

Contrast Is All You Need

Burak Kilic^{1,2,*}, Floris Bex^{1,3,†} and Albert Gatt^{1,†}

¹Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

²Shareforce B.V., Rotterdam, The Netherlands

³Tilburg Institute for Law, Technology, and Society, Tilburg University, Tilburg, The Netherlands

Abstract

In this study, we analyze data-scarce classification scenarios, where available labeled legal data is small and imbalanced, potentially hurting the quality of the results. We focused on two finetuning objectives; SetFit (Sentence Transformer Finetuning), a contrastive learning setup, and a vanilla finetuning setup on a legal provision classification task. Additionally, we compare the features that are extracted with LIME (Local Interpretable Model-agnostic Explanations) to see which particular features contributed to the model's classification decisions. The results show that a contrastive setup with SetFit performed better than vanilla finetuning while using a fraction of the training samples. LIME results show that the contrastive learning approach helps boost both positive and negative features which are legally informative and contribute to the classification results. Thus a model finetuned with a contrastive objective seems to base its decisions more confidently on legally informative features.

Keywords

LegalNLP, Contrastive Learning, NLP, Explainable AI

1. Introduction

The scarcity of publicly available, high quality legal data is causing a bottleneck in legal text classification research [1]. While there are a few publicly available datasets, such as CUAD [2], and LEDGAR [3], these datasets are unbalanced. They may provide good baselines to start with; however, the scarcity of samples for specific classes means that there is no guarantee of robust performance once models are adapted to downstream classification tasks.

Few-shot learning methods have proven to be an attractive solution for classification tasks with small datasets where data annotation is also time-consuming, inefficient and expensive. These methods are designed to work with a small number of labeled training samples and typically require adapting a pretrained language model to a specific downstream task.

In this paper¹, we focus on three major aims. First, we finetune the LegalBERT[5] model on the publicly available LEDGAR provision classification dataset. We compare the success of a contrastive learning objective and a

more standard objective to finetune the pretrained model.

Secondly, we finetune the same baseline model with these two finetuning objectives with the balanced dataset created from LEDGAR. Lastly, to analyze the trustworthiness and explain individual predictions, we extract the tokens from the model as features by using LIME [6] to compare which features had more positive or negative impacts.

2. Related Work

The legal text classification has been tackled with various BERT techniques to adopt domain-specific legal corpora [7] [8]. While these studies often report state-of-the-art results with BERT-based models, they do not address the issue of data scarcity for specific applications.

There have been several pieces of research on efficient finetuning setups that can potentially address this necessity, such as parameter efficient finetuning (PEFT), pattern exploiting training (PET), and SetFit (Sentence Transformer Finetuning) [9], an efficient and prompt-free framework for few-shot finetuning of Sentence Transformers (ST). SetFit works by first finetuning a pretrained ST on a small number of text pairs, in a contrastive Siamese manner. Also, SetFit requires no prompts or verbalizers, unlike PEFT and PET. This makes SetFit simpler and faster. We explain how SetFit works in more depth in the following section.

Proceedings of the Sixth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2023), June 23, 2023, Braga, Portugal.

*Corresponding author.


[†]These authors contributed equally.

✉ b.kilic@uu.nl (B. Kilic); f.j.bex@uu.nl (F. Bex); a.gatt@uu.nl (A. Gatt)

🌐 <https://www.shareforcelegal.com> (B. Kilic);

<https://www.uu.nl/staff/FJBex> (F. Bex); <https://albertgatt.github.io/> (A. Gatt)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

¹While our paper shares a similar title with "Attention is all you need" [4], we focus on a different topic.

2.1. SetFit: Sentence Transformer Finetuning

SetFit is a prompt free framework for few-shot finetuning of ST, addressing labeled data scarcity by introducing contrastive learning methods to generate positive and negative pairs from the existing dataset to increase the number of samples.

There are two main steps involved in SetFit, from training to inferencing. First, a contrastive objective is used to finetune the ST, and then the classification head is trained with the encoded input texts.

At the inference stage, the finetuned ST also encodes the unseen inputs and produces the embeddings accordingly. Then the classifier head gives the prediction results based on the newly generated embeddings.

