As easy as PIE: understanding when pruning causes language models to disagree

Anonymous ACL submission

Abstract

Language Model (LM) pruning compresses the model by removing weights, nodes, or other parts of its architecture. Typically, pruning focuses on the resulting efficiency gains at the cost of effectiveness. However, when looking at how individual data points are affected by pruning, it turns out that a particular subset of data points always bears most of the brunt (in terms of reduced accuracy) when pruning, but this effect goes unnoticed when reporting the mean accuracy of all data points. These data points are called PIEs and have been studied in image processing, but not in NLP. In a study of various NLP datasets, pruning methods, and levels of compression, we find that PIEs impact inference quality considerably, regardless of class frequency, and that BERT is more prone to this than BiLSTM. We also find that PIEs contain a high amount of data points that have the largest influence on how well the model generalises to unseen data. This means that when pruning, with seemingly moderate loss to accuracy across all data points, we in fact hurt tremendously those data points that matter the most. We trace what makes PIEs both hard and impactful to inference to their overall longer and more semantically complex text. These findings are novel and contribute to understanding how LMs are affected by pruning.

1 Introduction

Deep neural networks (NNs) are becoming increasingly larger, with remarkable improvements to their inference capabilities, but also very high computational demands. The latter has motivated research in the area of NN pruning, whose goal is to reduce a model (in terms of its parameters, nodes, layers, or any other aspect of its architecture) to a smaller version, without significant loss of inference quality. Pruning has been shown to produce smaller, hence more efficient NNs, with small loss to their effectiveness (Li et al., 2020a; Hooker et al., 2019). Similar findings are also reported when pruning Language Models (LMs) (Gupta and Agrawal, 2022; Wang et al., 2020; Sun et al., 2023; Sanh et al., 2020a; Michel et al., 2019) in NLP.

Class: neutral	
Premise: A group of seven individuals wearing rafting gear, white water raft down a river. **Hypothesis:** Seven men and women are in a yellow boat.	
Unpruned model prediction: entailment	
Pruned model prediction: neutral	
Class: entailment	
Premise: A woman is painting a mural of a woman's face. **Hypothesis:** There is a woman painting.	
Unpruned model prediction: entailment	
Pruned model prediction: contradiction	

Table 1: Examples where pruned and unpruned models disagree (from the SNLI dataset).

When pruning NNs, typically the focus is on the high efficiency gains achieved at the cost of effectiveness, commonly measured in terms of test set accuracy. However, when zooming in on precisely how individual data points are affected by pruning, it turns out that models of similar accuracy scores can have notably different weights and therefore make wildly different inferences on a subset of data points. In other words, the similar accuracy scores between pruned and unpruned models do not mean that pruning affects all data points in a uniform way, but rather that some parts of the data distribution are much more sensitive to pruning than others. This effect can go unnoticed when one measures pruning effectiveness in terms of mean accuracy, because taking the mean can hide such important score variations in the data. However, this does not change the fact that certain types of data are disproportionately impacted by pruning, which begs the question: what are the characteristics of these data points and how important is their detection?

In response to this, *Pruned Identified Exemplars* (PIEs) are defined as the subset of data points where pruned and unpruned models disagree (Hooker et al., 2019) (see example in Table 1). Studies in image processing reveal that PIEs are harder to classify, not only for NNs, but also for humans, because they a) tend to be mislabeled (ground truth noise), b) may have overall lower quality (inherently noisy signal), or c) may depict multiple objects (more challenging task) (Hooker et al., 2019). Hence, this subset of data points where pruned and unpruned models tend to disagree are also some of the most difficult data points for the model to handle. PIEs are those critical data points on which we would suffer the most damage, if the model were to be deployed out in the wild. Despite this, to our knowledge, PIEs have not been studied in NLP.

Motivated by this gap in understanding how LMs are actually affected by pruning, we study whether PIEs exist in text, what are their textual characteristics, and what this practically means for inference. Using eight pruning methods on two different LM architectures (BiLSTM and BERT) and four common NLP datasets for sentiment classification, document categorisation and natural language inference, we contribute the first study of PIEs in LM pruning for NLP. Our empirical analysis shows that there is always a subset of data points where pruned and unpruned models disagree, and that this subset is larger for BERT than BiLSTM. We also find that these data points, namely PIEs, are overall semantically more complex, contain on average more difficult words, and have generally longer text than the rest of the data. Furthermore, we find that PIEs contain a high amount of *influential examples*, i.e. data points that have the largest influence on how well the model generalises to unseen data. These findings are novel, and practically, they mean that, when pruning LMs for efficiency, and in particular BERT, with seemingly small drops to overall accuracy, we are in fact impacting notably the accuracy of a particular subset of our data, which also happens to be the most critical part of our data with respect to how well the model is expected to generalise to unseen data, or more simply put, how well the model actually learns. This effect is much more pronounced for BERT than for BiLSTM.

2 Pruned Identified Exemplars (PIEs)

We formally define PIEs and propose an extension of this definition to multi-label classification.

2.1 Formal definition of PIEs

Pruned Identified Exemplars (PIEs) are data instances[1] where the predictions of pruned and unpruned models differ (Hooker et al., 2019). Assume a single-label classification task, where each instance x belongs to a single class. Let $P = \{p_1, ..., p_N\}$ be the set of N different initializations of the pruned model, and $U = \{u_1, ..., u_N\}$ the set of N different initializations of the unpruned model.[2] Let $m(P, x)$ be the majority class assigned to x over all the initializations of the pruned model after training. This is computed as the most frequently predicted class for the instance x across all N initializations in P, i.e., the mode of the N predicted classes.[3] Similarly, $m(U, x)$ is the most frequent class predicted by the unpruned model initializations. Then, x is a PIE if $m(P, x) \neq m(U, x)$, i.e., the majority class assigned to x by the pruned and unpruned model is different.

2.2 PIEs in multi-label classification

We extend the above definition of PIEs to multi-label classification, where an instance x can belong to more than one class. We treat multi-label classification as multiple single-label classifications: an instance x is a PIE, if there exists a class such that the pruned and unpruned models disagree. Let $\tilde{m}(P, x)$ be the set of majority classes assigned to x over all the initializations of the pruned models. A class is assigned to the set of majority classes if $> N/2$ initializations of the pruned model predict that x belongs to that class. Similarly, $\tilde{m}(U, x)$ is the set of majority classes assigned by the unpruned model. Then, x is a PIE if $\tilde{m}(P, x) \neq \tilde{m}(U, x)$, i.e., the sets of majority classes predicted for x by the pruned and unpruned models differ. The inequality between $\tilde{m}(P, x) \not\subseteq \tilde{m}(U, x)$ and $\tilde{m}(P, x) \not\supseteq \tilde{m}(U, x)$ means that x is a PIE even if the pruned and unpruned model disagree only on a single class.

Holste et al. (2023) propose the following alternative way of selecting PIEs in a multi-label setting. For each instance, they compute the average prediction over all initializations. Then, the

[1] We will use the terms *instance* and *data point* interchangeably henceforth.

[2] N must be the same for pruned and unpruned models.

[3] In case of ties, classes are sorted ascendingly by their associated number, and the first class is assigned.

Dataset	# train	# test	# val	# classes	Classification
IMDB	20000	25000	5000	2	single-label
SNLI	549367	9824	9842	3	single-label
Reuters	6737	1429	1440	23	multi-label
AAPD	53840	1000	1000	54	multi-label

Table 2: Dataset statistics after preprocessing.

Scoring → Scheduling ↓	Impact Based	Magnitude	Random
Iterative + Weight Rewinding	IIBP-WR	IMP-WR	-
Iterative + Fine tuning	IIBP-FT	IMP-FT	IRP-FT
At Initialization	IBP-AI	MP-AI	RP-AI

Table 3: Our 8 pruning methods. *Random* cannot be combined with *Weight Rewinding* because weights that are rewinded to their initial values are not random.

instances are ranked by the average prediction, and agreement is measured as the Spearman rank correlation between the rankings for the pruned and unpruned models. The 5^{th} percentile of instances with highest disagreement (lowest Spearman rank correlation) are considered PIEs. This approach does not allow to exactly quantify the amount of PIEs for the pruned and unpruned models. In addition, in Holste et al. (2023), an instance can be considered as non PIE even if there is disagreement between the pruned and unpruned models, simply because that instance is outside the 5^{th} percentile. Our definition of PIEs is stricter than Holste et al.'s (2023), since disagreement even on a single class determines the instance to be a PIE.

3 Study design

Our aim is to study whether PIEs exist in text data, what are their textual characteristics, and what this practically means for inference. We present the datasets, LMs, and pruning methods of our study.

Datasets. We use two single-label datasets: IMDB (Maas et al., 2011) for sentiment analysis, and SNLI (Bowman et al., 2015) for natural language inference. We also use two multi-label datasets for document categorisation: Reuters-21578[4], and AAPD (Yang et al., 2018). Statistics are in Table 2 (see Appendix A.2 for preprocessing details).

Language Model Architectures. We select two common types of LMs to represent both transformers and Recurrent Neural Networks (RNNs): BERT (Devlin et al., 2019), and bidirectional LSTM (BiLSTM) (Hochreiter and Schmidhuber, 1997). We train BiLSTM from scratch, but we finetune a pretrained version of $\text{BERT}_{\text{BASE}}$. See Table 5 in Appendix A.1 for details on the LMs, and Appendix A.1 for our tuning methodology.

Pruning methods. We use eight common pruning methods, shown in Table 3. Each of them is a combination of *scheduling* and *scoring*.

Scheduling controls the moment and frequency of the pruning iterations during training. We use two scheduling variations: (i) pruning the model before training (*at initialization*), and (ii) pruning in multiple iterations during training (*iterative*). Only for iterative pruning, we use two tuning strategies: *finetuning* and *weight rewinding*. In finetuning, we retrain the model after pruning and update its weights. In weight rewinding, we rewind weights to their initial state (Frankle and Carbin, 2019).

Scoring refers to selecting which weights to prune. A score is given to each LM weight, and the weights with the lowest score according to a threshold are pruned. We use 3 scoring variations: 1. The score is the absolute value of a weight (*magnitude based pruning* (Frankle and Carbin, 2019)); 2. The score is the weight multiplied by its accumulated gradient on 100 randomly sampled data points of the training set (*impact based pruning* (Lee et al., 2019)); 3. The score is randomly assigned a value between 0 and 1 (*random* (Jin et al., 2022)).

Overall, we prune each LM at 20%, 50%, 70%, 90%, 99% (see Table 5 in Appendix A.1 for details). For each configuration, we train 30 initializations. This results in 9840 runs (= 2 LMs x 4 datasets x 8 pruning methods x 5 pruning thresholds x 30 initializations + 2 LMs x 4 datasets x 30 unpruned model initializations), that require ca. 28000 AMD MI250X GPU hours. Our tuning methodology for pruning is detailed in Appendix A.4.

4 Experimental findings

We show how pruning impacts inference, the role of PIEs, and the textual characteristics of PIEs.

4.1 Pruning and occurrence of PIEs

Figure 1 shows the accuracy/F1 of pruned versus unpruned models (see Table 6 in Appendix A.1 for details on the number of parameters pruned). We see that pruning BERT/BiLSTM up to 50% gives overall tolerable drops to accuracy/F1 for most pruning methods. IIBP-FT is the pruning method with the overall smallest drop in accuracy/F1 com-

[4] https://www.daviddlewis.com/resources/.

pared to the unpruned model, and even outperforms the unpruned BiLSTM at times. We also see that, while unpruned BERT outperforms unpruned BiLSTM, pruning BERT hurts accuracy/F1 more than pruning BiLSTM, especially for pruning at 70%-99%. Hence **BERT is more sensitive to pruning than BiLSTM**, indicating that parameters in BERT are not as easily disposable as in BiLSTM. Otherwise put, BERT seems to make better use of its parameters than BiLSTM, because their removal has a bigger impact on it than on BiLSTM.

Table 4 shows the % of all data points[5] that are PIEs per model, dataset, pruning method and pruning threshold. We see that, as the pruning threshold increases, so does the proportion of PIEs, with very few marginal exceptions. This means that the particular subset of data points where unpruned and pruned models disagree becomes larger, the more we prune. In Table 4 we shade the PIEs of the best and worst pruned model (according to their accuracy/F1 in Figure 1) as green and gray respectively. We see that the best pruned model (green) has almost always a smaller percentage of PIEs than the worst pruned model (gray), per pruning threshold. In other words, **as the amount of PIEs increases, overall accuracy/F1 lowers, meaning that PIEs clearly impact inference quality**.

For the multi-label datasets, it is important to know, not only the proportion of data points that are PIEs, but also their distribution across classes. So, Figure 2 plots the distribution of all data points versus PIEs, across classes, for IIBP-FT, which is the pruner with the best overall F1 in Figure 1. The plots of the other configurations are in Appendix B.1. We show PIEs resulting from the least (20%) and most (99%) pruning, which should capture the lowest and highest % of PIEs according to Table 4. Figure 2 shows that PIEs are found across all classes of the dataset, from the least frequent to the most frequent class, and roughly follow the distribution of all data points across classes. This observation, combined with the findings of Table 4, means that **the impact of PIEs on inference quality is considerable on all classes of the dataset, regardless of class frequency.**

To probe further into the extent of this impact, Figure 3 shows accuracy only on PIEs versus accuracy on all data points, for BERT and SNLI. The plots of the other configurations are in Appendix

[5]From now on, whenever we refer to all data points, we mean all data points in the test set, unless otherwise specified.

