Sentence-Based Model Agnostic NLP Interpretability

Yves Rychener[1,2], **Xavier Renard**[2], **Djamé Seddah**[3], **Pascal Frossard**[1], **Marcin Detyniecki**[2,4,5]

[1]EPFL, Lausanne, Switzerland
[2]AXA, Paris, France
[3]Inria, Paris, France
[4]Sorbonne Université, Paris, France
[5]Polish Academy of Science, Warsaw, Poland

Abstract

Today, interpretability of Black-Box Natural Language Processing (NLP) models based on surrogates, like LIME or SHAP, uses word-based sampling to build the explanations. In this paper we explore the use of sentences to tackle NLP interpretability. While this choice may seem straight forward, we show that, when using complex classifiers like BERT, the word-based approach raises issues not only of computational complexity, but also of an out of distribution sampling, eventually leading to non founded explanations. By using sentences, the altered text remains in-distribution and the dimensionality of the problem is reduced for better fidelity to the black-box at comparable computational complexity.

1 Introduction

With the introduction of the transformer architecture (Vaswani et al., 2017) and its variants in recent years, we have seen a rise of complex NLP models, capable of solving more and more difficult tasks, in some cases even with superhuman capabilities (Nangia and Bowman, 2019).

The growth in complexity of the NLP models gives rise to the question of interpretability. Even if some argue that neural network architectures can be interpreted by white box approaches, which have access to model internals like gradients and activations (Arras et al., 2016; Dimopoulos et al., 1995), in practice, NLP pipelines between the input and the output of the overall model are often complex including various preprocessing methods. The final model is therefore a combination of programmatic rules and a learned classifier. In this case, a model agnostic approach which does not need access to model internals nor specific developments or continuous gradient flow is often more suitable to draw interpretable insights.

Models like LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) are examples of black-box interpreters which can be applied to texts. They create an interpretation for a text sample, called *local* interpretation. To this end, a dataset of similar texts is sampled by repeatedly removing words from the original text and observing the change in output, called the neighborhood. The local behaviour of the model is then approximated using a weighted regression on the presence of words.

In this work, we explore the limits of this approach of using words when it comes to complex language models like BERT (Devlin et al., 2019). Our main contributions are the identification of the segmentation step as a crucial, often overlooked stage in black-box NLP interpretability. In addition to displaying the problems arising from this negligence, we show that our interpreter using sentences as elementary units does not face the same problems. Finally, we achieve substantially higher performance in the benchmark problem used for assessing fidelity to the underlying classifier.

2 The Case Against Word-Based Black-Box Interpretability

While the approach of removing words is suitable for Bag-Of-Words (BOW) models without n-grams, the use of models like BERT (Devlin et al., 2019), which try to model word interactions using the attention mechanism, raises the question of knowing if this is the appropriate sampling mechanism for BERT-like models: Removing random words from a text can sometimes make it unreadable for humans, since key interactions, like verb-subject, are broken. Is this also observed with BERT? What are other consequences of word-based sampling? We compare the commonly used word based sampling to sentence-based sampling, which we argue is a more natural choice for interpretability, since sentences represent syntactically closed units and can greatly reduce the number of samples needed for explanation.

2.1 Distributional Shift

Detecting if a language model is *confused* by an input text is difficult. We can however compare the distributions of the text embedding for *normal* text and for altered text from the neighborhood generated by model-agnostic interpretability approaches. A different distribution signifies that the altered texts are not *normal*, but also raises questions on the accuracy of the language models on those texts: For neural networks, it is well studied that the Out-Of-Distribution (OOD, different distribution than training distribution) performance can be significantly worse than In-Distribution (ID, same distribution as training data) performance (Nguyen et al., 2015; Amodei et al., 2016; Hendrycks and Gimpel, 2016; Liang et al., 2017; Moosavi-Dezfooli et al., 2017), with sometimes dramatic errors known as adversarial attacks. In order for the explanation, which is based on the altered texts, to be truthful, classifier accuracy must be maintained for those samples. This can only be guaranteed if remain ID after alteration, which we will show is not the case with word-based sampling.

Inspired by Lee et al. (2018) where hidden activations were used to detect OOD samples for images, we designed the two following experiments. For both of them, we use the last embedding of the classification token as text embedding.