ST finetuning To better handle the limited amount of labeled training data in few-shot scenarios, contrastive training approach is used. Formally, we assume a small set of K -labeled samples $D = (x_i, y_i)$, where x_i and y_i are sentences and their class labels, respectively. For each class label $c \in C$, R positive triplets are generated: $T_p^c = (x_i, x_j, 1)$, where x_i and x_j are pairs of randomly chosen sentences from the same class c , such that $y_i = y_j = c$. Similarly, a set of R negative triplets are also generated: $T_n^c = (x_i, x_j, 0)$, where x_i are sentences from class c and x_j are randomly chosen sentences from different classes such that $y_i = c$ and $y_j \neq c$. Finally, the contrastive finetuning data set T is produced by concatenating the positive and negative triplets across all classes where $|C|$ is the number of class labels, $|T| = 2R|C|$ is the number of pairs in T and R is a hyperparameter. SetFit will generate positive and negative samples randomly from the training set, unless they are explicitly given [9].

This contrastive finetuning approach enlarges the size of training data. Assuming that a small number (K) of labeled samples is given for a binary classification task, the potential size of the ST finetuning set T is derived from the number of unique sentence pairs that can be generated, namely $K(K - 1)/2$, which is significantly larger than just K .

Classification head training In this second step, the fine-tuned ST encodes the original labeled training data $\{x_i\}$, yielding a single sentence embedding per training sample: $Emb(x_i) = ST(x_i)$ where $ST()$ is the function representing the fine-tuned ST. The embeddings, along with their class labels, constitute the training set for the classification head $TCH = (Emb(x_i), y_i)$ where $|TCH| = |D|$. A logistic regression model is used as the text classification head throughout this work.

Inference At inference time, the fine-tuned ST encodes an unseen input sentence (x_i) and produces a sen-

tence embedding. Next, the classification head that was trained in the training step, produces the class prediction of the input sentence based on its sentence embedding. Formally this is predicted label $i = CH(ST(x_i))$, where CH represents the classification head prediction function.

3. Data

We present experimental results both on the original LEDGAR dataset, and on a balanced version. We describe the original dataset first, then we give a brief description of how the dataset was further balanced for the presented experiments.

3.1. Data source

As a main corpus, we used the publicly available LEDGAR² provision classification dataset, consisting of 60,000 training samples in total, with 100 different provision labels.

We did not apply any additional preprocessing or data modification techniques to keep the data as it is to make the experiments reproducible.

To create a dedicated test dataset for the unbalanced data scenario, we randomly selected 25 samples per label from the corpus, in total approximately 2,500 samples. The rest of the 57,500 samples are used to generate the train/dev sets.

The training sets are created by selecting 4, 8, 12, and 16 samples per label for SetFit, and 50, 100, 150, and 200 for the vanilla finetuning setup. Therefore the maximum number of samples is calculated as: maximum number of samples per label multiplied by the number of total labels, as can be seen in Figure 1. In practice, in the case of the vanilla finetuning setup, we end up with fewer training samples than this total. This is because some labels are extremely sparse, and there are fewer total samples than the stipulated maximum per label.

3.2. Crawling and balancing

The original LEDGAR dataset is imbalanced. The smallest label consists of only 23 samples, and the largest has 3167 samples in the original training dataset. Therefore, to create a new balanced dataset, we selected the most frequent 32 labels.

For labels with more than 1000 samples, we downsampled to 1000 samples per label. For labels with fewer than 1000 samples, we upsampled by crawling and retrieving additional data from LawInsider,³ removing any duplicates. As a result, a new dataset has been created that consists of 32 classes, with each having 1000 provisions.

²<https://autonlp.ai/datasets/ledgar>

³<https://www.lawinsider.com/>

Additionally, we also created a dedicated test dataset for the balanced data scenario, and selected 25 samples per label randomly for the 32 labels, for a total of 800 samples. The remaining 31,200 samples are used for training with a random 80/20 train/dev split.

For finetuning with the balanced dataset, we again train with varying sizes of training data, using 4, 8, 12, and 16 samples per label for SetFit, and 50, 100, 150, and 200 for the vanilla finetuning setup, as can be seen in Figure 2.

Note that, unlike the case of the unbalanced data, the total sizes for the vanilla finetuning setup in the balanced case correspond to the totals obtained by multiplying the maximum sample size with the number of labels.

4. Experiments

4.1. Models

It has been shown that models which have been pre-trained on domain-specific legal data outperform general-purpose models [5]. Therefore, throughout this paper, the baseline we use is a finetuned LegalBERT using legal-bert-base-uncased.⁴ We compare this standard, or "vanilla" finetuned baseline to a model finetuned with the contrastive objective used in SetFit.