Single-label

Dataset	Model	Pruner	20%	50%	70%	90%	99%
IMDB	BERT	IIBP-WR	7	10	10	10	11
		IIBP-FT	4	10	13	13	11
		IBP-AI	8	10	10	10	50
		IMP-WR	4	9	11	12	50
		IMP-FT	3	10	14	13	50
		MP-AI	6	10	10	12	50
		IRP-FT	8	13	12	50	50
		RP-AI	10	10	10	50	50
	BiLSTM	IIBP-WR	2	3	4	7	10
		IIBP-FT	5	5	5	5	3
		IBP-AI	2	5	7	10	16
		IMP-WR	2	3	4	3	11
		IMP-FT	5	5	5	5	5
		MP-AI	2	5	6	9	16
		IRP-FT	5	5	5	3	29
		RP-AI	2	4	6	9	18
SNLI	BERT	IIBP-WR	7	13	16	27	35
		IIBP-FT	3	5	8	13	28
		IBP-AI	6	12	18	34	47
		IMP-WR	5	11	16	30	66
		IMP-FT	3	6	12	16	66
		MP-AI	5	12	27	35	66
		IRP-FT	4	11	17	66	66
		RP-AI	8	26	32	66	66
	BiLSTM	IIBP-WR	4	5	7	16	29
		IIBP-FT	6	5	5	6	23
		IBP-AI	4	7	11	23	39
		IMP-WR	5	5	5	14	62
		IMP-FT	6	6	5	8	32
		MP-AI	4	7	12	20	62
		IRP-FT	6	5	5	16	46
		RP-AI	4	8	13	23	62

Multi-label

Dataset	Model	Pruner	20%	50%	70%	90%	99%
Reuters	BERT	IIBP-WR	5	8	16	31	100
		IIBP-FT	5	6	7	10	32
		IBP-AI	6	18	35	84	100
		IMP-WR	4	6	14	100	100
		IMP-FT	5	6	8	23	100
		MP-AI	5	12	32	100	100
		IRP-FT	5	10	24	100	100
		RP-AI	7	34	41	100	100
	BiLSTM	IIBP-WR	4	5	7	14	37
		IIBP-FT	4	5	5	5	9
		IBP-AI	4	7	14	34	44
		IMP-WR	5	5	6	7	29
		IMP-FT	5	4	4	6	13
		MP-AI	5	6	9	19	33
		IRP-FT	5	5	6	7	31
		RP-AI	4	7	10	22	35
AAPD	BERT	IIBP-WR	31	40	48	59	81
		IIBP-FT	29	37	45	51	63
		IBP-AI	34	48	59	78	100
		IMP-WR	31	38	49	79	100
		IMP-FT	28	40	45	56	100
		MP-AI	34	47	57	94	100
		IRP-FT	33	63	62	100	100
		RP-AI	38	59	76	100	100
	BiLSTM	IIBP-WR	26	34	39	64	88
		IIBP-FT	41	40	37	28	59
		IBP-AI	22	33	53	82	100
		IMP-WR	26	32	38	56	83
		IMP-FT	39	41	37	34	69
		MP-AI	21	30	41	62	86
		IRP-FT	41	36	30	44	88
		RP-AI	24	35	49	67	87

Table 4: Percentage of datapoints that are PIEs per configuration. Green and gray mark the percentages of datapoints that are PIEs for the best (green) and worst (gray) pruner per dataset and pruning threshold.

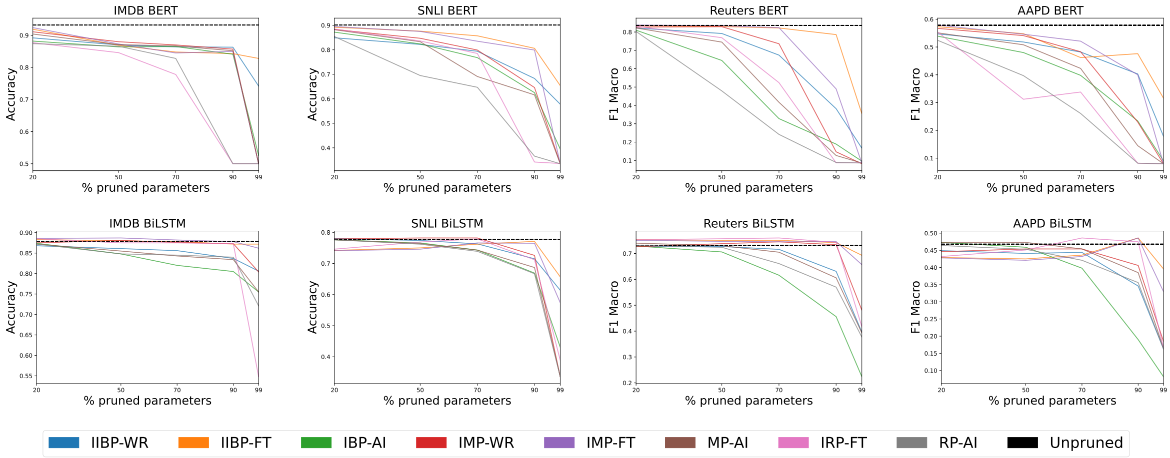

Figure 1: Accuracy/F1 (y axis) of unpruned and pruned LMs per pruning threshold (x axis), over 30 initializations.

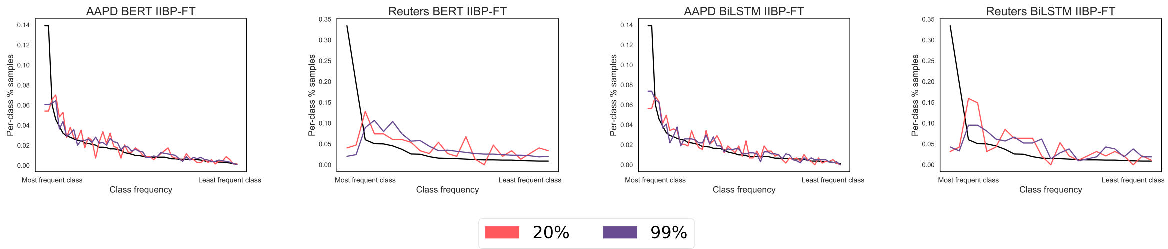

Figure 2: Distribution of all data points and of PIEs at 20% and 99% pruning, across classes sorted by frequency (x axis), for the multi-label datasets (test set) and IIBP-FT pruner.

B.1 and have overall similar trends. We see that accuracy is overall lower on PIEs (orange) than on all data points (blue), for both pruned and unpruned models, with few marginal exceptions for 99% pruning and BILSTM, where the scores are almost the same. The fact that accuracy is lower for PIEs than for all data points confirms the findings reported above. However, interestingly, Figure 3 also shows that the impact of pruning upon accuracy is much larger on the subset of PIEs than on all data points: the gap between the two orange lines (PIEs) in Figure 3 is notably larger than the gap between the two blue lines (all data points). Even when pruning 20%-50%, which according to Figure 1 has overall small drops to the mean accuracy of all data points for most pruning methods, still, the drop in accuracy to the data points of the dataset that are PIEs is much larger. This means that **PIEs always bear most of the brunt when pruning, but this effect goes unnoticed when reporting the mean accuracy over all data points**.

4.2 Influential examples in PIEs

The above findings suggest that PIEs are hard for inference. Next, we try to quantify this hardness, by studying how many of the PIEs are in fact *influential examples*, i.e. data points that have the largest influence on how well the model generalises to unseen data, irrespective of whether this influence is positive or negative. We do this using the EL2N score (Paul et al., 2021) as per Jin et al. (2022).

Given a model with weights w_t during training iteration t, and given an example (x, y) where x is the input and y is its label, $EL2N(x, y)$ is the L2 distance between the predicted probabilities $p(w_t, x)$ during t[6] and the one-hot label:

$$EL2N(x, y) = \mathbb{E}\left[||p(w_t, x) - y||_2\right] \qquad (1)$$

Examples are grouped into 20 bins based on their EL2N score percentiles. Higher EL2N scores mean that the model undergoes larger weight updates when the example is presented early in training. So, the bigger the weight changes, the higher the EL2N score, and the higher the influence of an example. Note that the above takes place during training, so we obtain PIEs on the training set.

[6] As the EL2N score is not reliable until at least one epoch of fine-tuning has been computed (Fayyaz et al., 2022), we only monitor the scores after the model has undergone training for at least one epoch (the first epoch that exceeds 30% of the total training epochs).

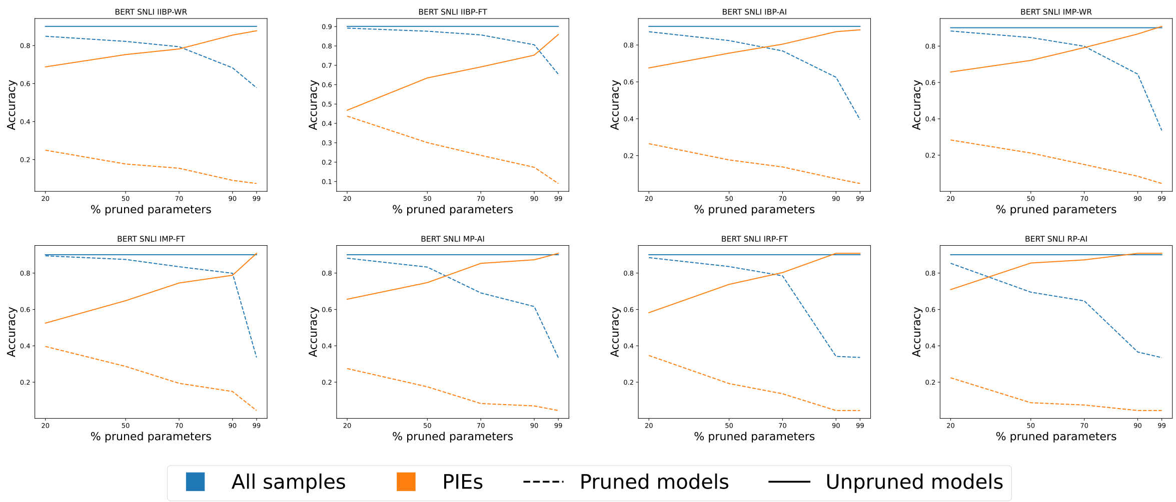

Figure 3: Accuracy (y axis) of unpruned (solid line) & pruned (dotted line) BERT on SNLI, for all data points (blue) or only for PIEs (orange), per pruning threshold (x axis), over 30 initializations. Each plot is a different pruner.

Figure 4 shows the distribution of PIEs across the degree of influence of all data points in the training set for IIBP-FT (the rest of the plots are in Appendix B.2). We see that PIEs are concentrated among the most influential data points (right hand side of the plots). This is even more so for BERT, where up to 80% - 100% of its most influential data points are in fact PIEs, compared to up to 70% for BiLSTM. This explains the finding of Section 4.1 that BERT is more affected by pruning than BiLSTM, because (a) more influential examples are PIEs in BERT than in BiLSTM, and (b) accuracy/F1 is lower among PIEs than among all data points, as we saw in Figure 3. We conclude that **a considerable amount of those data points that have the largest influence on how well the model generalises to unseen data are PIEs**.

4.3 Textual characteristics of PIEs

The above findings motivate the need to understand what the text of PIEs actually looks like. We do this using the following eight scores of text readability and length: (1) Automated readability index (Senter and Smith, 1967); (2) Coleman–Liau index (Coleman and Liau, 1975); (3) Flesch–Kincaid grade level (Kincaid Jr et al., 1975); (4) Linsear Write (O'hayre, 1966); (5) Gunning Fog index (Gunning, 1969); (6) Dale–Chall readability (Dale and Chall, 1948); (7) Number of difficult words; and (8) Text length, counted as the number of tokens per text. (1)-(6) are different approximators of text readability in terms of what formal education level would be needed in order to understand the text. (6) approximates comprehension difficulty based on a list of 3000 easily understandable words. (7) is a count of the number of words that are not in the Dale-Chall list of understandable words.

We compute the above scores first on all data points and then only on PIEs. Figure 5 shows the resulting plots for SNLI and BERT (the plots of the other configurations are in Appendix B.3). The black horizontal line represents all data points and PIEs having the same scores. Any divergence from this line reflects how much the scores of PIEs differ from those of all data points. E.g., the point 1.05 on the y axis of the Gunning Fog index plot means that the text of PIEs is approximately 1.05 times harder to understand than the text of all data points.

In Figure 5 we see that the formal education level needed for text understanding is overall higher for PIEs than for all data points (plots (a)-(e) and (g)). We also see that the text of PIEs has overall a larger amount of difficult words (plot (f)), and is on average longer than the text of all data points (plot (h)). Overall, according to the average scores of all pruning methods (turquoise line), PIE text is up to 1.03 times harder to understand than the text of all data points (plots (a)-(e) and (g)), with words that are up to 1.06 times more difficult (plot (f)), and text length that is up to 1.02 times longer (plot (h)). This means that **PIEs tend to be semantically more complex than the average text**. Note that the scores presented in plots (a)-(g) are designed to approximate human (as opposed to computational) difficulty in understanding text. This implies that **PIEs are more difficult than the average text,**

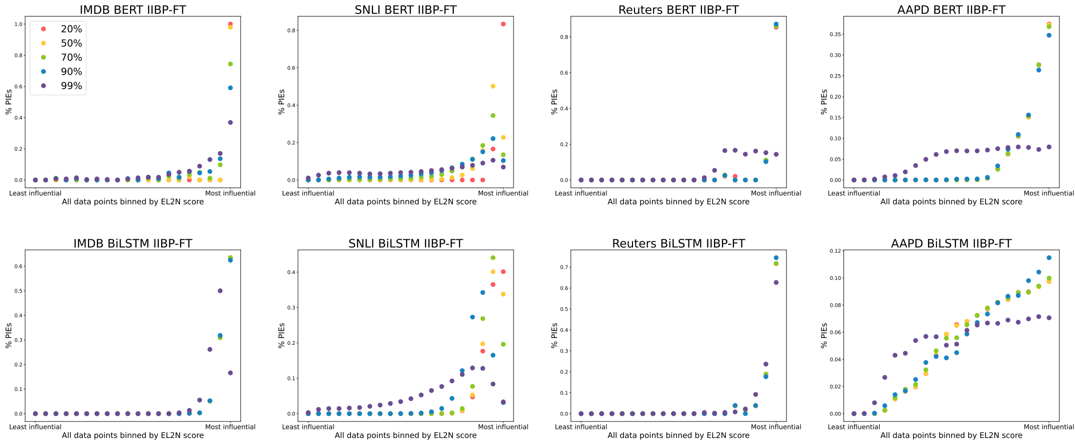

Figure 4: Percentage of data points that are PIEs (y axis) versus degree of influence (EL2N score) of all data points in the training set (x axis) for IIBP-FT across pruning thresholds (different colours).

not only for LMs (as shown in Figure 3), **but also for humans** (as shown in Figure 5).

