Analyzing Biases to Spurious Correlations in Text Classification Tasks

Adian Liusie
Cambridge University
al826@cam.ac.uk

Vatsal Raina
Cambridge University
vr311@cam.ac.uk

Vyas Raina
Cambridge University
vr313@cam.ac.uk

Mark Gales
Cambridge University
mjfg@cam.ac.uk

Abstract

Machine learning systems have shown impressive performance across a range of natural language tasks. However, it has been hypothesized that these systems are prone to learning spurious correlations that may be present in the training data. Though these correlations will not impact in-domain performance, they are unlikely to generalize well to out-of-domain data, limiting the applicability of systems. This work examines this phenomenon on text classification tasks. Rather than artificially injecting features into the data, we demonstrate that real spurious correlations can be exploited by current state-of-the-art deep-learning systems. Specifically, we show that even when only 'stop' words are available at the input stage, it is possible to predict the class significantly better than random. Though it is shown that these stop words are not required for good in-domain performance, they can degrade the ability of the system to generalize well to out-of-domain data [1].

1 Introduction

Machine learning systems have shown impressive performance across a wide range of natural language processing (NLP) tasks such as question-answering, sentiment classification and summarization (Zhang et al., 2021; Sun et al., 2019; Aghajanyan et al., 2020). Often these systems reach or even exceed human performance (Bajaj et al., 2022), which has led to increasing deployment of these automatic systems in real-world applications. There is, however, a caveat to the superhuman claim: standard benchmarks (Rajpurkar et al., 2016; Wang et al., 2018) often assume that the training and evaluation data are drawn independently and identically from the same underlying distribution, an assumption that is rarely valid in the real world due to different deployment environments and constantly evolving evaluation distributions (Quiñonero-Candela et al., 2008). High performance on the in-domain test set demonstrates that the system goes beyond memorization to successfully handle unseen examples. However this may only be true for a restricted domain, and hence the model may not generalize well to examples outside the training domain (Hendrycks and Dietterich, 2019).

An obstacle for generalization of machine learning systems is caused by the presence of spurious correlations. For example, in sentiment classification there may be a bias in the training data such that positive examples are longer than negative examples. In such scenarios, a model may use length as a significant feature to classify, which is problematic since length is 'a spurious feature' and should not provide sentiment information. Although the model may still have good performance on the in-domain test set (where this bias holds), reliance on this spurious feature may cost generalizability for real world out-of-domain (OOD) settings as it distracts the system from learning the true underlying 'core' features of the task (Lapuschkin et al., 2019). Biases have been studied in literature, where the focus is primarily on ensuring models don't use sensitive properties such as gender and race (Blodgett et al., 2020). In this work we are instead concerned with biases to other less sensitive spurious correlations.

Spurious correlations have been explored in NLP (Eisenstein, 2022). Many 'shortcuts' (spurious features with high in-domian correlation, Geirhos et al. (2020)) have been found for many NLP tasks: Lovering et al. (2021) show that NLP models are prone to relying on spurious features provided they are easy to extract, Cai et al. (2017) show that neural models are able to complete sto-

[1] GitHub Repository: https://github.com/adianliusie/stopword-bias

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 78–84
November 20–23, 2022.

ries using only the final sentence, while Gururangan et al. (2018) show that clues left in the hypothesis are alone sufficient to achieve reasonable natural language inference performance. It is further shown that when such models are evaluated on adversarial data sets where the spurious correlations are eliminated (Zellers et al., 2019; Bhagavatula et al., 2019; Hendrycks et al., 2021), model performance drops drastically.

This work diverges from the standard setup, and instead examines the susceptibility of models learning biases from innocuous, unimportant features. In particular, we explore the predictive abilities of 'stop' words such as 'and', 'of' and 'the' for a range of varying text classification tasks. We further explore whether models rely on such spurious correlations and make biased decisions in OOD settings.