4.2. Experimental Setup

The finetuning setup is the most crucial stage of the experimenting setup. Therefore, we kept the common hyperparameters of SetFit and vanilla setups the same. The rest of the parameters were kept as their default values, provided by the respective implementations. The important hyperparameter for SetFit finetuning is the R parameter, which defines the number of positive and negative pairs to be generated from the given training set. We kept this parameter as its default value, 20 across all the experiments. For both models, we used 1 epoch for the finetuning. Table 1 gives detailed common hyperparameters of finetuning setups for both SetFit Trainer⁵ and Vanilla Trainer.⁶

5. Results

5.1. F1-score comparisons: Original dataset

In Table 2, we compare the F1-scores for different experiments, with the test set described in Section 3. The

⁴<https://huggingface.co/nlpaueb/legal-bert-base-uncased>

⁵<https://github.com/huggingface/setfit>

⁶https://huggingface.co/docs/transformers/main_classes/trainer

Table 1

Common hyperparameters for SetFit and Vanilla Trainer

Hyperparameter	Value
Learning Rate	2e-5
Warmup Ratio	0.1
Seed	42
Batch Size	8
Epoch	1
Metric	accuracy

Table 2

F1-score comparison between SetFit and Vanilla finetuning, original LEDGAR dataset

Models	Samples	Micro-F1	Macro-F1	Weighted-F1
Vanilla	4933	0.5805	0.5151	0.5273
SetFit	400	0.6565	0.6348	0.6423
Vanilla	9756	0.6734	0.6180	0.6317
SetFit	800	0.6808	0.6709	0.6781
Vanilla	14379	0.7083	0.6632	0.6780
SetFit	1200	0.7104	0.6962	0.7054
Vanilla	18734	0.7190	0.6712	0.6864
SetFit	1600	0.7206	0.7097	0.7183

original LEDGAR dataset is used in this experiments, with an 80/20 train/dev split.

As can be seen from the table above, SetFit’s contrastive learning approach yielded a better F1-score compared to the vanilla finetuning, despite only using a fraction of the training samples.

Additionally, we observed that Weighted-F1 displays a larger gap between models compared to Micro-F1. This is particularly expected, since the problem of unbalanced data is exacerbated in the vanilla finetuning setup as the maximum number of samples per label increases.

5.2. Accuracy comparisons: Original and balanced dataset

In Figure 1, we compare the finetuning results of SetFit and vanilla models, finetuned on the original LEDGAR dataset with the same training split and test dataset as the previous experiment.

We observed that the models achieve comparable accuracies overall, despite the differences in Weighted F1-scores in the Table 2. However, it is still noteworthy that the contrastive learning approach achieves accuracy comparable to the vanilla finetuned model with very small sample sizes.

In Figure 2, we compare the accuracy of the two approaches, this time with the balanced LEDGAR dataset. In this experiment we also used 80/20 train/dev split. The results show that the contrastive learning finetuning has

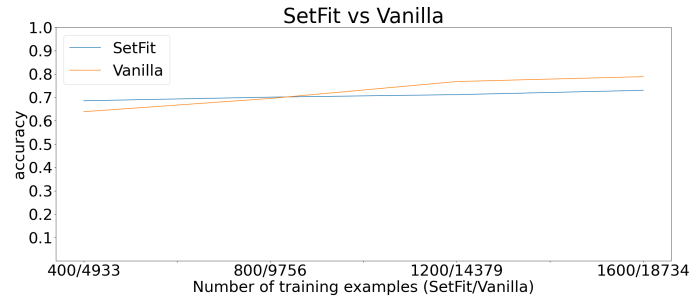


Figure 1: Accuracy comparison between SetFit and Vanilla finetuning, original LEDGAR dataset

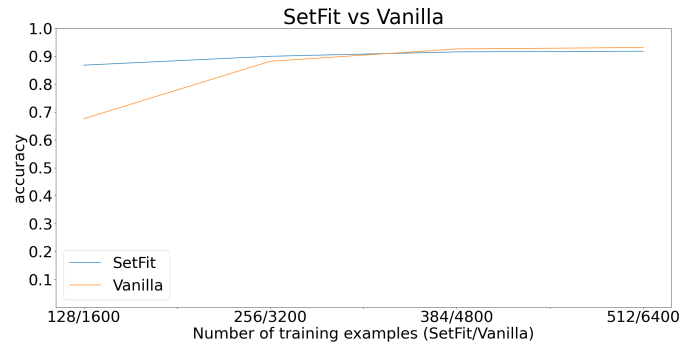


Figure 2: Accuracy comparison between SetFit and Vanilla finetuning, balanced LEDGAR dataset

a warmer start compared to vanilla finetuning, particularly in small data scenarios. However, as can be seen from the graph, SetFit is comparable with vanilla model across all the experiments as well.