5 Related work

Pruning LMs. LM pruning has typically been successful when models are first trained and then pruned (Li et al., 2020b). Most LM pruning methods work either globally or locally (Zhu et al., 2023; Sun et al., 2023; Frantar and Alistarh, 2023). In the global case, entire neurons, layers, or even large sections of the LM are pruned simultaneously. Examples include pruning entire attention heads in transformer models like BERT without severe inference degradation (Michel et al., 2019), pruning entire blocks of layers with substantial efficiency gains and minimal effectiveness loss (Lagunas et al., 2021; Ma et al., 2024), or identifying a smaller sub-network, a "winning ticket", within a large model that can achieve performance comparable to the original model when trained separately (Yu et al., 2020; Prasanna et al., 2020). Such global compression methods can lead to more interpretable and manageable models, but have the disadvantage that they tend to be architecture-specific. Unlike these global approaches, in local pruning, LM parameters/weights are pruned one layer at a time. This makes local pruning agnostic to particular model architectures (LeCun et al., 1989), making it possible to compare the effect of pruning on different types of LMs. As a result, local pruning has been successfully applied in NLP (Zhu et al., 2023; Sun et al., 2023; Frantar and Alistarh, 2023; Mishra and Chakraborty, 2021). In our study, we use only local pruning methods, allowing us to study PIEs in both transformers and RNNs.

For BERT in particular, it has been shown that a substantial amount of pruning can be applied during pre-training without significant loss in inference (Sanh et al., 2020b). It has also been shown that specific parameters that are redundant to such transformer architectures can be accurately identified by dedicated second-order pruning methods, such as Optimal BERT Surgeon (Frantar and Alistarh, 2022). However, another body of recent work also shows that complex LM pruning methods do not always work better than simpler, more straightforward pruning (Sun et al., 2024; Frantar and Alistarh, 2023).

Finally, researchers have also assessed, not only the accuracy, but also the loyalty (preservation of individual predictions) and robustness (resilience to adversarial attacks) of pruned BERT models (Xu et al., 2021). The findings reveal that traditional pruning methods that seem to maintain overall accuracy, may in fact affect the loyalty and robustness of the model. This line of work, similarly to ours, suggests that more nuanced analyses and evaluation approaches are needed to understand how pruning affects LMs beyond simple average accuracy.

Impact of pruning on subsets of data. While conventional pruned model evaluation has focused on inference time, number of pruned parameters, and effectiveness of the pruned models (Blalock et al., 2020; Gupta and Agrawal, 2022; Paganini and Forde, 2020; Renda et al., 2020), an understudied aspect has been the impact of model pruning on subsets of data. As language data is often power distributed, pruning can have a more severe effect on the performance of the least frequent, tail

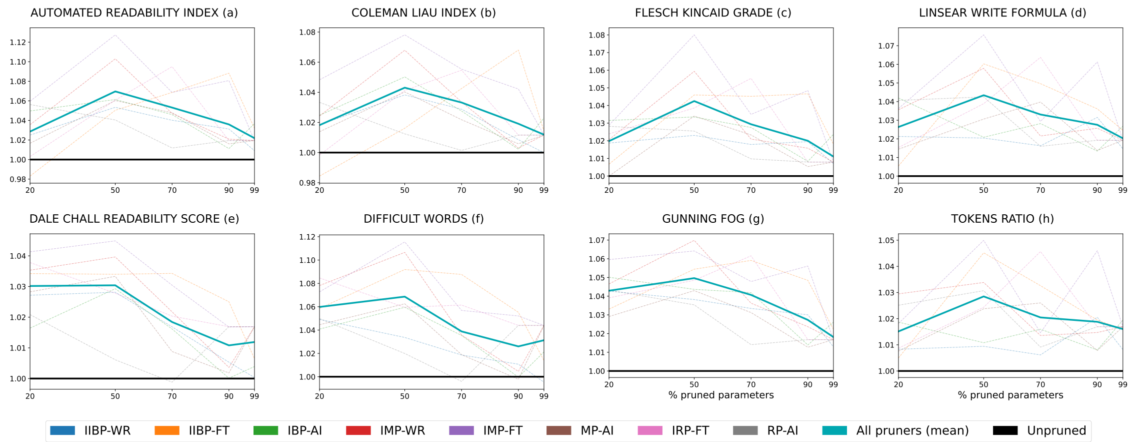

Figure 5: How the text of PIEs differs from the text of all data points, according to 7 readability scores (plots (a)-(g)) and text length (plot (h)). Ratio between the scores of PIEs and the scores of all data points (y axis), across pruning thresholds (x axis), for BERT and SNLI. The solid black horizontal line represents equal scores in PIEs and all data points. The solid turquoise line is the mean score of all pruners. Any line above the solid black line means that PIEs are harder to understand (plots (a)-(g)) or have longer text (plot (h)), on average, than all data points.

classes (Holste et al., 2023). This can make models less robust and more prone to overfit shortcuts (Du et al., 2023), result in disparate accuracy across subgroups of data (Tran et al., 2022; Hooker et al., 2020), and affect prediction quality based on sample frequency (Ogueji et al., 2022). Close to ours is the study of Hooker et al. (2019), who defined PIEs, and found them harder for both NNs and humans to classify. This study was limited to image processing. To our knowledge, our study is the first in-depth examination of PIEs for NLP, with novel findings about where and how often PIEs occur in text data, how they impact inference, and why.

6 Conclusions

We empirically studied how LMs are affected by pruning in the text domain. Unlike most work in this area which looks at overall gains in efficiency and costs to inference effectiveness, we zoomed in on precisely how pruning affects a particular subset of data points where pruned and unpruned models systematically disagree (*Pruning Identified Exemplars* (PIEs)). Using two LM architectures, four datasets, eight pruning methods, and five pruning thresholds, we found that PIEs impact inference quality considerably, but this effect goes undetected when reporting the mean accuracy across all data points. This effect is invariable to class frequency and increases the more we prune. BERT is overall more susceptible to this effect than BiLSTM. We also found that PIEs tend to contain a high amount of influential examples (data points that have the largest influence on how well the model generalises to unseen data). Probing into what it is about PIEs that makes them both hard and impactful to inference, we found that their text is overall longer and more semantically complex, and harder to process not only for LMs but also for humans, based on human text readability approximations.

Overall, our findings suggest that, the more influential and complex a data instance is, the higher the chance that pruned and unpruned models will disagree on its prediction, impacting disproportionately a subset of the dataset, yet going generally unnoticed when reporting mean accuracy on the whole test set of data points. This can pose significant risks to LMs, such as focusing on easier examples, and sacrificing inference quality on more difficult examples that are however linked to better generalisation. Given the increased call for compressing LMs, pruning them without considering the effect to PIEs can make models vulnerable in high-stakes applications, where relying solely on good top-line performance is inadequate to guarantee the model's reliability and trustworthiness across data instances and independently of class distribution.

Future work includes studying PIEs when pruning LLMs, and ways of balancing the impact of pruning fairly across PIEs and all data points.

Limitations

We evaluated the effects of pruning across eight pruning methods, two LM architectures, and four datasets. While these are representative, we cannot rule out the possibility that other pruning methods or model architectures might yield different results. Moreover, while we train BiLSTM from scratch, BERT utilizes an existing backbone model. This may affect some specific findings. Nonetheless, our findings across all tested experimental conditions, datasets, and models consistently point in the same direction and unanimously support our conclusions.

Future work could expand on our research by exploring larger architectures and alternative pruning methods. While we utilized extensive resources from the LUMI supercomputing infrastructure (over 28000 AMD MI250X GPU hours), it was not practically feasible to experiment with the latest large language models in our setting where we aimed varying many pruning thresholds, methods, and datasets. However, future studies could investigate individual architectures and pruning methods in isolation and benchmark their results against our findings.

We also did not explore the design of new pruning algorithms that take into account properties of the data, such as the link between the influence of the examples and pruned and unpruned models disagreement. These could help to mitigate both general effectiveness drops as well as improved handling of examples that are important for training and downstream usage of the models, which we leave for future work.

Ethics Statement

We adhere to the ACM Code of Ethics and Professional Conduct to ensure our work's integrity, fairness, and transparency.

Our study aims to enhance the understanding of natural language model pruning. Our results reveal the trade-offs between performance and the impact of pruning for examples that are potentially lower frequency and minority class, but may be highly important for downstream usage of the models. This can be particularly the case for high stakes domains, such as fact checking, medical informatics, and conversational and retrieval models that can impact decisions and opinions of individuals. By investigating the nuances of model pruning, we aim to inform modeling practices that consider both technical performance and potential weaknesses of compressed models. This can be critical in many specific application domains, but that is not always accounted for in standard performance analysis focusing on average effectiveness. To this end, our research identifies cases and settings where pruned models may underperform, providing valuable insights to avoid potential harm.

We have conducted our research fully transparently, documenting our methodologies and choices. While our study did not involve human subjects directly, it utilized publicly available datasets that include human annotations. We ensured that the use of these datasets complied with their respective terms of use.

We have respected all intellectual property rights in our research, and to our best knowledge properly citing all sources and datasets used. Our work builds on existing literature while providing new contributions to the field. We have also appropriately acknowledged the contributions of other researchers and sources that have informed our work.

We acknowledge that access to computing resources can be a barrier for some researchers aiming to reproduce our results. Our code to run the models was trained with a LUMI supercomputer[7], available for academic use to reproduce the results. We make our code and setup available to the scientific audience for further validation by the research community[8].

Acknowledgements

References

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pretrained bert networks. *Advances in neural information processing systems*, 33:15834–15846.

[7]https://www.lumi-supercomputer.eu/

[8]Code will be released upon acceptance

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan Awadallah. 2023. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1766–1778, Dubrovnik, Croatia. Association for Computational Linguistics.

Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Mohammad Taher Pilehvar, Yadollah Yaghoobzadeh, and Samira Ebrahimi Kahou. 2022. Bert on a data diet: Finding important examples by gradient-based pruning. *arXiv preprint arXiv:2211.05610*.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. 2021. Pruning neural networks at initialization: Why are we missing the mark? In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Elias Frantar and Dan Alistarh. 2022. The optimal bert surgeon: Scalable and accurate second-order pruning for large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10062–10079.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR.

Robert Gunning. 1969. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13.

Manish Gupta and Puneet Agrawal. 2022. Compression of deep learning models for text: A survey. *ACM Trans. Knowl. Discov. Data*, 16(4):61:1–61:55.

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

Gregory Holste, Ziyu Jiang, Ajay Jaiswal, Maria Hanna, Shlomo Minkowitz, Alan C Legasto, Joanna G Escalon, Sharon Steinberger, Mark Bittman, Thomas C Shen, et al. 2023. How does pruning impact long-tailed multi-label medical image classifiers? In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 663–673. Springer.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. 2019. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*.

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*.

Tian Jin, Michael Carbin, Dan Roy, Jonathan Frankle, and Gintare Karolina Dziugaite. 2022. Pruning's effect on generalization through the lens of training and regularization. *Advances in Neural Information Processing Systems*, 35:37947–37961.

JP Kincaid Jr, Rogers Robert P Fishburne, L Chissom Richard, and S Brad. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. *(No Title)*.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander Rush. 2021. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10619–10629, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. *Advances in neural information processing systems*, 2.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. 2019. Snip: single-shot network pruning based on connection sensitivity. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. 2020a. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5958–5968. PMLR.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E. Gonzalez. 2020b. Train large, then compress: rethinking model size for efficient training and inference of transformers. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2024. Llm-pruner: on the structural pruning of large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Abhishek Kumar Mishra and Mohna Chakraborty. 2021. Does local pruning offer task-specific models to learn effectively? In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 118–125.

Kelechi Ogueji, Orevaoghene Ahia, Gbemileke Onilude, Sebastian Gehrmann, Sara Hooker, and Julia Kreutzer. 2022. Intriguing properties of compression on multilingual models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9092–9110, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

John O'hayre. 1966. *Gobbledygook has gotta go*. US Department of the Interior, Bureau of Land Management.

Michela Paganini and Jessica Forde. 2020. On iterative neural network pruning, reinitialization, and the similarity of masks. *arXiv preprint arXiv:2001.05050*.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3208–3229, Online. Association for Computational Linguistics.

Alex Renda, Jonathan Frankle, and Michael Carbin. 2020. Comparing rewinding and fine-tuning in neural network pruning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020a. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, pages 20378–20389. Curran Associates, Inc.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020b. Compressing bert: Studying the effects of weight pruning on transfer learning. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 143–155.

RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, Technical report, DTIC document.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models.

Cuong Tran, Ferdinando Fioretto, Jung-Eun Kim, and Rakshit Naidu. 2022. Pruning has a disparate impact on model accuracy. *Advances in neural information processing systems*, 35:17652–17664.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Canwen Xu, Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2021. Beyond preserved accuracy: Evaluating loyalty and robustness of bert compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10590–10600.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. Sgm: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. 2020. Playing the lottery with rewards and multiple languages: lottery tickets in RL and NLP. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*.

A Implementation Details

A.1 Language Model Architectures

We use the pretrained uncased version of BERT-base from HuggingFace as is, which has 12 encoders with 12 self-attention heads (Wolf et al., 2020). BERT takes as input the tokenized text. We set the output layer size to match the number of classes of the data set the model is trained on. During training, we tune all of BERT's parameters

Our BiLSTM models receive as input a vector representation of the words in the text. To build such a vector we use Glove embeddings of size 300 (Pennington et al., 2014). We input the embeddings to a multilayer BiLSTM. We set the output layer size of the BiLSTM models to match the number of classes of the data set the model is trained on. On BiLSTM, we always use rectified linear units (ReLu) as activation functions.

We present the "percentage of pruned parameters" based on the total number of parameters that can be pruned in the model, instead of all of the parameters of the model (Chen et al., 2020). In Table 5 and Table 6 we report information about the number of remaining parameters in the architectures at different pruning amounts.

LM	Dataset	# parameters	20%	50%	70%	90%	99%
BERT	IMDB	109,483,778	15%	39%	55%	70%	77%
	SNLI	109,484,547					
	Reuters	109,499,927					
	AAPD	109,523,766					
BiLSTM	IMDB	647,810	20%	50%	69%	89%	98%
	SNLI	647,939					
	Reuters	650,519					
	AAPD	654,518					

Table 5: Number of LM parameters and % of parameters that are removed when pruning at 20%–99%. Numbers differ per dataset because the different size of the classification layer leads LMs to a different final amount of parameters.

Architecture	Unpruned	20	50	70	90	99
BERT	$1.1x10^8$	$9.2x10^7$	$6.7x10^7$	$5.0x10^7$	$3.2x10^7$	$2.5x10^7$
BiLSTM	$6.5x10^5$	$5.2x10^5$	$3.3x10^5$	$2.0x10^5$	$6.8x10^4$	$1.0x10^4$

Table 6: Number of parameters for the unpruned models, and remaining parameters when pruning at 20%-99%.

