Legal Language Modeling with Transformers

Lazar PERIC [a], Stefan MIJIC [a], Dominik STAMMBACH [a] and Elliott ASH [a]

[a] *ETH Zurich*

Abstract. We explore the use of deep learning algorithms to generate text in a professional, technical domain: the judiciary. Building on previous work that has focused on non-legal texts, we train auto-regressive transformer models to read and write judicial opinions. We show that survey respondents with legal expertise cannot distinguish genuine opinions from fake opinions generated by our models. However, a transformer-based classifier can distinguish machine- from human-generated legal text with high accuracy. These findings suggest how transformer models can support legal practice.

Keywords. legal text generation, language modeling, transformer, human validation, machine detection of generated text.

1. Introduction

The capacity to efficiently generate human-like text has rapidly increased with the development of neural language models implemented through deep transformer architectures [1]. Besides breezing past the previous state-of-the-art baselines (e.g. perplexity on the Penn Treebank and other corpora [2]), auto-regressive models like GPT-2 have amazed both researchers and the public with their ability to generate believable and expressive text – for example, the now proverbial news article about scientists discovering unicorns in South America [3]. The even better coherence, originality and fluency achieved by GPT-3 [2] promise additional gains as these models are further extended and refined.

In the legal domain, especially, there is enormous potential for well-intended use of such language models. The law is the domain of natural language, and natural language models could be used for many tasks, such as the generation of contracts, briefs, and rulings. Many legal writing tasks are somewhat repetitive, and in these tasks especially legal language models could save practitioners significant time and resources. Some early steps in this direction are taken by [4, 5], who use a character-level recurrent neural network to draft text in the style of international treaties.

While the promise of these language models is clear, concerns about potential misuse have led to extensive discussions, including disagreements about whether these models should be released to the public [3]. A complementary line of papers has explored automated detection of machine-generated text, mostly focusing on newspaper articles

Proceedings of the 2020 Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL)

[6, 7]. Correspondingly, other work has explored human responses to news articles generated by language models, finding that overall humans respond to machine-generated news text the same way they respond to human-generated news text [8].

We add to this literature by focusing on generating legal documents. Legal corpora have two important differences from the generic or news-oriented corpora that have been the focus of previous work. First, the documents tend to be much longer and to rely on long-range connections within documents. This document length poses a problem for the standard Transformer's memory usage, which grows quadratically with the input sequence size. One remedy to this problem is Transformer-XL [9], which allows (in theory) access to arbitrarily long memory sequences using a recurrence mechanism and relative positional encoding.

Second, legal language is technical. Similar to [8], we ask how well humans are able to distinguish machine- from human-generated documents. But the language used by judges is more technical, and less expressive, than that used by journalists. Understanding and writing effective legal documents requires many years of training, which could make the task difficult for language models. On the other hand, legal language is more formulaic and repetitive than generic language [10], so it might be easier to generate believably. Given that the benefits (and risks) of text generation in the law are at least as significant as those in journalism, more empirical work on this topic is needed.

This paper explores the use of language models for generating legal text. We use a corpus of 50'000 judicial opinions from U.S. Circuit Courts, appeals courts which review the decisions made by lower courts. We train a Transformer-XL model from scratch and fine-tune a pre-trained GPT-2 model on this corpus. We use both models to machine-generate text conditioned on the start of judicial opinions not considered during training.

We evaluate the model outputs qualitatively and based on a set of language metrics. We validate our method via a survey where human annotators tried to guess genuine from generated opinion text. Even respondents with legal training had difficulty distinguishing judge-generated from machine-generated texts. These results suggest that transformer language models can learn to generate consistent and coherent legal text.

Legal disputes are an adversarial process. In a world where language models are often used to support legal arguments, it would be useful to detect machine-generated text when used by an adversary. Following the approach for fake news detection in [6], we show that our machine-generated texts can be used to train a classifier that discriminates genuine from machine-generated texts with a very high accuracy, seemingly outperforming humans on that task.

To summarize, our findings suggest the legal language models could be a promising tool for legal practice. Such technology could be further enhanced and used to assist practitioners in writing legal documents. In addition, the models could help differentiate human- from machine-written legal text.