5.3. LIME feature comparisons

In machine learning in general, but especially in domains such as law, trustworthiness of AI systems is crucial. The ability to explain model predictions is central to increasing trustworthiness in at least two respects. First, explanations have an impact on whether a user can trust the prediction of the model to act upon it; second, they also influence whether a user can trust the model to behave in a certain way when deployed.

Several approaches to explaining model predictions have been proposed in the literature, including LIME [6], SHAP [10], and GRAD-CAM [11].

Through the training results mentioned in previous

sections, we observed that SetFit models were comparable with vanilla models, despite using a fraction of the dataset. However, we get very little information about whether the models base their decisions on features which are intuitively correct, that is, if the models are classifying the provisions with legally informative features, or arbitrary ones.

LIME is a technique based on the creation of interpretable, surrogate models over the features that are locally faithful to the original classifier. This means that interpretable explanations need to use representations of those features that are understandable, trustworthy, and justifiable to humans [6].

For the text classification tasks, LIME features are restricted to the words that are presented in the provisions. Thus, the positively weighted words that lead toward a particular label are called "positive" features. Likewise, the negatively weighted words that reduce the model's estimate of the probability of the label are called "negative"

Table 3
F1-score comparisons of Adjustments and Authority provisions

Class Label	Vanilla	SetFit
Adjustments	0.7368	0.8571
Authority	0.5063	0.2903

features.

We kept the LIME hyperparameters the same in each model explanation for fair comparison and the details are as follows: The limit for the total number of words per classification is defined as K , and the complexity measure for the models is defined as:

$$\Omega(g) = \infty \mathbb{1}[\|w_g\|_0 > K]$$

where the g is defined as a simple interpretable sparse linear model (logistic regression in the case of SetFit, multinomial logistic regression in the vanilla model); w_g is defined as the weight vector of g . The $K = 10$ is selected across all the experiments for simplicity and potentially can be as big as the computation allows. The size of the neighborhood for local exploration is set to 25. The distance function D was kept as the default, cosine distance metric.⁷

Thus, in this section, we compare the positive and negative features of SetFit and vanilla models extracted using the LIME setup mentioned above.

To ensure a fair comparison, we used the SetFit model trained with 800 training samples and the vanilla model trained with 9756 training samples. As shown in Figure 1, the two models converged and obtained comparable performance with these settings. We selected two test labels to compare, namely Adjustments and Authority provisions. Again, for a fair comparison, we chose the labels based on the cases where one technique did better than the other, in terms of their respective F1-scores. For the Adjustments label, the SetFit model outperformed the vanilla model, and for the Authority label, vanilla finetuning outperformed the SetFit model. Thus, we aim to observe the differences in the model-predicted features for these labels. Table 3 shows the F1-score differences of these provisions.

We begin by comparing the positive features which both approaches have in common (i.e. the features they both assign a positive weight to), for the two target labels. These are shown in Figure 3 and Figure 8. The figures suggest that the contrastive approach from SetFit seems to help to boost legally informative features more than vanilla models, even in the small data scenarios. For instance, words like "adjustments", "shares", "dividend", "stock", etc. can give a first strong hint about the Adjustments provision classification results, as well as words

like "authority", "power", "act", "execute", "binding", etc. for the Authority provision. Thus, domain experts can make decisions based on their usefulness.

In Figures 4 to 7, we show the top positive features for the two models separately, for each label. We note that similar observations can be made with respect to these figures, that is, the contrastive learning framework boosts the positive weight of features that are intuitively more legally informative. Nevertheless, we also see that less informative features, including stop words, are also assigned some positive weight.

Additionally, we also observed similar behavior with the negative features in Figures 9 to 12. For negative features, the SetFit model trained with a contrastive objective assigns a greater negative magnitude. Thus, it appears that negative role of these features is accentuated in the contrastive setting, relative to the standard finetuning setup. For instance, words like "changes", "shall" and "without" for the Adjustments provision and "which", "common", "document" and "carry" for the Authority provision sound generic and may not give legally informative hints to humans. However, in the vanilla model case, similar legally non-informative negative features are also present but not enough to perturb the model's decisions.

6. Conclusions & Future Work

This paper presented a detailed comparison of legal provision classification. Motivated by the challenge of low-resource scenarios and data imbalance, we compared the performance of a LegalBERT model finetuned in a standard setting, to one finetuned using a contrastive objective.

Following previous work [5], we assumed that models pretrained on legal data are better able to retain the legal knowledge and terminologies in the process of finetuning. On the other hand, our experiments show that the type of finetuning approach matters, especially where data is relatively scarce. In particular, the contrastive learning approach showed promising results in terms of evaluation metrics, achieving performance comparable or better than the vanilla finetuning setup. The results also showed that the positive and negative features extracted from the models differ significantly, favoring the SetFit model, despite using almost 11 times less data.