A.2 Datasets and Preprocessing

In table 7 we report dataset statistics after preprocessing. IMDB (Maas et al., 2011) is a single-label sentiment analysis dataset, made of reviews of movies. Each review is either positive or negative. IMDB has the longest sentences and the fewest classes across all our datasets on average. SNLI is a single-label natural language inference dataset. Each sample contains two sentences, and the task is to determine if the relationship between them is entailment, contradiction, or neutral. The dataset is available under a CC BY-SA license. SNLI has the most training samples and the shortest sentences among all our datasets on average. Reuters-21578 is a multi-label document categorization dataset, made of Reuters news belonging to 120 topics. Each news item is categorized and can belong to multiple topics. After preprocessing, the dataset has 23 classes. The dataset is available under CC BY license. Reuters has the fewest training samples among our datasets. AAPD is a multi-label document categorization dataset of article abstracts in computer science. Each arrticle can belong to multiple subjects, and the task is to identify the subjects given the abstract. The dataset is available under CC BY license. AAPD has the most classes across our datasets.

Dataset preprocessing. IMDB has 25000 training examples and 25000 test examples. To perform hyperparameter tuning of our models, we apply stratified sampling from the original training set to create a validation set of 5000 samples. On SNLI we use the original data set splits. On Reuters-21578 we remove all of the topics that do not appear in at least 100 documents and all of the documents that do not belong to at least one of the remaining topics. We perform stratified sampling and create three partitions by allocating 30% of the samples to the training set, 15% to the validation set, and 15% to the test set. For computational efficiency, before computing the statistics shown in Table 7, we convert texts in the Reuters dataset to lowercase and remove punctuation and numbers. Lastly, we use the original splits for the AAPD data set.

We further pad and truncate texts to submit train-

Dataset	# train	# test	# val	Mean/median	Min/max len	Std len	Tokens 85%	Max tokens	# classes	Task	Classification
IMDB	20000	25000	5000	268/201	8/2753	197	430	512	2	Sentiment analysis	single-label
SNLI	549367	9824	9842	23/22	5/124	7	30	128	3	Natural language inference	single-label
Reuters	6737	1429	1440	126/79	5/1305	137	232	256	23	Document categorization	multi-label
AAPD	53840	1000	1000	167/161	1/599	70	242	256	54	Document categorization	multi-label

Table 7: Datasets' statistics after preprocessing. # train, # test, and # val are respectively the number of instances in train, test, and validation sets. Mean/median, and Min/max are respectively the mean, median, minimum, and mximum number of tokens in the dataset's instances. Tokens 85% represent a value such that 85% of the datasets' texts have fewer or equal tokens than such value. Max tokens are the number of tokens, starting from the beginning of the text, after which we truncate texts. # classes is the number of classes. Task is the task solved using the dataset.

ing examples in batches, and we select a strategy to handle terms that are not present in the model's vocabulary (OOV). We explain these two steps next.

To fully take advantage of the available hardware, we submit training examples to the models in batches. When multiple texts with a different amount of tokens are present in a batch, our models require padding on the shorter texts in such a way that each input has the same amount of tokens. To have batches where each text is of equal size, we truncate long texts and pad short ones. Note that we do not remove documents based on a minimum amount of tokens in the text. To truncate the texts, we find a threshold after which we perform truncation. We define this threshold as the first power of two after which, by selecting the value as a threshold, at least 85% of the texts in the dataset do not need to be truncated. The resulting thresholds are reported as "Max tokens" in Table 7. An exception is made for SNLI. The SNLI dataset is made of short texts, and even the longest text is under 128 tokens. Hence we consider 128 tokens, representing the whole text for each sample in the data set. We then proceed to pad short texts in each batch to always exactly match the number of tokens specified in Table 7. For BERT we use the huggingface's tokenizer padding and pad all of the texts in each dataset to the respective "Max tokens" value in Table 7. BERT will mask and ignore the padding. For the BiLSTM model, we represent padding as a randomly generated embedding according to the mean and std distribution in Glove.

On BERT, OOV terms are assigned the default UNK token. On BiLSTM, we represent OOV terms with a vector defined as the average over all of the present word embeddings. The result of our preprocessing will be texts with exactly "Max tokens" tokens in which OOV terms are represented by the UNK token on BERT and as the average embedding vector on BiLSTM.

A.3 Pruning Methods

Model parameters are pruned one layer at a time. We prune uniformly across layers, i.e. we remove the same percentage of parameters in each layer. Following Chen et al. (2020) and Yu et al. (2020); Prasanna et al. (2020), we do not prune embedding layers and biases of the LMs (Gupta and Agrawal, 2022). We also do not prune the final classification layer, because its weights are likely disproportionately important to reach high effectiveness (Frankle et al., 2021).

With iterative pruning, we select a pruning percentage and keep it fixed for each pruning iteration to reach our pruning goal in exactly three iterations across all datasets, LMs, and pruning percentages. We train the model (BERT or BiLSTM) fully for N epochs, prune according to the selected percentage, and then retrain for N epochs. This process repeats until we achieve our pruning target as per (Jin et al., 2022). In total, this procedure requires four times the training iterations when compared to pruning at initialization.

A.4 Hyperparameter Tuning

We tune the unpruned model's hyperparameters for each combination of architecture and dataset. The resulting hyperparameters are then used to train both unpruned and pruned models. We do not tune hyperparameters of the pruning algorithms. The only tunable aspect when pruning at initialization is the percentage of parameters to prune. However, in our experiments, we fix five different values for this hyperparameter and we test such values on all pruning algorithms, hence, we do not optimize the percentage of pruned parameters. When pruning iteratively (with or without weight rewinding)

Dataset	Architecture	Batch size	Epochs		Best epoch	lr		Best lr
			Min	Max		Min	Max	
IMDB	BERT	32	2	6	3	2e-5	2e-4	0.00007
	BiLSTM	1024	10	30	26	2e-4	2e-3	0.00196
SNLI	BERT	256	2	6	2	2e-5	2e-4	0.00014
	BiLSTM	4096	30	50	39	2e-4	2e-3	0.00180
Reuters	BERT	128	5	15	14	2e-5	2e-4	0.00016
	BiLSTM	512	30	100	72	2e-4	2e-3	0.00152
AAPD	BERT	256	5	15	13	2e-5	2e-4	0.00015
	BiLSTM	2048	30	60	50	2e-4	2e-3	0.00184

Table 8: Search space and best configuration for the hyperparameter tuning of the models. Min and Max epochs represent the range of epochs used to perform hyperparameter tuning. Best epoch is the best epoch found with hyperparameter tuning. Min and Max lr are the range learning rate is tuned on. Best lr is the best learning rate found during hyperparameter optimization. The batch size is set to maximize the GPU usage.

we also need to select the number of pruning iterations and the amount of parameters to prune at each pruning iteration. To allow for comparison between pruning algorithms, we select a fixed percentage of parameters to remove during each iteration, such that in exactly 3 iterations the desired amount of parameters will be pruned. Hence those hyperparameters are inferred and fixed in each setting, leaving no hyperparameters to be optimized when pruning iteratively.

The hyperparameter tuning is performed separately on architectures and separately for each data set. We tune the hyperparameters using the random optimization from the weights and biases (WandB) platform with a budget of 100 objective function evaluations (Biewald, 2020). Hyperparameter tuning is set to maximize accuracy and macro F1 in the validation set for the single-label and multi-label tasks respectively. The search spaces optimal hyperparameter values are summarized in Table 8.

B Results

In Table 3 we report accuracy and F1 score with their standard deviation, obtained by unpruned models and pruned models at different amounts of pruned parameters.

In Table 10 we report accuracy and F1 score on PIEs obtained by unpruned models and pruned models at different amounts of pruned parameters. We highlight in blue the cases where the pruned models are on average more effective than the unpruned models on PIEs.

B.1 Pruning and occurrence of PIEs

We report here the additional results of Section 4.1.

In Figure 6 we show the distribution of all data points and of PIEs at 20% to 99% pruning, across classes sorted by frequency for the multi-label datasets. We observe the same overall trend in all settings. Regardless of the language model architecture, the percentage of PIEs in the most frequent class for Reuters is much lower than the percentage of examples belonging to the same class in all data points. This means that the disagreement between pruned and unpruned models is not focused on the most frequent class of Reuters. The disagreement is skewed instead towards the less frequent classes. On AAPD we observe a similar behaviour, however, the percentage of PIEs belonging to the most frequent class is higher, hence the disagreement is slightly more balanced across all classes.

In Figures 7, 8, 9, 10, 11, 12, and 13 we report the accuracy of unpruned and pruned models on PIEs and all samples in the dataset per pruning method, across pruning thresholds. The accuracy on PIEs is lower than the accuracy on all data points for both pruned and unpruned models. The accuracy of the unpruned model on PIEs increases when increasing the amount of pruned parameters, while the accuracy of the pruned model decreases in the same setting. This is because the pruned model misclassifies more samples that are correctly classified by the unpruned model, increasing the amount of disagreement, hence the number of PIEs too.

B.2 Influential examples in PIEs

We report here the additional results of Section 4.2. Figures 14, 15, 16, 17, 18, 19, and 20 report the percentage of data points that are PIEs versus the degree of influence of all data points in the training set, for each pruning algorithm. PIEs are concentrated on the most influential examples. The higher the amount of pruned parameters, the more PIEs are distributed across examples with different influence on model generalization.

B.3 Textual characteristics of PIEs

We report here the additional results of Section 4.3.

In most cases, the formal education level needed to understand PIEs is higher than for all data points, with the exception of AAPD. AAPD leads to significant disagreement between pruned and unpruned models, even with 20% parameter pruning (See Table 4)). This is due to our extension of PIEs for multi-label settings, which considers a sample as a PIE if there is prediction disagreement on any class. The more classes in the dataset, the higher the chance of samples being labelled as PIEs. AAPD has 53 classes, the highest class count of

Single-label: Accuracy

dataset	model	pruning algo	0%	20%	50%	70%	90%	99%
IMDB	BERT	IIBP-WR	**.932 ± .005**	.892 ± .009	.870 ± .016	.864 ± .026	**.863 ± .011**	.742 ± .136
		IIBP-FT	**.932 ± .005**	.919 ± .004	.869 ± .008	.848 ± .007	.843 ± .010	**.828 ± .064**
		IBP-AI	**.932 ± .005**	.882 ± .010	.864 ± .021	.865 ± .016	.841 ± .069	.526 ± .079
		IMP-WR	**.932 ± .005**	.911 ± .009	**.880 ± .006**	**.870 ± .009**	.857 ± .007	.500 ± .000
		IMP-FT	**.932 ± .005**	**.924 ± .004**	.873 ± .007	.845 ± .004	.850 ± .007	.500 ± .000
		MP-AI	**.932 ± .005**	.904 ± .008	.871 ± .009	.867 ± .010	.852 ± .011	.500 ± .000
		IRP-FT	**.932 ± .005**	.877 ± .011	.846 ± .009	.778 ± .141	.500 ± .000	.500 ± .000
		RP-AI	**.932 ± .005**	.874 ± .004	.866 ± .012	.828 ± .114	.500 ± .000	.500 ± .000
	BiLSTM	IIBP-WR	**.879 ± .016**	.868 ± .021	.861 ± .026	.856 ± .027	.837 ± .025	.806 ± .026
		IIBP-FT	**.879 ± .016**	.883 ± .011	.880 ± .013	.878 ± .010	.872 ± .011	**.872 ± .013**
		IBP-AI	**.879 ± .016**	.874 ± .017	.848 ± .019	.820 ± .032	.805 ± .029	.755 ± .022
		IMP-WR	**.879 ± .016**	.875 ± .020	.881 ± .012	.876 ± .011	.873 ± .025	.804 ± .018
		IMP-FT	**.879 ± .016**	**.886 ± .010**	**.887 ± .010**	**.882 ± .009**	**.878 ± .007**	.862 ± .013
		MP-AI	**.879 ± .016**	.872 ± .019	.855 ± .018	.843 ± .023	.834 ± .015	.755 ± .021
		IRP-FT	**.879 ± .016**	.885 ± .010	.875 ± .011	.875 ± .012	.873 ± .017	.548 ± .073
		RP-AI	**.879 ± .016**	.872 ± .037	.848 ± .027	.845 ± .026	.840 ± .016	.721 ± .024
SNLI	BERT	IIBP-WR	**.901 ± .002**	.849 ± .098	.822 ± .004	.794 ± .007	.683 ± .044	.578 ± .053
		IIBP-FT	**.901 ± .002**	.892 ± .002	**.876 ± .003**	**.857 ± .003**	**.806 ± .090**	**.654 ± .071**
		IBP-AI	**.901 ± .002**	.872 ± .002	.824 ± .005	.768 ± .028	.625 ± .016	.395 ± .086
		IMP-WR	**.901 ± .002**	.883 ± .003	.847 ± .004	.799 ± .004	.646 ± .033	.336 ± .008
		IMP-FT	**.901 ± .002**	**.895 ± .002**	.875 ± .002	.835 ± .004	.799 ± .005	.336 ± .008
		MP-AI	**.901 ± .002**	.882 ± .002	.833 ± .003	.691 ± .016	.616 ± .011	.335 ± .007
		IRP-FT	**.901 ± .002**	.885 ± .003	.836 ± .004	.785 ± .008	.342 ± .034	.336 ± .008
		RP-AI	**.901 ± .002**	.854 ± .004	.695 ± .007	.647 ± .005	.366 ± .069	.335 ± .007
	BiLSTM	IIBP-WR	**.778 ± .004**	**.780 ± .004**	.774 ± .005	.763 ± .005	.715 ± .007	.614 ± .007
		IIBP-FT	**.778 ± .004**	.742 ± .004	.750 ± .004	.762 ± .003	**.771 ± .004**	**.657 ± .011**
		IBP-AI	**.778 ± .004**	.776 ± .004	.766 ± .004	.743 ± .007	.669 ± .009	.431 ± .104
		IMP-WR	**.778 ± .004**	.779 ± .004	**.782 ± .004**	**.782 ± .004**	.726 ± .009	.336 ± .007
		IMP-FT	**.778 ± .004**	.741 ± .004	.746 ± .004	.766 ± .003	.765 ± .004	.574 ± .019
		MP-AI	**.778 ± .004**	.776 ± .004	.764 ± .006	.743 ± .005	.687 ± .007	.336 ± .007
		IRP-FT	**.778 ± .004**	.746 ± .004	.769 ± .004	.779 ± .004	.712 ± .007	.389 ± .070
		RP-AI	**.778 ± .004**	.776 ± .003	.762 ± .005	.739 ± .006	.667 ± .017	.336 ± .007