2. Data and Methods

2.1. Data

Our empirical setting is U.S. Circuit Courts, the intermediate appellate courts in the federal court system. Circuit Court judges review the decisions of the District Courts, deciding whether to affirm or reverse. The judges explain their decision by providing a written opinion. Our corpus comprises 50’000 of these U.S. Circuit Court opinions, uniformly sampled from the universe of opinions for the years 1890 through 2010.[1] The sample includes both lead (majority) opinions and addendum opinions (concurrences and dissents).

We undertake minimal pre-processing, so that our generator can replicate the original style of the texts. We do remove some metadata and XML markup but keep capitalization, punctuation, etc. In particular, we preserve the special legal citation notation used by U.S. courts.

The opinions are in general quite lengthy, containing an average of 2024 tokens (words) per article. Figure 1 shows the average opinion length over time as well as the number of documents per year. We see that average length gradually decreased from the 1890s reaching a minimum in the 1970s. After that, the average length of these opinions has grown steadily until the present day. Notably, it is around 1970 when digital legal research databases came into use.

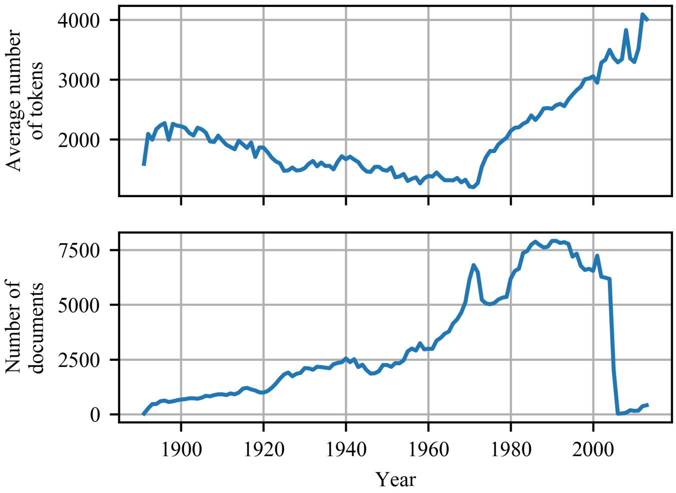

Figure 1. Average number of tokens in a document (top) and number of documents (bottom) per year.

[1] The opinion texts can be obtained from CourtListener.

2.2. Language Model Objective

Our approach to representing legal documents is an auto-regressive language model. We are given an unsupervised corpus U. The objective is to learn a set of parameters Θ to maximize

$$L_1(U) = \sum_i logP(u_i|u_{i-k},...,u_{i-1};\Theta), \tag{1}$$

the probability of token u_i appearing at position i conditional on the k tokens appearing before u_i.

We experimented with two transformer architectures for auto-regressive language modeling. First, we trained a Transformer-XL model from scratch on our corpus. Second, we fine-tuned an existing GPT-2 checkpoint on our legal corpus. The implementations for each of these approaches are described in the following two subsections.

2.3. From-Scratch Transformer-XL Model

We train a Transformer-XL model from scratch. After applying a word tokenizer to the training corpus, we built a vocabulary of 528'476 different token types. Sentence boundaries (detected using spaCy [11]) are annotated as *<eos>*. If seed documents contain words that did not appear in the training corpus, those are replaced with a special *<unk>* token. Figure 2 shows an excerpt of the beginning of a new opinion after preprocessing.

> [* 1209] JERRE S. WILLIAMS , Circuit Judge : <eos> This appeal is lodged by H. Roger Lawler , a debtor in bankruptcy . He claims that an award in bankruptcy of $ <unk> in attorneys ' fees to Richard W. Horton and Vernon O. <unk> is too high . Horton and <unk> are cross - appealing the amount of the award

Figure 2. Sample of the dataset.

The Transformer-XL model consists of 12 layers, where each layer consists of 16 attention heads. The model dimension is set to 512, the inner dimension in each fully-connected feedforward layer (following the attention layer) is set to 2048. These values were derived from the configuration of the Transformer-XL base model trained on the WikiText-103 dataset, which was used for text generation in [9]. This yields 312M trainable model parameters.

2.4. Fine-Tuned GPT-2 Model

Secondly, we consider a fine-tuned GPT-2 model. We started with a GPT-2_{BASE} checkpoint, downloaded from huggingface. The model consists of 124M parameters and was pre-trained on a general-domain corpus. We further fine-tuned the model on the language modeling task on the Circuit Court opinions using a maximum sequence length of 256.

We used the aitextgen library with default settings to perform further training and text generation.