As future work, investigating the limitations of SetFit deeper with more hyperparameters on legal data may be beneficial for pushing the model capabilities further. Also, we plan to use other explainability tools such as SHAP or GRAD-CAM to compare the extracted features. Finally, an evaluation of the appropriateness of the positive and negative features identified using explainability methods needs to be carried out with domain experts.

⁷<https://github.com/marcotcr/lime>

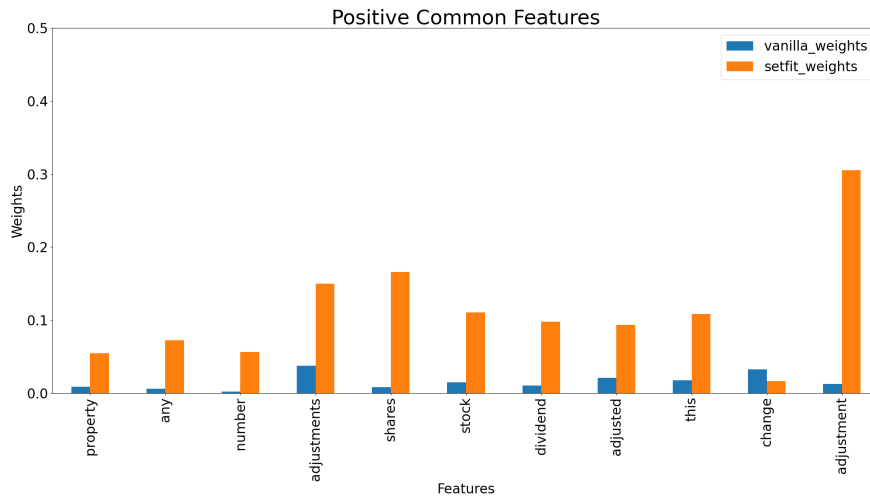


Figure 3: SetFit vs Vanilla finetuning, common positive LIME features comparison for Adjustments provision