Multi-label: F1 Macro

dataset	model	pruning algo	0%	20%	50%	70%	90%	99%
Reuters	BERT	IIBP-WR	**.836 ± .004**	.822 ± .011	.792 ± .018	.674 ± .064	.382 ± .046	.167 ± .041
		IIBP-FT	**.836 ± .004**	.835 ± .005	.830 ± .005	.822 ± .008	**.786 ± .029**	**.355 ± .061**
		IBP-AI	**.836 ± .004**	.810 ± .008	.645 ± .048	.328 ± .048	.189 ± .027	.096 ± .018
		IMP-WR	**.836 ± .004**	.827 ± .006	.829 ± .005	.736 ± .015	.147 ± .025	.082 ± .008
		IMP-FT	**.836 ± .004**	**.838 ± .005**	**.834 ± .005**	**.824 ± .006**	.490 ± .086	.085 ± .005
		MP-AI	**.836 ± .004**	.822 ± .006	.745 ± .021	.417 ± .057	.127 ± .031	.086 ± .004
		IRP-FT	**.836 ± .004**	.832 ± .005	.769 ± .013	.524 ± .075	.087 ± .001	.087 ± .002
		RP-AI	**.836 ± .004**	.803 ± .007	.479 ± .052	.242 ± .021	.089 ± .012	.086 ± .003
	BiLSTM	IIBP-WR	**.731 ± .017**	.728 ± .018	.727 ± .016	.716 ± .014	.631 ± .036	.396 ± .040
		IIBP-FT	**.731 ± .017**	**.753 ± .018**	.751 ± .013	.751 ± .015	.742 ± .014	**.693 ± .019**
		IBP-AI	**.731 ± .017**	.729 ± .020	.706 ± .019	.616 ± .029	.456 ± .036	.224 ± .028
		IMP-WR	**.731 ± .017**	.726 ± .017	.738 ± .015	.745 ± .012	.734 ± .011	.481 ± .032
		IMP-FT	**.731 ± .017**	.751 ± .013	.747 ± .014	.745 ± .017	**.746 ± .012**	.657 ± .028
		MP-AI	**.731 ± .017**	.740 ± .012	.730 ± .014	.705 ± .022	.606 ± .026	.393 ± .034
		IRP-FT	**.731 ± .017**	**.753 ± .015**	**.757 ± .015**	**.760 ± .014**	.743 ± .012	.417 ± .042
		RP-AI	**.731 ± .017**	.731 ± .019	.724 ± .015	.661 ± .028	.570 ± .030	.377 ± .042
AAPD	BERT	IIBP-WR	**.578 ± .007**	.547 ± .008	.518 ± .009	.482 ± .010	.403 ± .018	.179 ± .032
		IIBP-FT	**.578 ± .007**	.573 ± .009	**.548 ± .009**	.462 ± .153	**.476 ± .018**	**.316 ± .033**
		IBP-AI	**.578 ± .007**	.539 ± .009	.480 ± .015	.398 ± .023	.234 ± .042	.091 ± .016
		IMP-WR	**.578 ± .007**	.567 ± .009	.541 ± .007	.483 ± .008	.230 ± .029	.080 ± .001
		IMP-FT	**.578 ± .007**	**.579 ± .009**	.546 ± .008	**.521 ± .008**	.400 ± .019	.080 ± .000
		MP-AI	**.578 ± .007**	.551 ± .008	.508 ± .010	.423 ± .014	.145 ± .007	.080 ± .000
		IRP-FT	**.578 ± .007**	.554 ± .009	.312 ± .197	.338 ± .133	.082 ± .014	.080 ± .000
		RP-AI	**.578 ± .007**	.524 ± .009	.397 ± .015	.261 ± .029	.082 ± .007	.080 ± .000
	BiLSTM	IIBP-WR	**.468 ± .015**	.449 ± .022	.441 ± .022	.444 ± .020	.346 ± .022	.163 ± .028
		IIBP-FT	**.468 ± .015**	.429 ± .013	.425 ± .015	.436 ± .009	**.486 ± .010**	**.396 ± .012**
		IBP-AI	**.468 ± .015**	**.473 ± .014**	.459 ± .011	.398 ± .029	.190 ± .027	.082 ± .004
		IMP-WR	**.468 ± .015**	.446 ± .018	.454 ± .016	.454 ± .013	.406 ± .014	.185 ± .019
		IMP-FT	**.468 ± .015**	.428 ± .014	.421 ± .013	.432 ± .015	**.486 ± .009**	.330 ± .020
		MP-AI	**.468 ± .015**	**.473 ± .015**	**.473 ± .010**	.454 ± .011	.385 ± .020	.165 ± .025
		IRP-FT	**.468 ± .015**	.432 ± .012	.451 ± .013	**.486 ± .012**	.475 ± .010	.167 ± .025
		RP-AI	**.468 ± .015**	.464 ± .014	.453 ± .018	.421 ± .023	.356 ± .021	.163 ± .025

Table 9: Average macro accuracy/F1 score and std over 30 model initializations. Pruning algo is the used pruning algorithm according to Table 3. The best results for each percentage of pruned parameters and combination of dataset and architecture are in bold.

		Single-label										
		Pruner	**20%**		**50%**		**70%**		**90%**		**99%**	
IMDB	BERT	IIBP-WR	0.245	0.755	0.200	0.800	0.191	0.809	0.188	0.812	0.182	0.818
		IIBP-FT	0.356	0.644	0.195	0.805	0.161	0.839	0.163	0.837	0.188	0.812
		IBP-AI	0.227	0.773	0.195	0.805	0.198	0.802	0.205	0.795	0.056	**0.944**
		IMP-WR	0.290	0.710	0.206	0.794	0.194	0.806	0.179	0.821	0.056	**0.944**
		IMP-FT	0.385	0.615	0.200	0.800	0.156	0.844	0.167	0.833	0.056	**0.944**
		MP-AI	0.262	0.738	0.197	0.803	0.192	0.808	0.180	0.820	0.056	**0.944**
		IRP-FT	0.220	0.780	0.161	0.839	0.172	0.828	0.056	**0.944**	0.056	**0.944**
		RP-AI	0.198	0.802	0.199	0.801	0.198	0.802	0.056	**0.944**	0.056	**0.944**
	BiLSTM	IIBP-WR	0.371	0.629	0.322	0.678	0.283	0.717	0.232	0.768	0.207	0.793
		IIBP-FT	0.604	0.396	0.616	0.384	0.598	0.402	0.555	0.445	0.471	0.529
		IBP-AI	0.382	0.618	0.253	0.747	0.209	0.791	0.206	0.794	0.168	0.832
		IMP-WR	0.471	0.529	0.542	0.458	0.480	0.520	0.470	0.530	0.218	0.782
		IMP-FT	0.644	0.356	0.658	0.342	0.584	0.416	0.613	0.387	0.395	0.605
		MP-AI	0.404	0.596	0.281	0.719	0.241	0.759	0.225	0.775	0.178	0.822
		IRP-FT	0.633	0.367	0.577	0.423	0.576	0.424	0.404	0.596	0.126	0.874
		RP-AI	0.403	0.597	0.269	0.731	0.250	0.750	0.230	0.770	0.161	0.839
SNLI	BERT	IIBP-WR	0.250	0.688	0.177	0.753	0.155	0.782	0.091	0.855	0.074	0.878
		IIBP-FT	0.438	0.468	0.301	0.635	0.235	0.692	0.173	0.752	0.090	0.859
		IBP-AI	0.265	0.676	0.177	0.756	0.139	0.805	0.075	0.872	0.049	0.882
		IMP-WR	0.284	0.658	0.212	0.721	0.149	0.792	0.084	0.867	0.044	**0.909**
		IMP-FT	0.397	0.525	0.287	0.648	0.194	0.746	0.149	0.788	0.044	**0.909**
		MP-AI	0.275	0.656	0.175	0.748	0.083	0.853	0.069	0.873	0.044	**0.909**
		IRP-FT	0.347	0.582	0.192	0.738	0.136	0.803	0.044	**0.909**	0.044	**0.909**
		RP-AI	0.224	0.709	0.087	0.855	0.074	0.873	0.044	**0.909**	0.044	**0.909**
	BiLSTM	IIBP-WR	0.445	0.464	0.356	0.549	0.278	0.618	0.208	0.682	0.153	0.750
		IIBP-FT	0.467	0.413	0.529	0.366	0.517	0.368	0.391	0.493	0.184	0.719
		IBP-AI	0.382	0.518	0.298	0.582	0.258	0.626	0.177	0.722	0.124	0.760
		IMP-WR	0.434	0.454	0.461	0.429	0.447	0.451	0.225	0.670	0.068	**0.824**
		IMP-FT	0.495	0.381	0.496	0.379	0.522	0.361	0.375	0.522	0.141	0.758
		MP-AI	0.397	0.490	0.296	0.596	0.247	0.640	0.196	0.705	0.068	**0.824**
		IRP-FT	0.512	0.389	0.535	0.340	0.498	0.393	0.215	0.677	0.101	**0.796**
		RP-AI	0.371	0.524	0.281	0.620	0.243	0.649	0.175	0.728	0.068	**0.824**
		Multi-label										
		Pruner	**20%**		**50%**		**70%**		**90%**		**99%**	
Reuters	BERT	IIBP-WR	0.575	0.620	0.561	0.664	0.545	0.777	0.319	0.807	0.167	**0.837**
		IIBP-FT	0.608	0.591	0.572	0.567	0.589	0.621	0.530	0.659	0.302	0.820
		IBP-AI	0.572	0.656	0.506	0.780	0.276	0.825	0.182	**0.838**	0.096	0.836
		IMP-WR	0.563	0.602	0.529	0.570	0.545	0.726	0.147	**0.837**	0.082	0.836
		IMP-FT	0.619	0.602	0.555	0.596	0.590	0.627	0.393	0.794	0.085	0.836
		MP-AI	0.555	0.610	0.555	0.743	0.359	0.819	0.127	0.836	0.086	0.836
		IRP-FT	0.604	0.621	0.530	0.714	0.422	0.806	0.087	0.836	0.087	0.836
		RP-AI	0.560	0.666	0.428	0.815	0.196	0.825	0.089	0.836	0.086	0.836
	BiLSTM	IIBP-WR	0.466	0.462	0.498	0.500	0.483	0.509	0.509	0.620	0.362	0.701
		IIBP-FT	0.476	0.423	0.490	0.440	0.511	0.442	0.508	0.432	0.496	0.509
		IBP-AI	0.452	0.459	0.489	0.529	0.501	0.620	0.422	0.708	0.193	0.720
		IMP-WR	0.464	0.470	0.445	0.448	0.483	0.451	0.521	0.485	0.435	0.696
		IMP-FT	0.519	0.462	0.521	0.452	0.504	0.443	0.526	0.447	0.496	0.577
		MP-AI	0.495	0.470	0.472	0.478	0.514	0.557	0.504	0.638	0.356	0.711
		IRP-FT	0.500	0.423	0.480	0.399	0.488	0.416	0.512	0.440	0.375	0.704
		RP-AI	0.453	0.446	0.510	0.517	0.510	0.593	0.507	0.676	0.346	0.710
AAPD	BERT	IIBP-WR	0.471	0.511	0.453	0.529	0.432	0.553	0.367	0.563	0.175	**0.580**
		IIBP-FT	0.498	0.506	0.476	0.515	0.417	0.542	0.418	0.556	0.292	0.576
		IBP-AI	0.463	0.515	0.430	0.548	0.366	0.566	0.229	**0.582**	0.091	0.578
		IMP-WR	0.492	0.507	0.475	0.525	0.428	0.552	0.225	**0.582**	0.080	0.578
		IMP-FT	0.502	0.506	0.484	0.532	0.462	0.537	0.366	0.569	0.080	0.578
		MP-AI	0.475	0.517	0.452	0.544	0.390	0.568	0.143	**0.579**	0.080	0.578
		IRP-FT	0.483	0.519	0.295	0.560	0.311	0.565	0.082	0.578	0.080	0.578
		RP-AI	0.448	0.516	0.364	0.568	0.255	**0.583**	0.082	0.578	0.080	0.578
	BiLSTM	IIBP-WR	0.391	0.416	0.405	0.443	0.413	0.445	0.333	0.461	0.160	**0.469**
		IIBP-FT	0.393	0.439	0.380	0.430	0.388	0.424	0.432	0.413	0.380	0.459
		IBP-AI	0.402	0.392	0.410	0.422	0.378	0.454	0.184	0.468	0.082	0.468
		IMP-WR	0.399	0.427	0.397	0.417	0.421	0.441	0.383	0.453	0.180	**0.469**
		IMP-FT	0.389	0.435	0.375	0.432	0.386	0.432	0.442	0.425	0.318	0.462
		MP-AI	0.393	0.385	0.434	0.430	0.421	0.439	0.365	0.457	0.162	**0.469**
		IRP-FT	0.386	0.431	0.413	0.429	0.436	0.411	0.448	0.439	0.164	0.468
		RP-AI	0.411	0.417	0.409	0.432	0.397	0.451	0.344	0.462	0.160	**0.470**

Table 10: Average pruned and unpruned models' effectiveness on PIEs when pruning 20, 50, 70, 90, and 99% of the parameters. For each pruning percentage column, the first value refers to the effectiveness of the pruned models on PIEs, the second value represents the effectiveness of the unpruned models on the same set of PIEs. We represent models' effectiveness through accuracy in Single-label and F1 macro in Multi-label settings. The blue colour identifies cases where the pruned models have higher effectiveness on PIEs than the unpruned ones. We represent in bold the cases where the effectiveness of the models on PIEs is higher than the effectiveness of the same models on the whole dataset instead.