2.5. Model Training

We trained the Transformer-XL model with a batch-size of 16 and used a sequence length of 128 during training. After having trained the network for 24'000 steps, i.e., the model has read 50M unique tokens, we achieve a perplexity of 36.85 on held-out data (perplexity indicates how well our language model predicts the held-out data). We trained the GPT-2 model on a batch-size of 2 for 36'000 steps, i.e., we provided 18.5M tokens during training. The fine-tuned GPT-2 model achieved a final perplexity of 28.40 on held-out data.

Figure 3 shows perplexity over time during training for the Transformer-XL architecture trained from scratch. It slowly converges. In future work, we would like to explore fitting models with more parameters on a larger number of opinions.

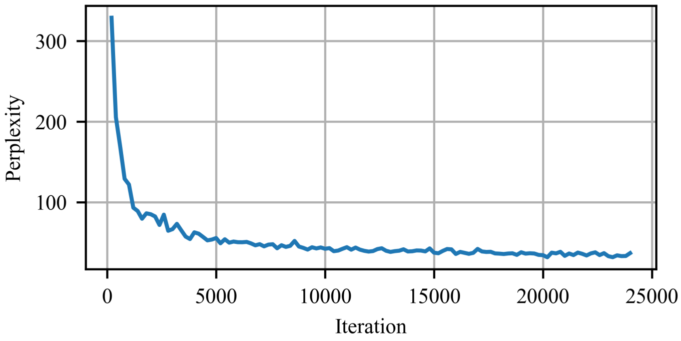

Figure 3. Perplexity during training.

2.6. Generation and Decoding

The next step is to generate text using our language models conditioned on a seed text snippet. For Transformer-XL, we used the memory recurrence mechanism of the architecture. The sequence length was set to 64 with a memory of 640 tokens. We apply the decoding strategy from the official codebase[2]. At each step during generation, we sample from the k most probable tokens in a uniform manner. The default value for k is set to 3.

For GPT-2, we predict the next token given the 255 tokens before that token. We use top-p decoding strategy with $p = 0.9$, implemented in the aitextgen library[3]. That is, we sample from the most probable tokens from the distribution over the next token such that the most probable tokens combine in total at least 90% of the probability mass.

For the experiments, we used our models to automatically generate text snippets conditioned on the actual start of 1.8K opinions, randomly sampled from the text.

[2] `https://github.com/kimiyoung/transformer-xl`
[3] `https://github.com/minimaxir/aitextgen`

2.7. Open Source Demonstration

We produced a Google Colab notebook for generating legal texts using the fine-tuned GPT-2 model, available at `https://colab.research.google.com/drive/1VjTUAUgXqZCsmDZYS2NO98_2w46oIEED?usp=sharing`. We have made the code for the Transformer-XL model publicly available at `https://github.com/lazar505/transformer-xl`.

3. Evaluation

3.1. A Generated Example

We show an example of generated text in Figure 4. At the top (a), we see the prompt: the start of a Circuit Court opinion describing appellant's request for an exception to a 90-day complaint filing period. In (b) we see the genuine completion. The third panel gives the generated completion for the from-scratch Transformer-XL model (c), the fourth for the vanilla GPT-2 model (d), and the fifth panel for the fine-tuned GPT-2 model (e).

(a) Prompt: Appellant urges that he acted diligently to present his claim to the court and that this diligence should justify the one-day delay. Without reciting all of the details of his efforts we note that the 90-day period for filing a complaint expired on May 14, 1973.
(b) Genuine Ending: On March 6, 1973, he had sought the services of an attorney to draft a formal complaint for him. The complaint was prepared and delivered to him prior to the filing deadline but the exact date is not clear from the record. Appellant's Memorandum is Support of Motion to Appeal would indicate this date as April 13, 1972.
(c) Transformer-XL: In this connection, the Nygren's argument is without support. Appellant argues, for example, for a period not exceeding five days after filing an indictment on October 15. Appellee argues, for example for an offense involving his activities under §506, which was committed on October 22 and April 22.
(d) Vanilla GPT-2: Appellant was able to present his claim in the court in less than 10 days, which is not a trivial amount of time to provide for a review of a case in which the complaint was filed within the 90-day period . Moreover, it is undisputed that appellant did not have any prior knowledge of the nature of the allegations and the nature of the issues raised in the complaint.
(e) Fine-tuned GPT-2 on Legal Text: The Government's failure to disclose the facts of the case, and the failure of the court to address the underlying factual issues is not at issue here. On November 19, 1973, respondent filed his civil suit in the District Court. The District Court denied the claim for $ 100 million, arguing that the District Court had not acted to protect his interest in the case, and that there was no constitutional or statutory impediment to enforcing the suit.