2 Spurious correlations

Spurious features have no causal relationship with the labels, but have strong correlations with the labels within a specific domain. More precisely, for input $\mathbf{x}$ and its corresponding label y, a model $\mathcal{M}$ aims to approximate the underlying distribution $p(y|\mathbf{x})$ for all $(\mathbf{x}, y) \in \mathcal{D}$, where $\mathcal{D}$ is the entire input-output space of the task. Typically, data is sampled from a restricted domain, $\mathcal{D}_a \subset \mathcal{D}$. Let $f_s(\cdot)$ denote a spurious feature extractor. Spurious features can be used effectively for prediction in the restricted domain $\mathcal{D}_a$ (Equation 1), but they have no causal link to the label in the general domain (Equation 2) and so are ineffective for prediction.

$$p(y|f_s(\mathbf{x})) \approx p(y|\mathbf{x}), \quad (\mathbf{x}, y) \in \mathcal{D}_a \quad (1)$$

$$p(y|f_s(\mathbf{x})) \approx p(y), \quad (\mathbf{x}, y) \in \mathcal{D} \quad (2)$$

We focus on identifying real spurious features in NLP tasks with significant correlations with the labels. These spurious correlations will consequently lead to biases in trained models, which though valid in-domain, may compromise OOD performance where the spurious correlations do not hold.

2.1 Shuffled stop words

We investigate the influence of stop words as real spurious features. Stop words were chosen because they mainly play a syntactic role in text and have low information content, and so are unlikely to be essential for text classification tasks. Also, due to the high frequency of stop words in language, models are prone to picking up distributional biases.

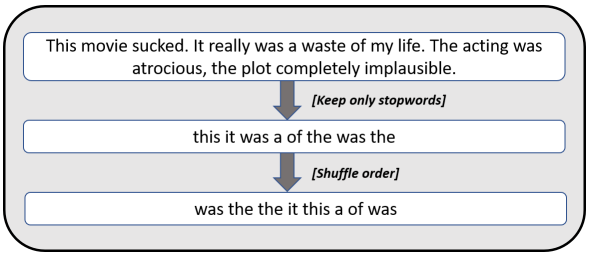


Figure 1: Corruption process on an example.

We introduce the shuffled stop words (SSW) evaluation setup where inputs are altered so that systems are forced to make predictions using only the stop words. Figure 1 outlines this process where first the input text is filtered to only retain the stop words[2] and the remaining words are then randomly shuffled to eliminate positional information. Hence, from the human perspective, this representation should have no causal relationship with the output label and any predictive bias must be solely due to the spurious features associated with the distribution of stop words.

2.2 Measuring stop word bias

We use the likelihood ratio as a statistical method to identify the degree of stop word bias present in a given binary classification corpus (where each example is either positive or negative). Let $\mathcal{S}$ be the set of all stop words. The distributions $P(x)$ and $Q(x)$ each assign every stop word $x \in \mathcal{S}$ a probability score proportional to the occurrences of x in all the samples for the positive and negative classes respectively. For input text $\mathbf{x}$ with words $(x_1, x_2, \ldots, x_n)$, the log of the likelihood likelihood ratio (Equation 3) can be used as a handcrafted feature $f_s^{(\text{sw})}(\mathbf{x})$ that is a proxy to measure whether $\mathbf{x}$ uses stop words more associated with the positive class than with the negative class.

$$f_s^{(\text{sw})}(\mathbf{x}) = \log \frac{\prod_i \mathbb{I}(x_i \in \mathcal{S}) P(x_i)}{\prod_i \mathbb{I}(x_i \in \mathcal{S}) Q(x_i)} \quad (3)$$

For a given dataset, to visualize the extent of a bias for a defined feature, we propose using retention plots. To generate retention plots, the feature score for each example is first calculated (i.e. $f_s^{(\text{sw})}(\mathbf{x})$) and the examples are then sorted based on the score. For a retention fraction of r, the plot displays the fraction of total positive examples found when only $(100 \cdot r)\%$ of examples with the lowest feature score

[2]Stop words are taken from NLTK: https://gist.github.com/sebleier/554280.

are retained. Therefore, if the defined feature is completely independent of the labels, one would expect the retention plot to be the straight line $y = r$ (`no bias` line). However if the chosen feature orders the examples such that the two classes are perfectly separable at a given threshold, then for a balanced dataset there will be a flat line up to r=0.5 (as there are no positive examples), followed by a steep increment since all the following examples are positive (`full bias` line) e.g. Figure 2.