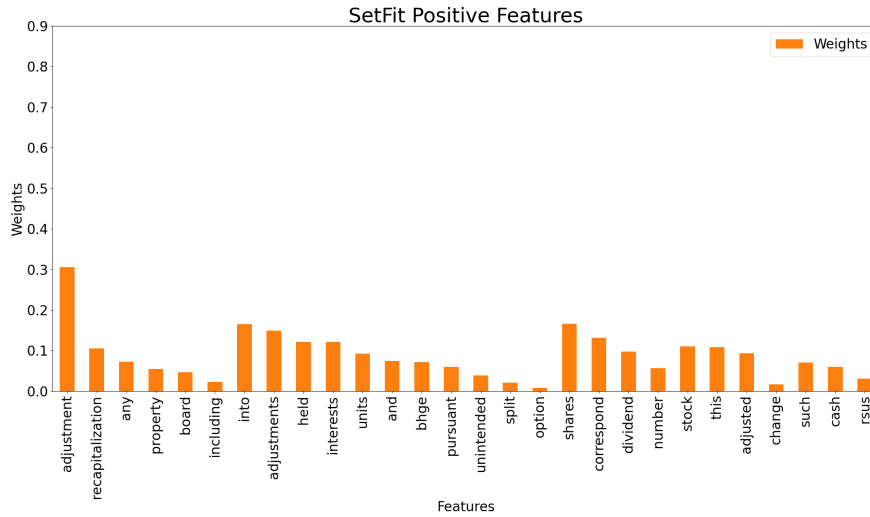


Figure 4: SetFit finetuning positive LIME features for Adjustments provision

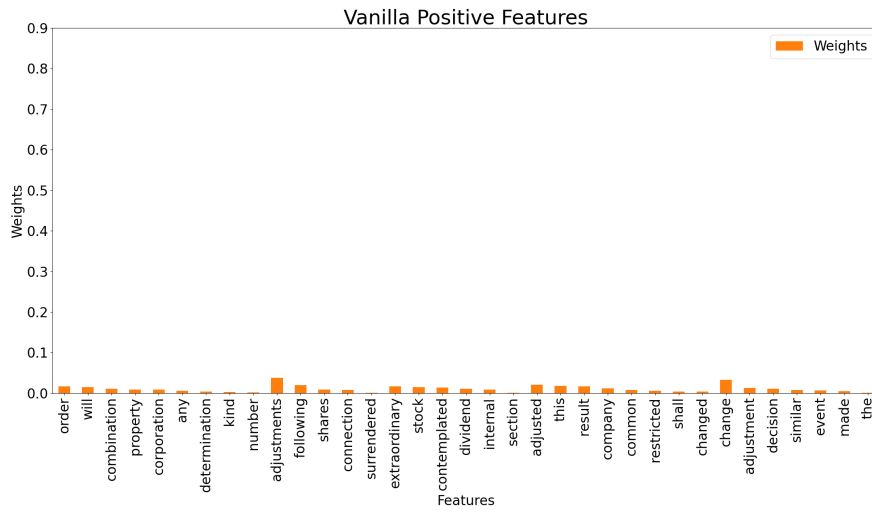


Figure 5: Vanilla finetuning positive LIME features for Adjustments provision

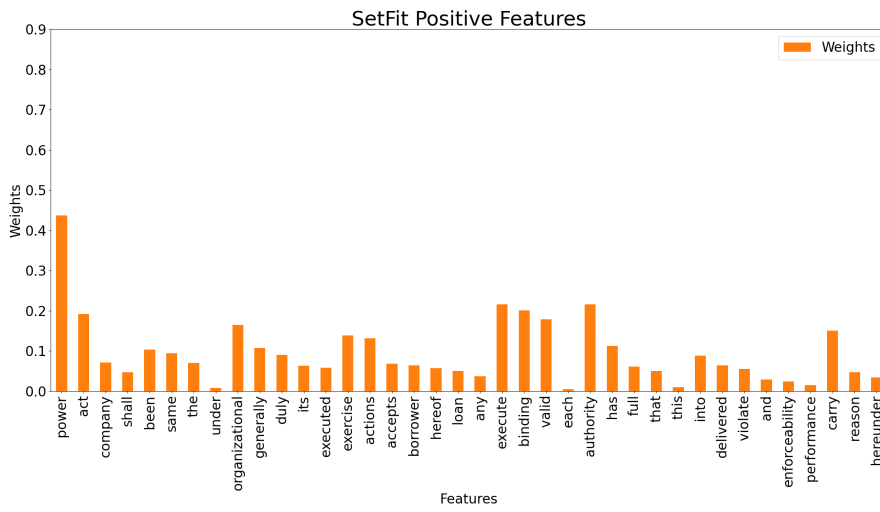


Figure 6: SetFit finetuning positive LIME features for Authority provision

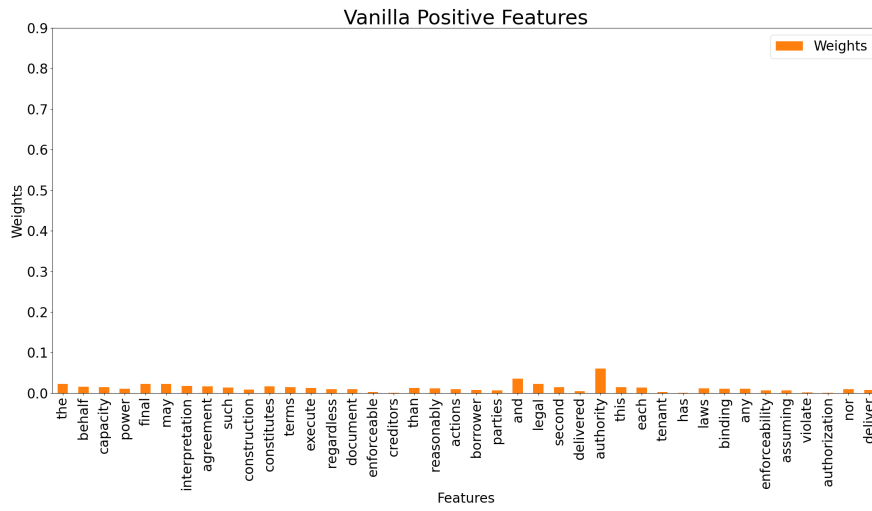


Figure 7: Vanilla finetuning positive LIME features for Authority provision

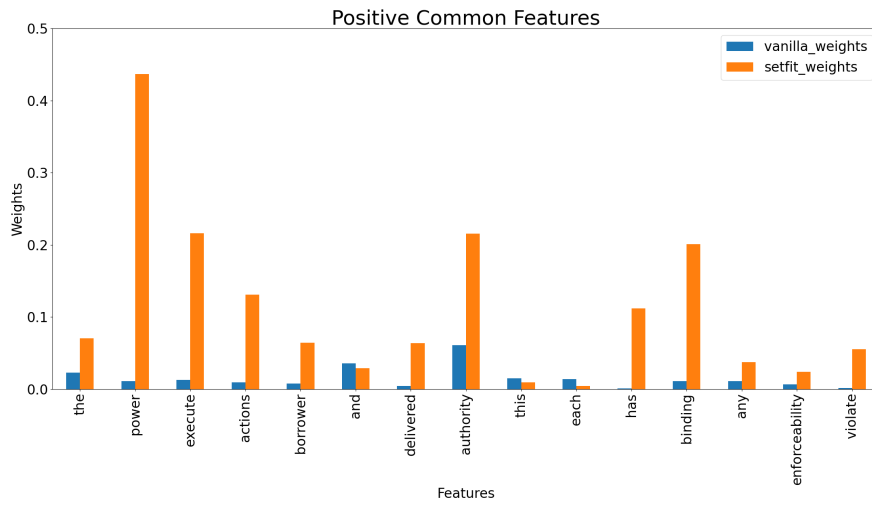


Figure 8: SetFit vs Vanilla finetuning, common positive LIME features comparison for Authority provision