Figure 4. Generated text example.

The genuine continuation (b) further explains the exact timeline of the events. Transformer-XL (c) discusses the argument of the appellant, with some discordant dates mentioned. Vanilla GPT-2 (d) discusses that the appellant was able to present the claim in less than 10 days, somewhat contradictory of the initial issue over taking more than 90 days. Finally, the fine-tuned GPT-2 (e) digresses into a discussion of the government's failure to disclose the facts of the case. While there is suspicious content in each of the machine-generated snippets, the style is legalistic and facially convincing.

3.2. Readability

To provide some statistical comparison of the text generator models, we calculated the Flesch–Kincaid readability score[4] of the generated samples and the actual cases for the 1.8K sample. As a non-legal comparison, we compute readability for the Harry Potter corpus. A higher score indicates easier readability.

Table 1. Readability scores.

Opinion	Readability Score
Actual Circuit Opinions	37.37
Opinions Transformer-XL	40.92
Opinions GPT-2	37.47
Harry Potter	72.8

We show results in Table 1, which indicate some interesting variation across the real and generated corpora. We see that the documents generated by GPT-2 are as hard to read as the original Court Opinions, while the ones generated by Transformer-XL have a higher readability score, indicating that they are slightly easier to read. This is in line with the overall lower perplexity obtained by our fine-tuned GPT-2 checkpoint, indicating that the GPT-2 checkpoint is closer to the legalese style of the opinions. All our generated text and the genuine opinions are rather technical and thus hard to read. Meanwhile, the average readability score for the Harry Potter books is almost twice as high than the Circuit Opinions and our generated opinions.

3.3. Human Survey Evaluation

We designed a survey to measure the ability of native English speakers and Legal Professionals to distinguish generated legal extracts from authentic Court opinions. We showed the participants an authentic prompt from a legal text, the actual opinion, and the generated continuation produced by our Transformer-XL and GPT-2 models. The participants were then asked to rank the text according to how authentic they believed the different snippets were. At the same time, they had to indicate how confident they were about their verdict. The ten prompts can be found in Appendix A[5].

We recruited 54 survey participants in total. For the following statistics, we only include the responses of 25 participants who correctly identified a trivial attention-test question with at least 70% confidence. More details regarding that example can be found in Appendix A.

The human annotators could not distinguish the human from machine generated legal texts. Comparing genuine texts to GPT-2, the annotators were successful only 49% of the time (slightly worse than random). With Transformer-XL, the participants were slightly more accurate, picking the genuine text 53% of the time. In addition, the annotators indicated a generally low level of confidence in their guesses. The average in-

[4] `https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests`

[5] Appendices are available at `https://www.dropbox.com/s/pra2jb7md0208xv/Legal_Language_Modeling_with_Transformers.pdf?raw=1`

dicated confidence was 57.6% for extracts produced by GPT-2 and 55.4% for extracts produced by Transformer-XL. The differences between GPT-2 and Transformer-XL are not statistically significant

If we focus on a subset of participants with legal expertise, (at least three years of legal education, $n = 6$), the results are almost identical. They could not correctly guess the genuine follow-up text, and they were not confident in their guesses. These numbers (although a small sample size) suggest that legal experts and other participants are similarly uncertain about determining whether a legal opinion is machine-generated or written by humans.

4. Automated Detection of Machine-Generated Opinions

We have a balanced dataset of 3.6K examples (1.8K judge-generated vs. 1.8K machine-generated texts) for both the Transformer-XL and the GPT-2 model. We fine-tune RoBERTa (a pre-trained bidirectional transformer [12]) to classify which continuations are judge-generated (the genuine opinion) and which are machine-generated. In both settings, we train on half of the examples and evaluate on the other half.

4.1. Training Transformer Discriminators

We fine-tune RoBERTa with a learning rate of 2e-5 and a batch-size of 16 for three epochs. We show results in Table 2. In the last columns of Table 2, we show precision, recall and F1 for the machine-generated class.