3 Experiments

3.1 Data

Data	imdb	rt	twitter	sst	yelp	boolq
train	20k	8530	16k	6920	448k	9426
val	5k	1066	2k	872	112k	3270
test	25k	1840	2k	1820	38k	3270

Table 1: Dataset splits' sizes

We consider several binary text classification tasks. IMDB (Maas et al., 2011), Rotten Tomatoes (RT) (Pang and Lee, 2005) and the Stanford Sentiment Treebank v2 dataset (SST) (Socher et al., 2013) are movie review datasets (positive/negative), which are sourced from different movie review platforms. Twitter's Emotion dataset (Saravia et al., 2018) categorizes tweets into one of six emotions, which are mapped to either positive (love, joy and surprise) or negative (fear, sadness and anger) to ensure the task is binary. The Yelp dataset (Zhang et al., 2015) consists of reviews from the Yelp platform, where the scores of 1-5 stars are split into positive (4,5) and negative (1,2) reviews. Finally, BoolQ (Clark et al., 2019) is a reading comprehension dataset where each example is a triplet of question, passage and answer (*yes/no*). Although most datasets are naturally balanced, if necessary the different dataset splits are filtered to be perfectly balanced. Table 1 gives the sizes of the train and test splits of all the datasets after processing.

3.2 Setup

Since pre-trained transformers have ubiquitously shown the best performance in NLP, we consider the pre-trained BERT model as the baseline (Devlin et al., 2019). We also consider a randomly initialized transformer (RIT) model with a BERT-based architecture to determine the impact of pre-training. All results are reported using ensembles of three models for each experiment [3].

3.3 Results

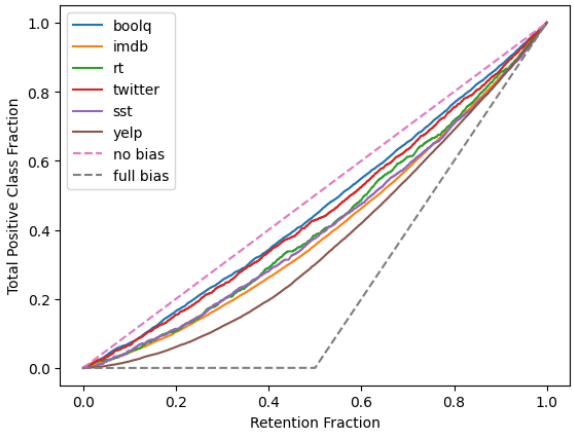


Figure 2: Retention plots for stop word bias.

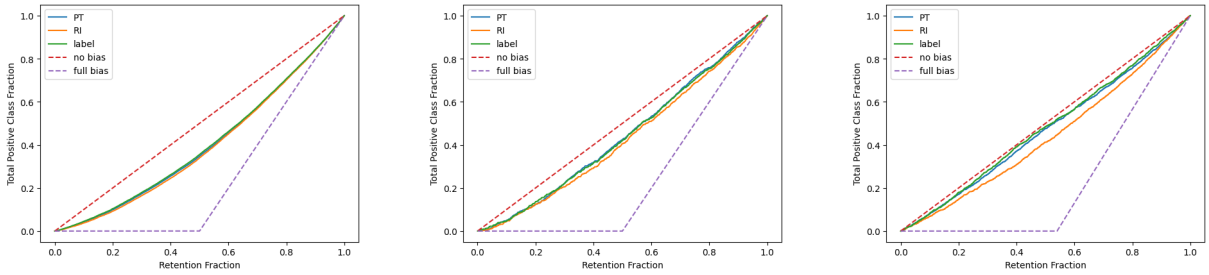
We first investigate whether stop word biases exist in text classification tasks, and if so, determine the prevalence of the bias. For each corpora, the log of the likelihood ratio (Equation 3) is calculated over the training split, and the retention plots are then plotted over the unseen test labels. Figure 2 shows the retention plots (described in Section 2.2) for various corpora, where for each corpus the significant deviations from the `no bias` line show that considerable correlations can be found between stop words and the labels.

To quantify how much information lies in these spurious features, we fine-tune a BERT model using only the shuffled stop words of the input text (and also evaluate it in the SSW setting). We compare this to the baseline, where BERT is fine-tuned in the standard setting, and also to the log of the likelihood ratio (LR) [4]. The results presented in Table 2 show that, surprisingly, stop words alone can be used to achieve reasonable in-domain performance across various text classification tasks. For all considered tasks, performance of both SSW and LR is significantly higher than the expected random value of 50%, with SSW accuracy at even 77% and 69% for yelp and IMDB respectively.

Although we establish significant correlations exist between the stop words and labels, a more practical consideration is to determine whether these spurious correlations impact model predictions. For this, we focus on sentiment classification. To simulate distributional shift, we use IMDB as in-domain,

[3] Training details provided in Appendix A.

