pattern Recognition*. USA: Oxford University Press, Inc., 1995.

[21] M. S, J. D, and D. M, "Evaluation of impact of neural networks in text classification," *Journal of University of Shanghai for Science and Technology*, vol. 23, pp. 1279–1292, 07 2021.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Text classification with an rnn," 2015. Software available from tensorflow.org.

[24] M. Gupta, K. Rajnish, and V. Bhattacharjee, "Impact of parameter tuning for optimizing deep neural network models for predicting software faults," *Scientific Programming*, vol. 2021, pp. 1–17, 06 2021.

[25] U. Sadl and Y. Panagis, "What is a leading case in eu law? an empirical analysis," *European Law Review*, vol. 40, pp. 15–34, Feb. 2015.

[26] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP*, 2013.