Table 2. Deep-learning detection of machine-generated texts

Setting	Accuracy (%)	Pr (%)	Rc (%)	F1 (%)
GPT-2 generated texts (RoBERTa$_{\text{GPT-2}}$)	94.2	90.1	99.2	94.4
Transformer-XL generated texts (RoBERTa$_{\text{T-XL}}$)	97.5	97.7	97.2	97.5
RoBERTa$_{\text{GPT-2}}$ predicting Transformer-XL texts	76.7	85.6	64.3	73.4
RoBERTa$_{\text{T-XL}}$ predicting GPT-2 texts	74.9	95.8	52.2	67.6

In the first two rows, we see that in contrast to humans (among them legal experts) considered in our survey, machine-learning models are very good at detecting machine-generated language. This is in line with observations made in [7], where BERT-classifiers outperform humans on the task of automatically detecting machine-generated text. Remarkably, in the setting where we fine-tune GPT, we detect almost all machine-generated examples (recall over 99%) while maintaining a precision of over 90%. In the Transformer-XL case, we have fairly balanced precision and recall, both over 97%.

In the last two rows, we show results for 2 additional experiments, which would be closer to a real-world setting where one would not know what model was used to generate fake legal text. We use our discriminator trained on detecting GPT-2 produced samples to detect Transformer-XL samples, and vice versa. In both settings, the overall accuracy of the classifier drops significantly. But still, performance is much better than human guessing. Interestingly, in both experiments, the discriminator achieves a high precision at the expense of a reduced recall.

It seems that overall, our models are capable of detecting machine-generated texts with very high accuracy. The models are effective even though we provided a modest-sized training set and have not performed any hyperparameter tuning. Appendix B provides some descriptive analysis of the errors made by the discriminators.

5. Conclusion

In this paper, we explored the capabilities of Transformer-based Decoders on generating U.S. Circuit Court opinions. We show examples of machine-generated text and validate the quality of our generated text via a survey. The results suggest that humans (among them legal experts) are in general uncertain about determining whether text is machine-generated or judge-generated. We additionally explore the capabilities of machine-learning models to predict whether a given snippet is machine-generated or an actual opinion. We find that machine-learning models are very good at that task, yielding way above 90% accuracy in different settings.

While the hurdle to generate authentic long texts is still high, the level of proficiency that our two Transformer models achieve for short texts is already high enough to make a direct identification difficult, even for specialized legal experts.

This research could be extended in a number of ways. More work could be done to make these text generators useful for lawyers or judges seeking assistance in drafting legal documents. For example, one could do conditioned generation based on the direction of the decision – affirm or reverse. Thus language models could be used to explore the legal arguments for or against a given position.

It should be mentioned that legal language models, like language models generally, are vulnerable to biases prevalent in the training data. For example, our judicial-opinion models could be biased in terms of the ideology and topics used by Circuit Court judges [13]. These biases will re-appear in text generated by our models, even if that is not intended. Measuring and adjusting for these biases is an important area for future work.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language Models are Few-Shot Learners,” *CoRR*, vol. abs/2005.14165, 2020.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners,” *OpenAI Blog*, 2019.

[4] W. Alschner and D. Skougarevskiy, “Can Robots Write Treaties? Using Recurrent Neural Networks to Draft International Investment Agreements,” in *JURIX*, ser. Frontiers in Artificial Intelligence and Applications, vol. 294, 2016, pp. 119–124.

[5] W. Alschner and D. Skougarevskiy, “Towards an automated production of legal texts using recurrent neural networks,” in *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law, ICAIL*, 2017, pp. 229–232.

[6] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, “Defending Against Neural Fake News,” in *Advances in Neural Information Processing Systems 32*, 2019, pp. 9054–9065.

[7] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, “Automatic Detection of Generated Text is Easiest when Humans are Fooled,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1808–1822.

[8] S. E. Kreps, M. McCain, and M. Brundage, “All the News that’s Fit to Fabricate: AI-Generated Text as a Tool of Media Misinformation,” *Available at SSRN 3525002*, 2020.

[9] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context,” *CoRR*, vol. abs/1901.02860, 2019.

[10] D. Simonson, D. Broderick, and J. Herr, “The Extent of Repetition in Contract Language,” in *Proceedings of the Natural Legal Language Processing Workshop*, 2019, pp. 21–30.

[11] M. Honnibal and M. Johnson, “An Improved Non-monotonic Transition System for Dependency Parsing,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1373–1378.

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[13] C. I. Hausladen, M. H. Schubert, and E. Ash, “Text classification of ideological direction in judicial opinions,” *International Review of Law and Economics*, vol. 62, p. 105903, 2020.