[4] If $f_s^{(\text{sw})}(\mathbf{x}) > 0$ then $\mathbf{x}$ positive otherwise negative.



(a) In-domain (b) Shifted domain (c) Out-of-domain

Figure 3: Ranked by spurious stop word distribution feature retention plots.

	imdb	rt	twitter	sst	yelp	boolq
stndrd	94.2	85.2	98.4	92.4	97.6	66.9
LR	64.3	60.4	58.2	62.4	70.4	57.5
SSW	68.7	60.5	57.8	60.3	77.3	63.1

Table 2: BERT model accuracy (%).

RT as the shifted-domain and Twitter as OOD[5].

	standard			SSW		
Model	in	shift	out	in	shift	out
BERT	94.2	82.1	71.2	57.7	53.7	50.0
RIT	88.2	73.7	59.1	60.0	57.3	50.5

Table 3: Accuracy (%), trained on IMDB (*standard*) and evaluated in both the *standard* and *SSW* settings.

Table 3 displays model performance when trained on the in-domain data and then evaluated across the various domains. For standard evaluation, we observe that pre-training leads to a performance improvement of 6% and is more robust to domain changes, with BERT dropping by 12.1% on the shifted domain and 23.0% on the OOD, while RIT drops by 14.5% and 29.1% respectively. The same systems are evaluated using SSW evaluation. We find that although the models were all trained with full text inputs, when evaluated on the shuffled stop words, the models all show 57%+ in-domain performance, providing evidence that models identify spurious stop word correlations.

To determine whether models truly rely on spurious features, we again generate retention plots. The retention plot is computed using the likelihood ratio (Equation 3) on the in-domain training set such that, irrespective of the evaluation domain, examples are sorted based on the IMDB training stop word distribution. To measure the models' inherent bias, we plot the retention curve with respect to the different models' predictions. That is, for a model's retention plot, an example is considered positive if the model predicted the example was positive.

The OOD retention plot shows that models are susceptible to learning the spurious in-domain stop word correlations. The significant deviation of RIT from the true labels shows that the model's scores are correlated with the in-domain stop word distribution, indicating the model has learned a stop word bias. Note that BERT only shows a mild bias to the stop words, which provides evidence that pre-trained models are more robust to relying on spurious features which may explain their better OOD generalizability (Hendrycks et al., 2020).

4 Conclusions

This work investigates the influence of spurious biases in standard text classification tasks. It is established that the stop word distributions of the positive and negative classes are substantially different, and this acts as a significant bias for several tasks including sentiment classification and question-answering. In particular, after corrupting an input example to only retain the shuffled stop words, a standard transformer-based language model achieves reasonable performance across tasks despite no meaningful task-specific information. It is further demonstrated that language models pick up on the training data's stop word distribution bias. Though, the spurious bias does not harm performance, when evaluated in-domain we observe that a randomly initialized transformer model maintains the spurious bias in OOD settings too where the same stop word bias does not hold. Hence, the learnt stop word bias from in-domain influences the predictions of the model in OOD, leading to performance degradation. Future work will investigate post-processing techniques to mitigate such spurious biases in deployed systems.

[5]Equivalent results for Yelp & SST given in Appendix B.

5 Acknowledgements

This research is funded by the EPSRC (The Engineering and Physical Sciences Research Council) Doctoral Training Partnership (DTP) PhD studentship and supported by Cambridge Assessment, University of Cambridge and ALTA.

6 Limitations

This work reveals that systems tend to be biased to stop-word distributions and this can contribute to a lack of generalization in out of domain settings. Nevertheless, this work is currently restricted to the task of text classification. It would be useful to investigate how stop word biases behave in other tasks, such as entailment, machine reading comprehension and grammatical error detection. Future work will also explore methods to correct for the stop word bias.

7 Risks and Ethics

There are no known ethical concerns or risks associated with the findings of this work.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2020. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*.

Payal Bajaj, Chenyan Xiong, Guolin Ke, Xiaodong Liu, Di He, Saurabh Tiwary, Tie-Yan Liu, Paul Bennett, Xia Song, and Jianfeng Gao. 2022. Metro: Efficient denoising pretraining of large scale autoencoding language models with model generated signals. *arXiv preprint arXiv:2204.06644*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending:strong neural baselines for the ROC story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 616–622, Vancouver, Canada. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Eisenstein. 2022. Informativeness and invariance: Two perspectives on spurious correlations in natural language.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL-HLT (2)*.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2021. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. Predicting inductive biases of pretrained models. In *ICLR*.