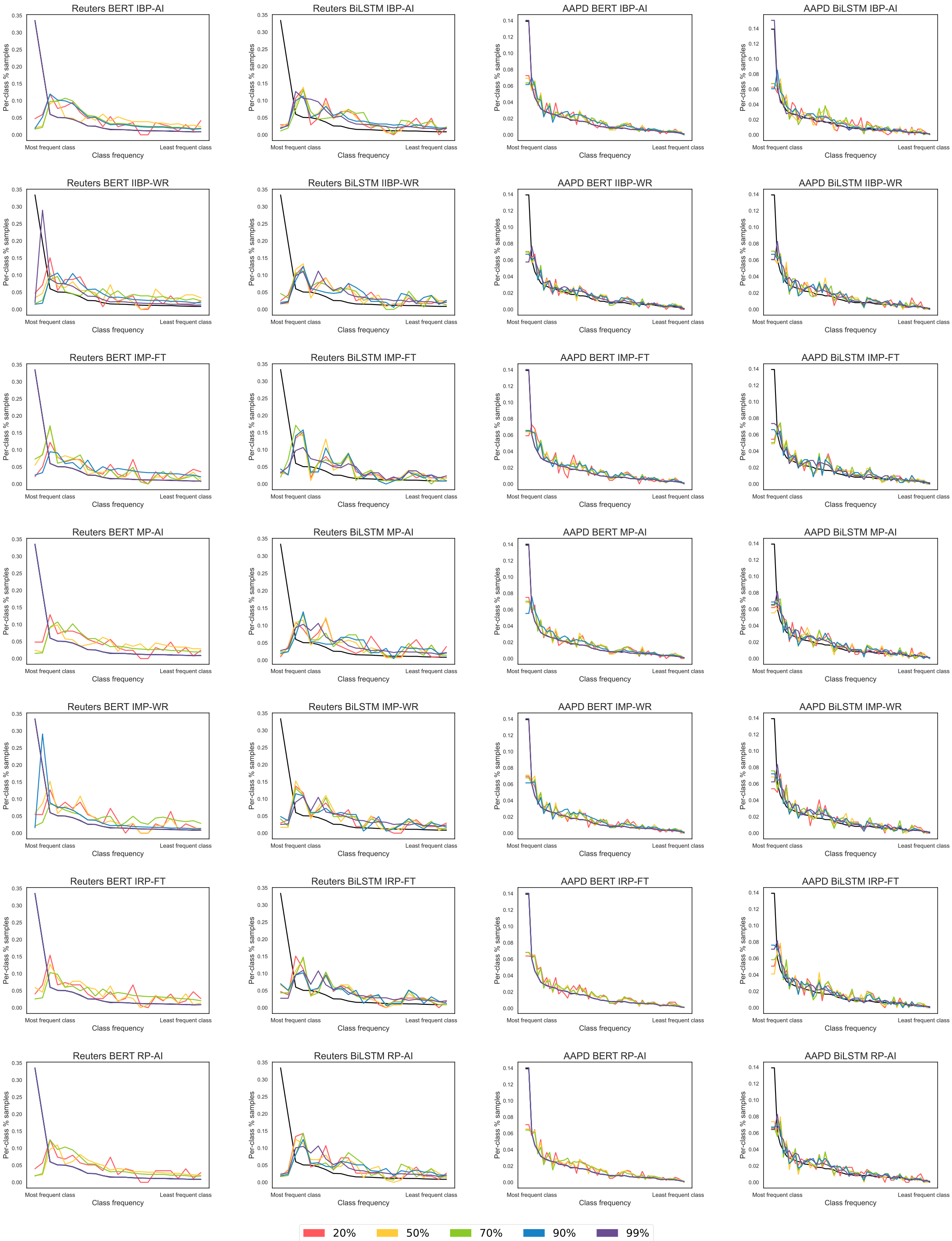

Figure 6: Distribution of all data points and of PIEs at 20% to 99% pruning, across classes sorted by frequency (x axis), for the multi-label datasets (test set).

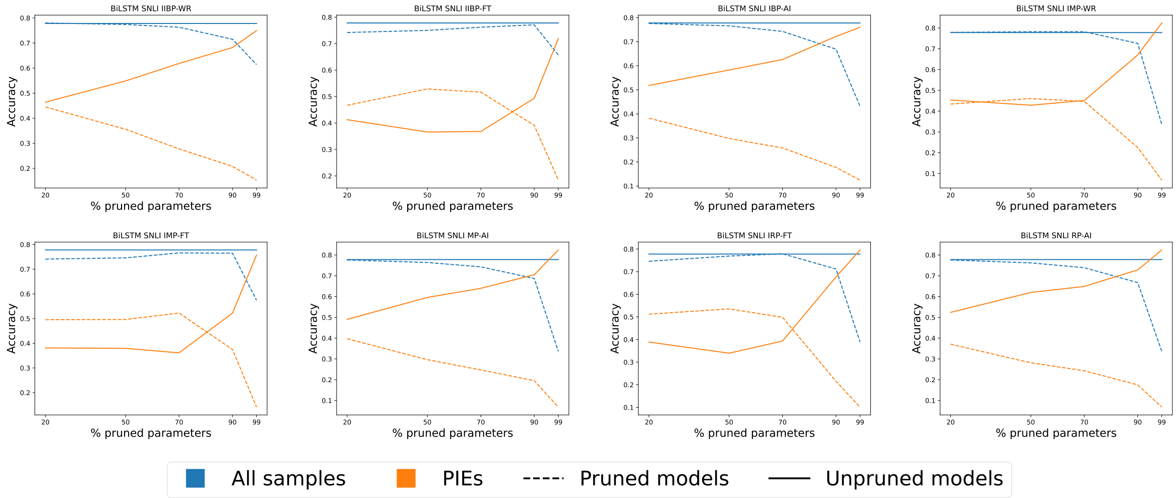

Figure 7: Accuracy of unpruned (black line) and pruned models on PIEs and all samples in the dataset per pruning method, across pruning thresholds (x-axis), over 30 initializations.

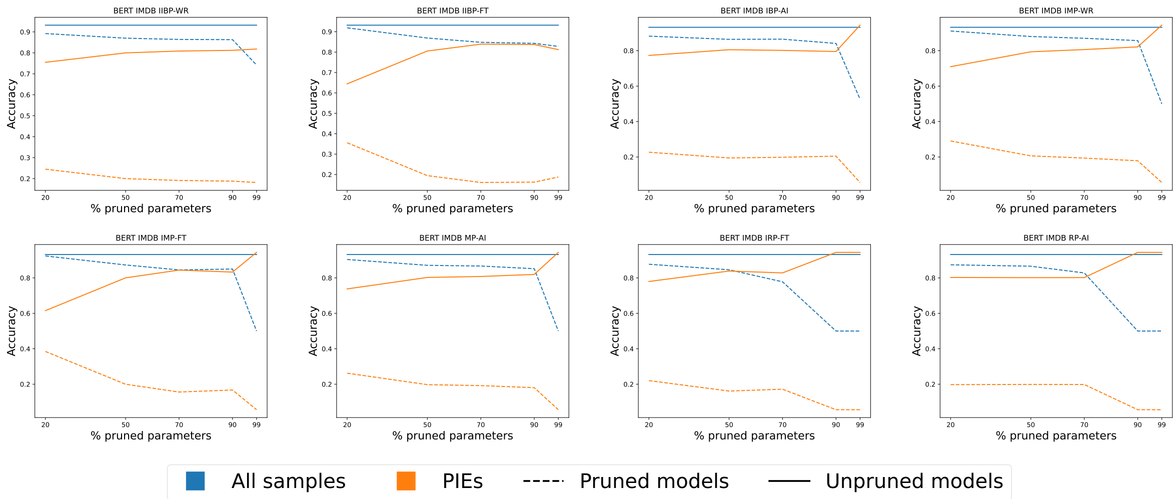

Figure 8: Accuracy of unpruned (black line) and pruned models on PIEs and all samples in the dataset per pruning method, across pruning thresholds (x-axis), over 30 initializations.

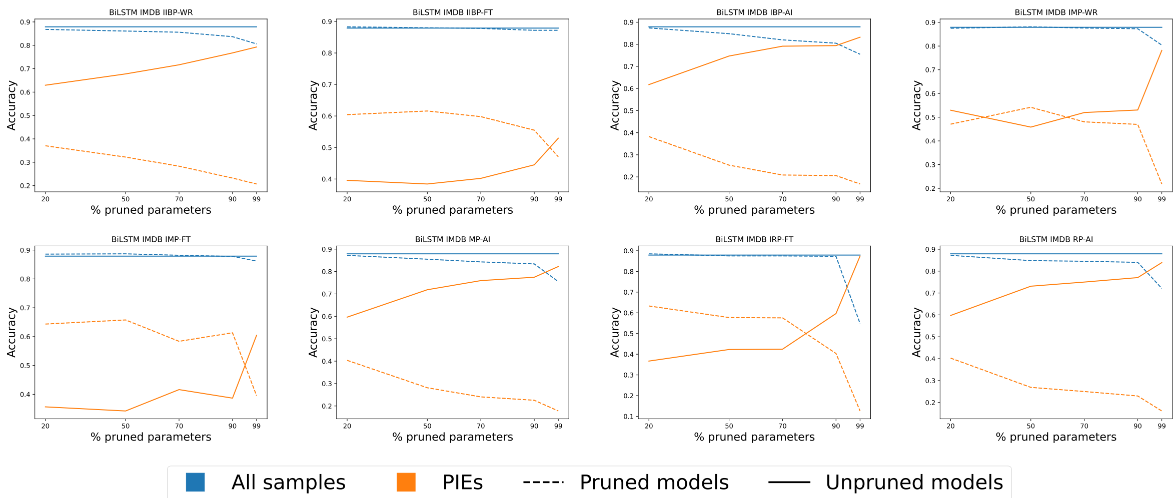

Figure 9: Accuracy of unpruned (black line) and pruned models on PIEs and all samples in the dataset per pruning method, across pruning thresholds (x-axis), over 30 initializations.

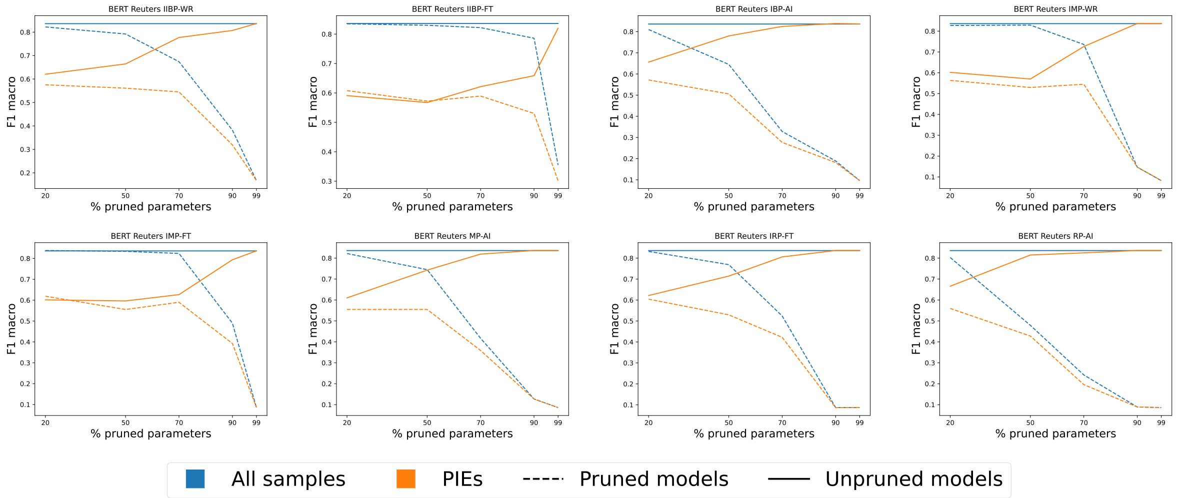

Figure 10: Accuracy of unpruned (black line) and pruned models on PIEs and all samples in the dataset per pruning method, across pruning thresholds (x-axis), over 30 initializations.

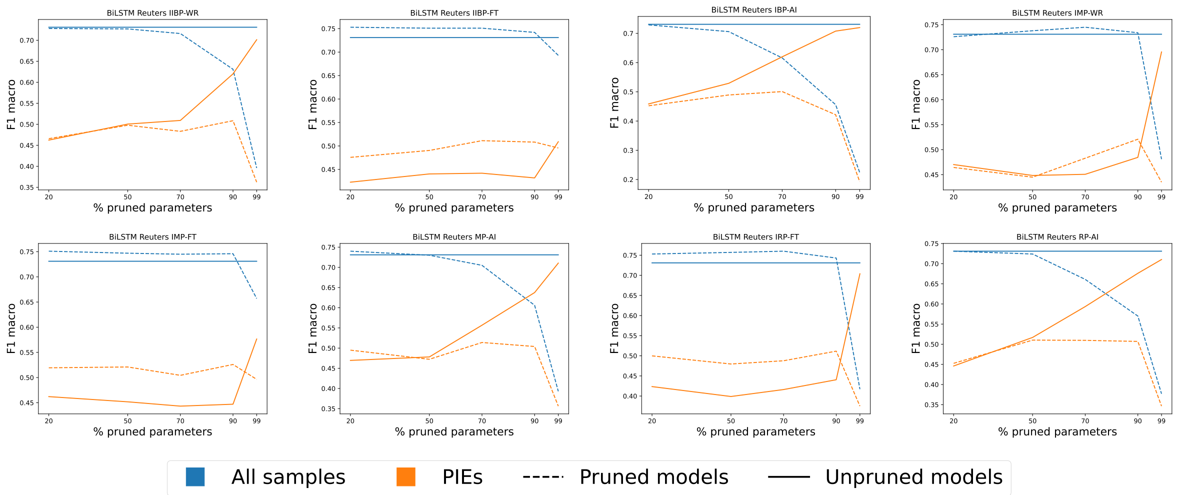

Figure 11: Accuracy of unpruned (black line) and pruned models on PIEs and all samples in the dataset per pruning method, across pruning thresholds (x-axis), over 30 initializations.

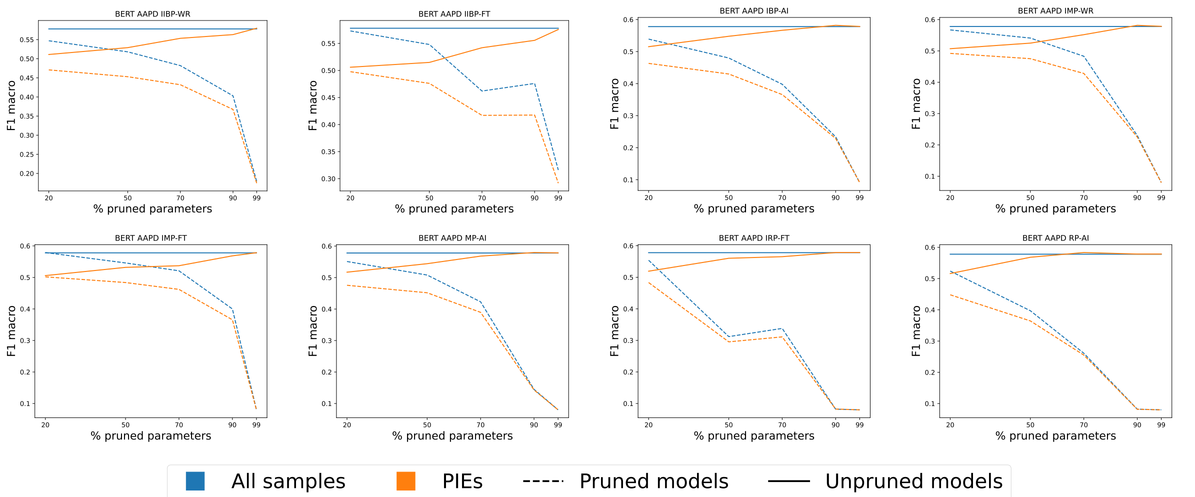

Figure 12: Accuracy of unpruned (black line) and pruned models on PIEs and all samples in the dataset per pruning method, across pruning thresholds (x-axis), over 30 initializations.