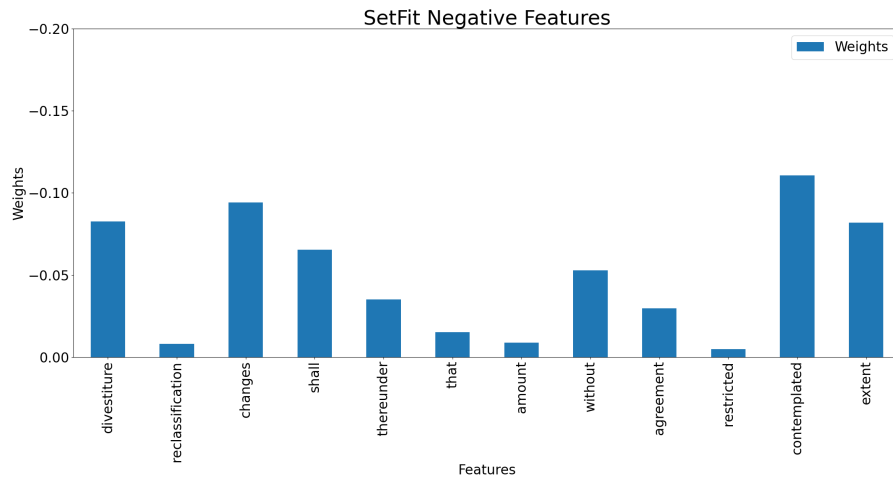


Figure 9: SetFit finetuning negative LIME features for Adjustments provision

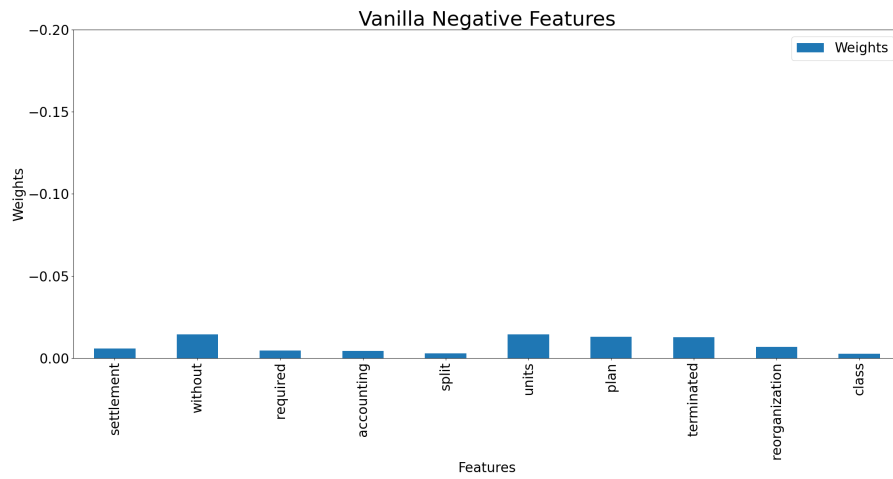


Figure 10: Vanilla finetuning, negative LIME features for Adjustments provision

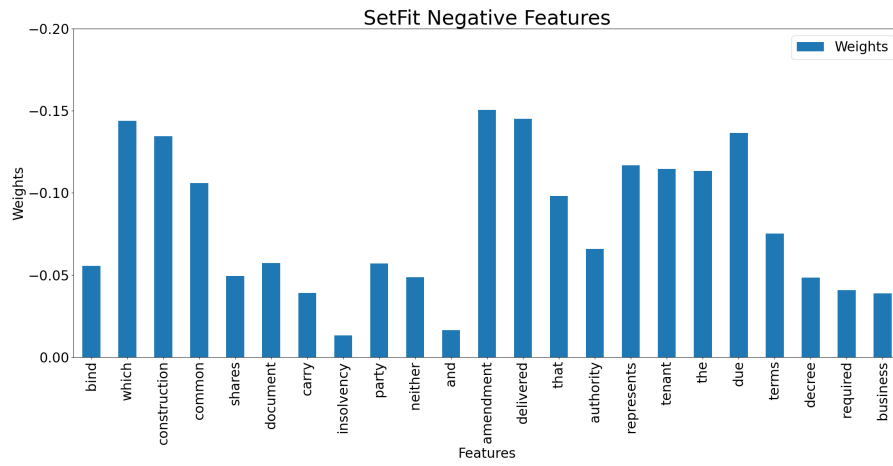


Figure 11: SetFit finetuning negative LIME features for Authority provision

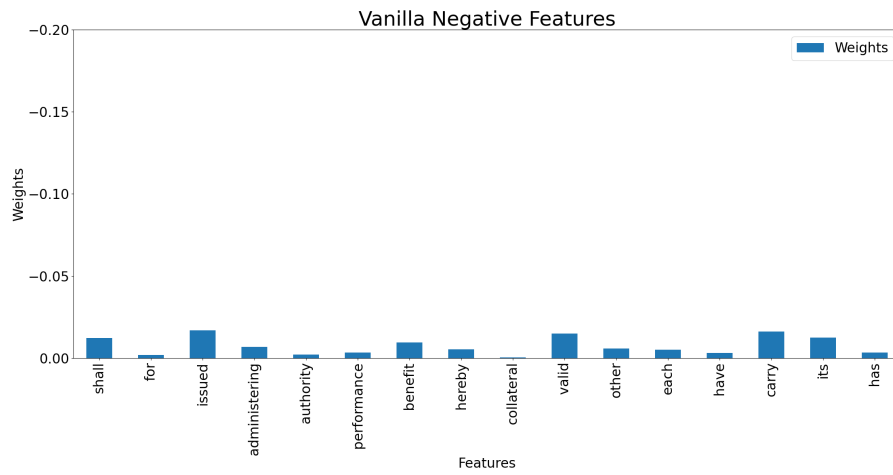


Figure 12: Vanilla finetuning, negative LIME features for Authority provision

References

- [1] E. Elwany, D. Moore, G. Oberoi, BERT goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding, CoRR abs/1911.00473 (2019). URL: <http://arxiv.org/abs/1911.00473>. arXiv:1911.00473.
- [2] D. Hendrycks, C. Burns, A. Chen, S. Ball, CUAD: an expert-annotated NLP dataset for legal contract review, CoRR abs/2103.06268 (2021). URL: <https://arxiv.org/abs/2103.06268>. arXiv:2103.06268.
- [3] D. Tuggener, P. von Däniken, T. Peetz, M. Cieliebak, Ledger: A large-scale multi-label corpus for text classification of legal provisions in contracts, in: International Conference on Language Resources and Evaluation, 2020.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, CoRR abs/1706.03762 (2017). URL: <http://arxiv.org/abs/1706.03762>. arXiv:1706.03762.
- [5] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, LEGAL-BERT: The muppets straight out of law school, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 2898–2904. URL: <https://aclanthology.org/2020.findings-emnlp.261>. doi:10.18653/v1/2020.findings-emnlp.261.
- [6] M. T. Ribeiro, S. Singh, C. Guestrin, "why should I trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13–17, 2016, 2016, pp. 1135–1144.
- [7] N. Limsopatham, Effectively leveraging BERT for legal document classification, in: Proceedings of the Natural Legal Language Processing Workshop 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 210–216. URL: <https://aclanthology.org/2021.nllp-1.22>. doi:10.18653/v1/2021.nllp-1.22.
- [8] I. Chalkidis, I. Androutsopoulos, A. Michos, Extracting contract elements, in: Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law, ICAIL '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 19–28. URL: <https://doi.org/10.1145/3086512.3086515>. doi:10.1145/3086512.3086515.
- [9] L. Tunstall, N. Reimers, U. E. S. Jo, L. Bates, D. Korat, M. Wasserblat, O. Pereg, Efficient few-shot learning without prompts, 2022. arXiv:2209.11055.
- [10] S. M. Lundberg, S. Lee, A unified approach to interpreting model predictions, CoRR abs/1705.07874 (2017). URL: <http://arxiv.org/abs/1705.07874>. arXiv:1705.07874.
- [11] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, D. Batra, Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization, CoRR abs/1610.02391 (2016). URL: <http://arxiv.org/abs/1610.02391>. arXiv:1610.02391.