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.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2008. *Dataset shift in machine learning*. Mit Press.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification . *arXiv:1509.01626 [cs]*.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2021. Retrospective reader for machine reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14506–14514.

Appendix A Training Details

This section details the training regimes and hyperparameter tuning process for the BERT and the randomly initialised transformer (RIT) models. The BERT pretrained language model is based on BERT-base-uncased [6] with 110M parameters per single model. An ensemble of 3 members is trained for each task. All input samples were truncated to 512 tokens. Grid search was performed for hyperparamter tuning with the initial setting of hyperparameter values motivated from the baseline systems of . Besides the default values for the standard hyperparamters, grid search was performed for the learning rate $\in \{1e^{-5}, 2e^{-5}, 5e^{-5}\}$ and the batch size $\in \{4, 8, 16\}$. The final hyperparameter settings included training for a maximum of 4 epochs with early-stopping on the validation split at a learning rate of $1e^{-5}$ with a batch size of 8. Equivalent hyperparamter settings were used for RIT. Cross-entropy loss was used at training time with models built using Titan RTX graphical processing units with training time under 2 hours for all datasets (except for Yelp which takes 4 hours).

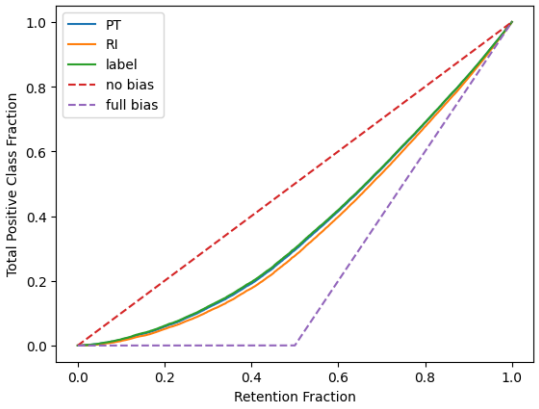
Appendix B Extra Experiments

Experiments in the main paper, Section 3.3, examine the impact of stopword bias on models' trained in-domain on IMDB data and then evaluated out-of-domain on the Twitter data. This section repeats the same set of experiments, but instead uses the Yelp dataset as in-domain and the SST-2 dataset as an out of domain test set. Table B.1 presents the performance of the BERT and RIT systems evaluated in the standard and SSW settings.

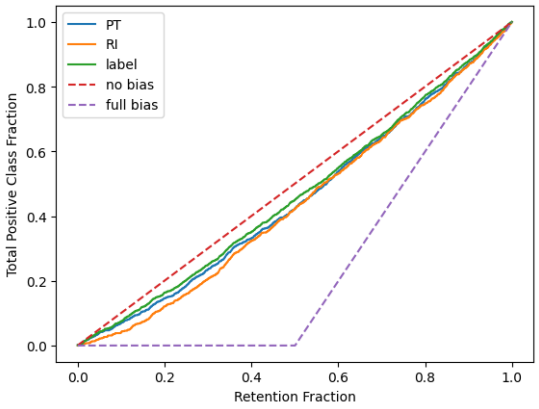
	standard		SSW	
Model	in	out	in	out
BERT	97.6	87.8	56.6	51.7
RIT	93.0	71.4	65.0	58.3

Table B.1: Accuracy (%), trained on Yelp (*standard*) and evaluated in both the *standard* and *SSW* settings, in-domain and out-of-domain (SST-2).

Next, to establish that performance degradation out of domain can be attributed to some extent to the stop word bias learnt by the models in-domain, Figure B.1 presents the retention plots for the labels and model predictions in and out of domain, using the in-domain (Yelp) stop word likelihood feature (Equation 3) to rank examples for retention (as in the main paper). As expected, the label plots show that a bias exists in-domain but this specific bias no longer holds out of domain. However, the model predictions (especially the RIT model) deviate from the unbiased label plot out of domain (Figure B.1b), demonstrating that the models are influenced by the bias they learnt on the in-domain training data.



(a) In-domain



(b) Out-of-domain

Figure B.1: Ranked by spurious stop word distribution feature retention plots for Yelp in-domain and SST out-of-domain

[6] Available at: https://huggingface.co/bert-base-uncased.