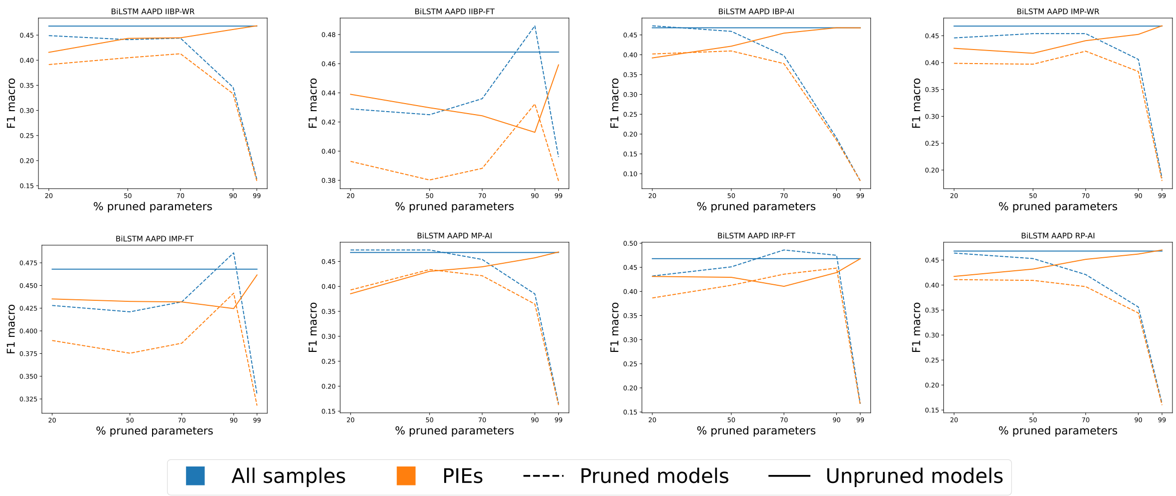

Figure 13: Accuracy of unpruned (black line) and pruned models on PIEs and all samples in the dataset per pruning method, across pruning thresholds (x-axis), over 30 initializations.

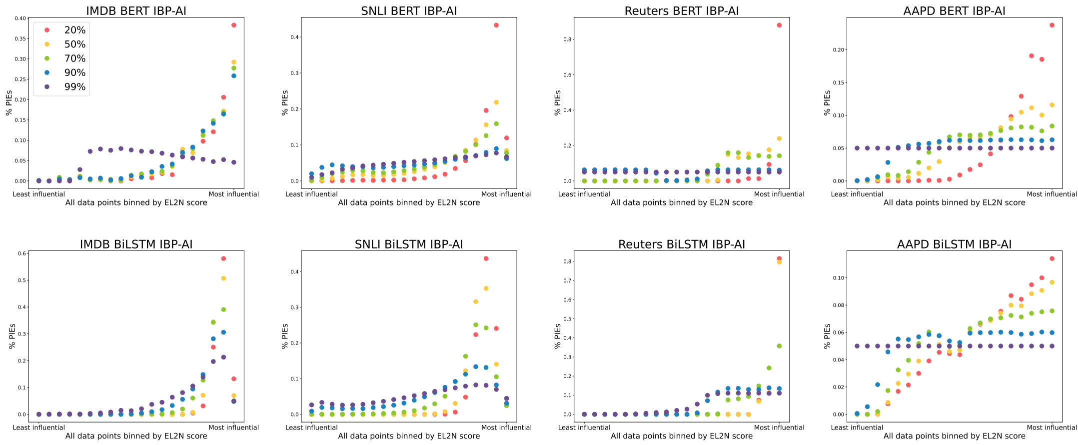

Figure 14: Percentage of data points that are PIEs (y axis) versus degree of influence (EL2N score) of all data points in the training set (x axis) for IBP-AI.

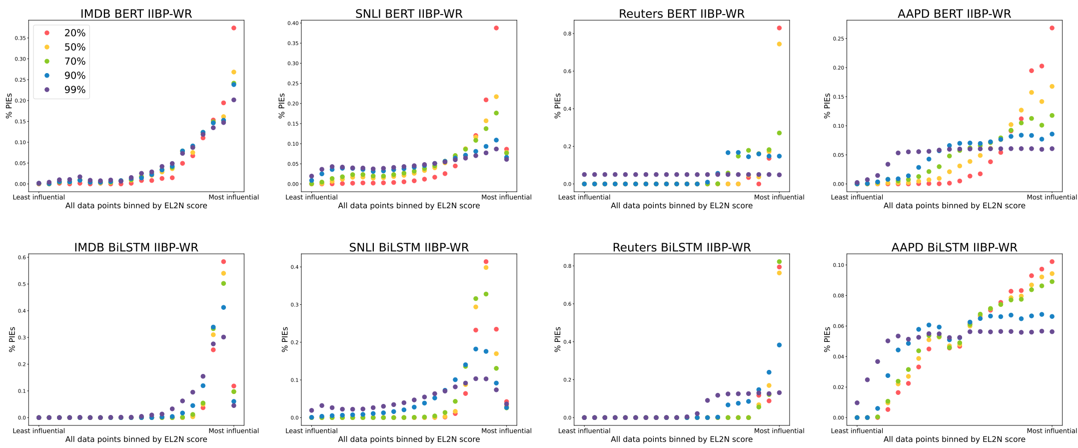

Figure 15: Percentage of data points that are PIEs (y axis) versus degree of influence (EL2N score) of all data points in the training set (x axis) for IIBP-WR at 20% and 99% pruning.

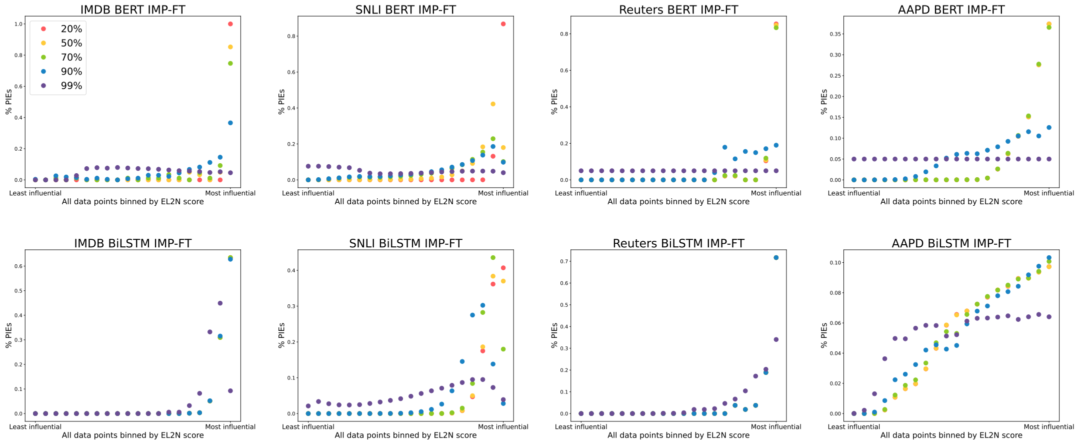

Figure 16: Percentage of data points that are PIEs (y axis) versus degree of influence (EL2N score) of all data points in the training set (x axis) for IMP-FT.

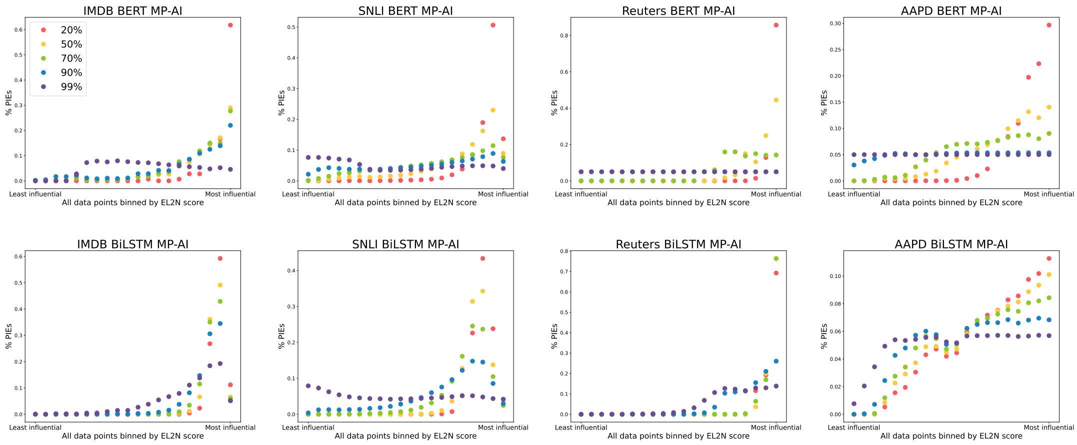

Figure 17: Percentage of data points that are PIEs (y axis) versus degree of influence (EL2N score) of all data points in the training set (x axis) for IMP-AI.

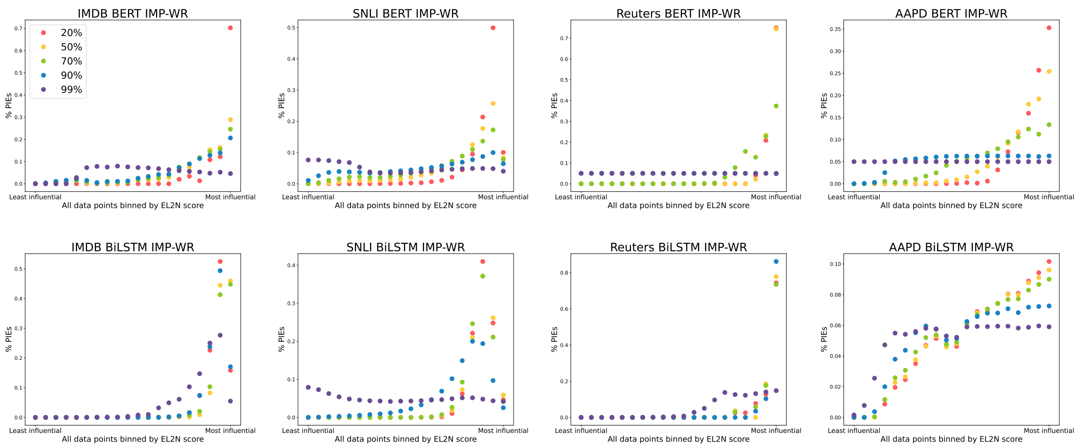

Figure 18: Percentage of data points that are PIEs (y axis) versus degree of influence (EL2N score) of all data points in the training set (x axis) for IMP-WR.

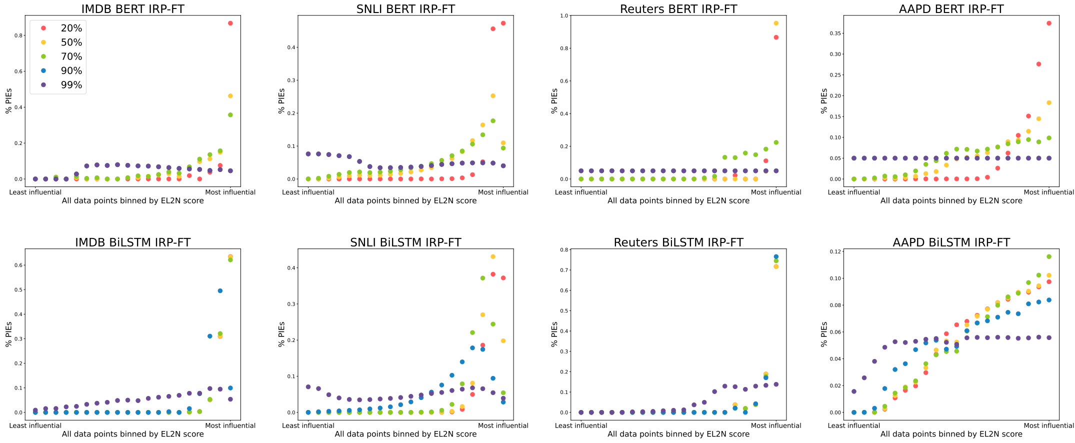

Figure 19: Percentage of data points that are PIEs (y axis) versus degree of influence (EL2N score) of all data points in the training set (x axis) for IRP-FT.

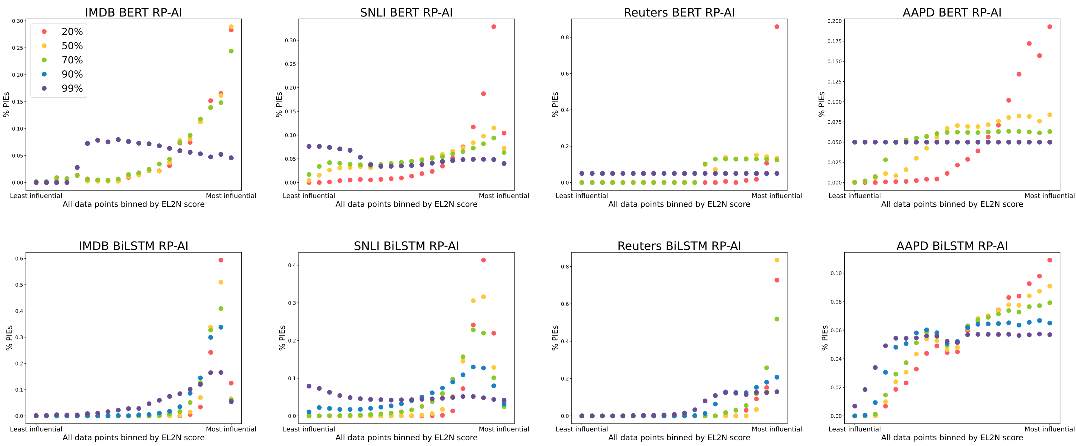

Figure 20: Percentage of data points that are PIEs (y axis) versus degree of influence (EL2N score) of all data points in the training set (x axis) for RP-AI.

all our datasets. As shown in the remaining settings, the more the disagreement between pruned and unpruned model predictions, the harder it is to observe a difference between the formal education level needed to understand PIEs and the dataset. Hence, on AAPD, we do not observe the same behaviour obtained in the three remaining datasets.

PIEs are overall longer than the text for all data points. PIEs can have up to 1.13 and 1.9 more tokens than the average number of tokens for a sample in the dataset for IMDB, and Reuters respectively. The behaviour can be observed with both BERT and BiLSTM models. About the ratio between the average number of tokens for the PIEs and in all the samples of the dataset on SNLI and AAPD datasets: we do not see the same behaviour as in IMDB and Reuters. SNLI is mostly made of short samples, hence it is harder to observe the behaviour on such a dataset, even if the trend is the same. On AAPD, the same observation on the formal education level needed to understand holds when discussing text length.

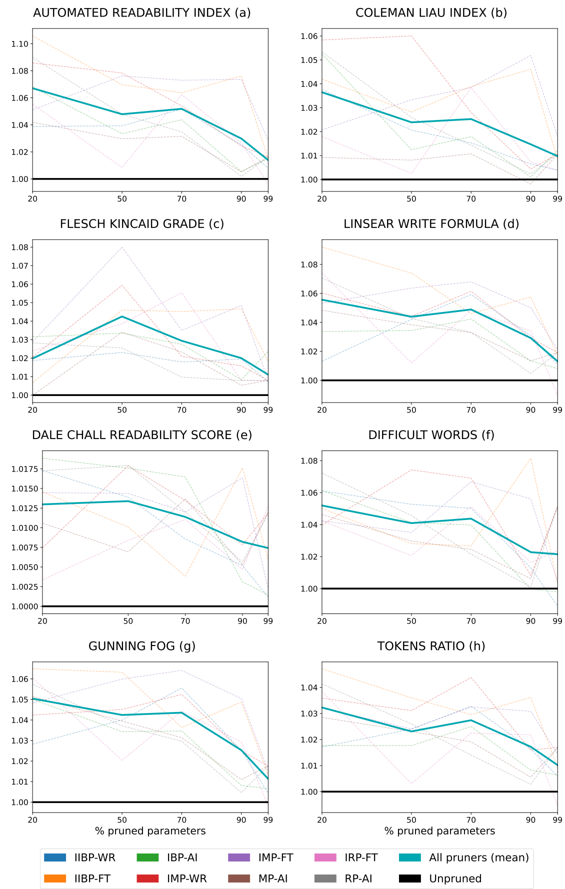

Figure 21: How the text of PIEs differs from the text of all data points, according to 7 readability scores (plots (a)-(g)) and text length (plot (h)). Ratio between the scores of PIEs and the scores of all data points (y axis), across pruning thresholds (x axis), for BiLSTM and SNLI. The solid black horizontal line represents equal scores in PIEs and all data points. The solid turquoise line is the mean score of all pruners. Any line above the solid black line means that PIEs are harder to understand (plots (a)-(g)) or have longer text (plot (h)), on average, than all data points.

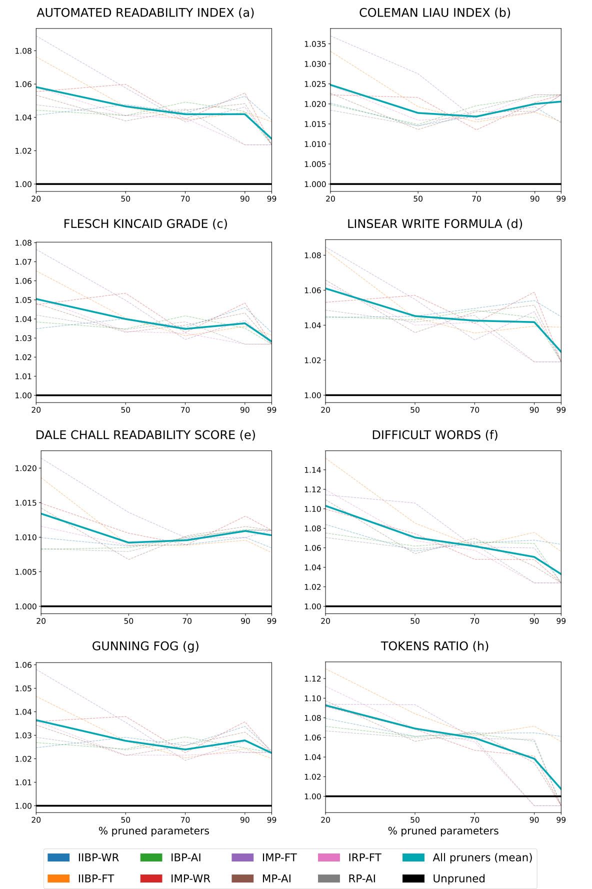

Figure 22: How the text of PIEs differs from the text of all data points, according to 7 readability scores (plots (a)-(g)) and text length (plot (h)). Ratio between the scores of PIEs and the scores of all data points (y axis), across pruning thresholds (x axis), for BERT and IMDB. The solid black horizontal line represents equal scores in PIEs and all data points. The solid turquoise line is the mean score of all pruners. Any line above the solid black line means that PIEs are harder to understand (plots (a)-(g)) or have longer text (plot (h)), on average, than all data points.

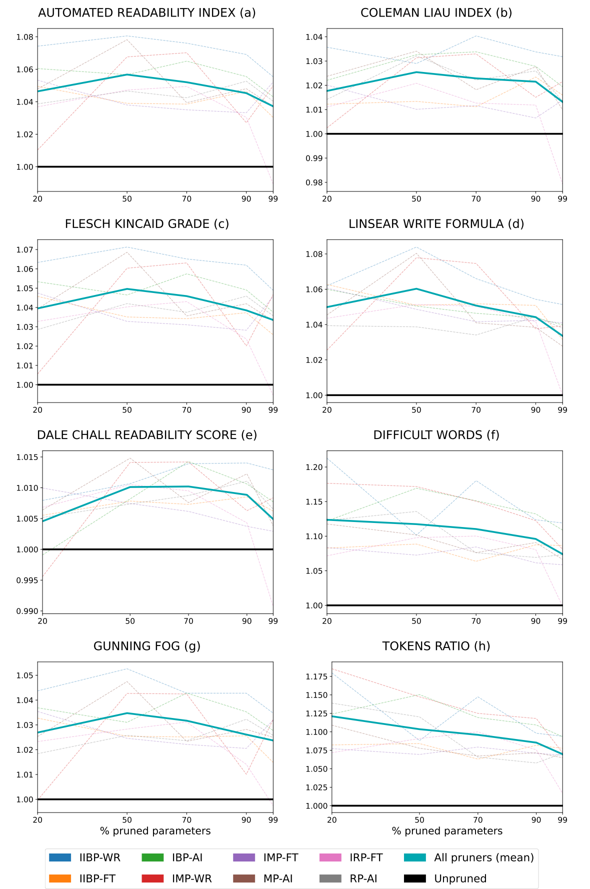

Figure 23: How the text of PIEs differs from the text of all data points, according to 7 readability scores (plots (a)-(g)) and text length (plot (h)). Ratio between the scores of PIEs and the scores of all data points (y axis), across pruning thresholds (x axis), for BiLSTM and IMDB. The solid black horizontal line represents equal scores in PIEs and all data points. The solid turquoise line is the mean score of all pruners. Any line above the solid black line means that PIEs are harder to understand (plots (a)-(g)) or have longer text (plot (h)), on average, than all data points.

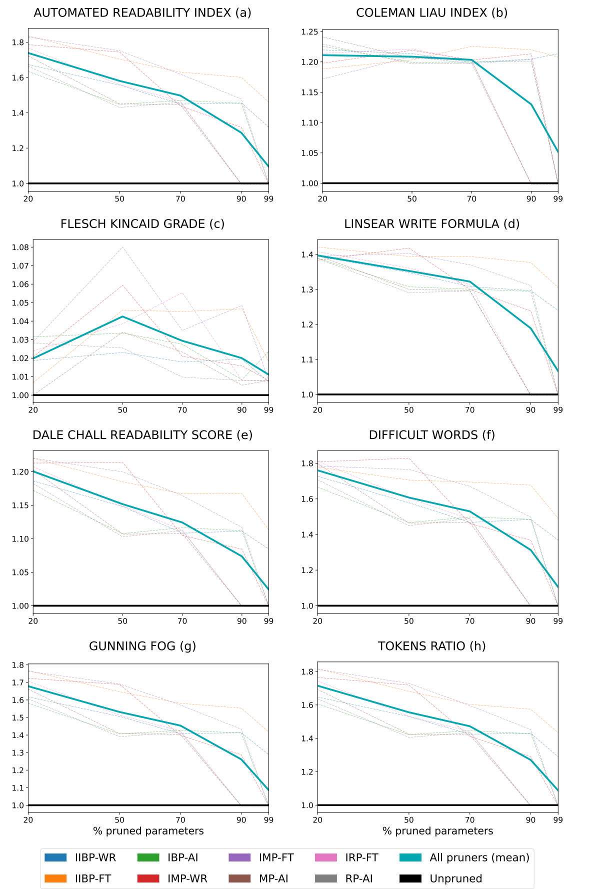

Figure 24: How the text of PIEs differs from the text of all data points, according to 7 readability scores (plots (a)-(g)) and text length (plot (h)). Ratio between the scores of PIEs and the scores of all data points (y axis), across pruning thresholds (x axis), for BERT and Reuters. The solid black horizontal line represents equal scores in PIEs and all data points. The solid turquoise line is the mean score of all pruners. Any line above the solid black line means that PIEs are harder to understand (plots (a)-(g)) or have longer text (plot (h)), on average, than all data points.

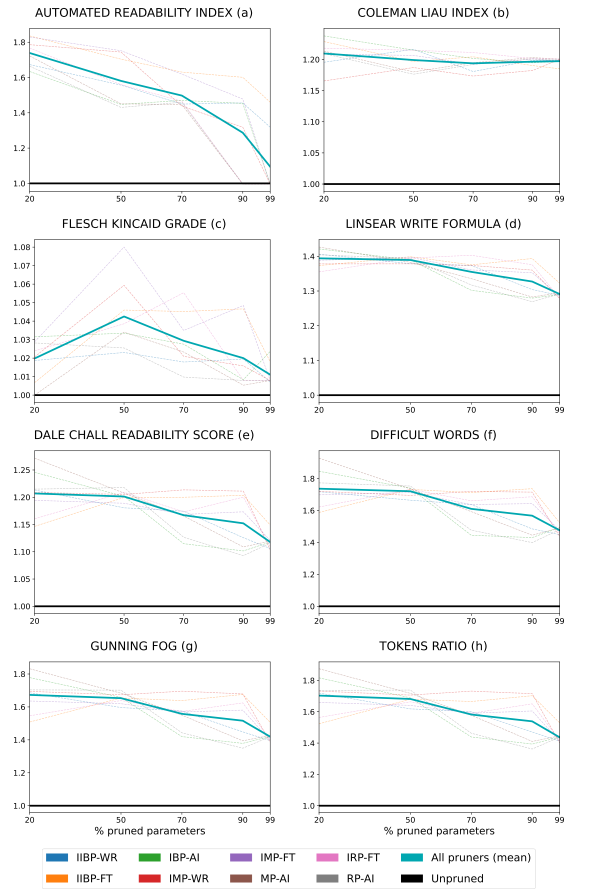

Figure 25: How the text of PIEs differs from the text of all data points, according to 7 readability scores (plots (a)-(g)) and text length (plot (h)). Ratio between the scores of PIEs and the scores of all data points (y axis), across pruning thresholds (x axis), for BiLSTM and Reuters. The solid black horizontal line represents equal scores in PIEs and all data points. The solid turquoise line is the mean score of all pruners. Any line above the solid black line means that PIEs are harder to understand (plots (a)-(g)) or have longer text (plot (h)), on average, than all data points.

Figure 26: How the text of PIEs differs from the text of all data points, according to 7 readability scores (plots (a)-(g)) and text length (plot (h)). Ratio between the scores of PIEs and the scores of all data points (y axis), across pruning thresholds (x axis), for BERT and AAPD. The solid black horizontal line represents equal scores in PIEs and all data points. The solid turquoise line is the mean score of all pruners. Any line above the solid black line means that PIEs are harder to understand (plots (a)-(g)) or have longer text (plot (h)), on average, than all data points.

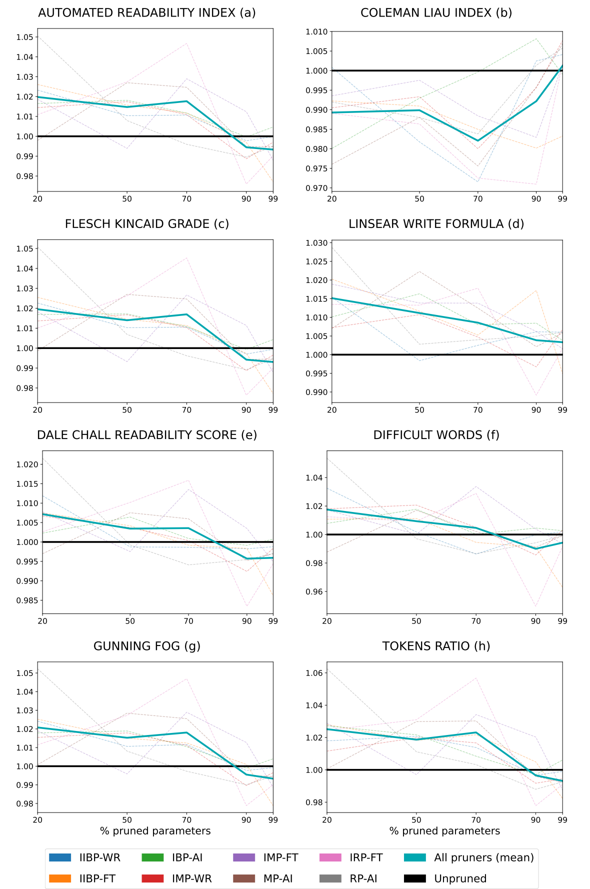

Figure 27: How the text of PIEs differs from the text of all data points, according to 7 readability scores (plots (a)-(g)) and text length (plot (h)). Ratio between the scores of PIEs and the scores of all data points (y axis), across pruning thresholds (x axis), for BiLSTM and AAPD. The solid black horizontal line represents equal scores in PIEs and all data points. The solid turquoise line is the mean score of all pruners. Any line above the solid black line means that PIEs are harder to understand (plots (a)-(g)) or have longer text (plot (h)), on average, than all data points.