2.1.1 Visualizing Distributional Shift with t-SNE

In the first experiment, we compare the distribution of the embeddings of the original text, after removing a random sentence and after randomly removing the same number of words. We compute the embeddings for 10'000 randomly selected Wikipedia snippets from the SQuAD dataset (Rajpurkar et al., 2016) using BERT (Devlin et al., 2019). The distributions of the embeddings (original text, sentence removed, words removed) are compared using t-SNE visualisation. Further, we report the 1-Wasserstein distance (W_1), which is often referred to as "earth mover distance", since it measures the minimum cost (probability mass multiplied by distance moved) to turn one probability distribution into another. The results are displayed in Figure 1. One can observe that the distribution obtained by removing words (orange) is significantly different from the original one (blue), while no big difference is observed with removing sentences (green). This is confirmed by a significant decrease in Wasserstein Distance from 8.6 to 4.1 when using sentences. This observation implies that removing random words from the text results in a strongly different embedding distribution than for normal texts, while this effect is much reduced when removing a sentence. Since the classifier is trained on normal texts, its OOD accuracy on the texts obtained by word-sampling, such as used by current state of the art model-agnostic interpretability methods, is questionable, while sentence sampling produces ID texts, for which normal accuracy can be expected.

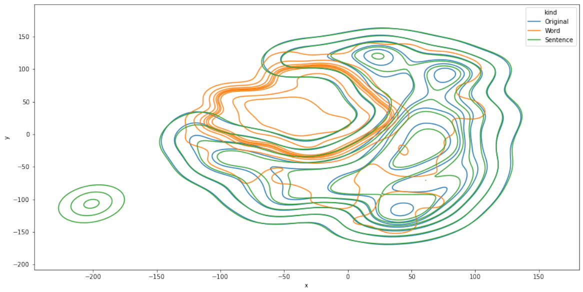

Figure 1: Distributional Shift with 10000 samples. $W_1(words) = 8.6$, $W_1(sentence) = 4.1$

2.1.2 Evaluating Distributional Shift with Classifier Accuracy

One may wonder if the observed effect is only because a relatively high number of words was removed, reflecting a strong alteration of the text. This is why we perform a second experiment, removing less than 1 word per sentence on average. We frame the detection of distributional shift as a classification problem. If no distributional shift was present, a classifier would not be able to distinguish between the two distributions, resulting in low accuracy.

We compare the embeddings of the original texts, with 5 words removed and with 1 sentence removed. In order to further study if the distributional shift effect is present across different pretraining schemes and prevails after distillation, we use a range of language models other than BERT (Devlin et al., 2019), namely DistilBERT (Sanh et al., 2019), ROBERTA (Liu et al., 2019) and ELECTRA (Clark et al., 2020), where DistilBERT is a distilled version of BERT, while ROBERTA and ELECTRA use different pretraining tasks, notably loosing next sentence prediction. Further, we employ a variety of different text domains by using context from SQuAD 2.0 (Rajpurkar et al., 2018) and SQuADShifts (Miller et al., 2020). For each binary classification (Original-Word and Original-Sentence), we train a Random Forest Classifier on the embeddings and observe its performance on a held-out test set. The results are given in Figure 2. Since the binary classifications are balanced, random predictions would yield a classification accuracy of 0.5. We observe that for all datasets and Language Models, the classification accuracy for sentence-removal is much lower compared to word-removal, almost down to random prediction. This suggests that the distributional shift with sentence-removal is much lower, confirming the results from Section 2.1.1 and suggesting that distributional shift is a problem across text domains and transformer-based Language Models. Further, using sentences seems to successfully address the issue for most language models except ROBERTA, where the altered text seems to still be OOD, although much improved. While this behaviour of the different language models is an interesting property, we leave its analysis for further works, since for the arguments presented here it suffices to note that sentence based interpretability shows preferable distributional properties.

2.2 Computational Complexity

In addition to distributional shift, computational complexity is an important problem when using language models, which often require substantial computation power for inference. Removing words or sentences as atomic units can be seen as sampling a binary vector encoding the presence/absence of words or sentences. The dimension of this binary vector is the number of atomic units. The neighborhood from which points for explanation are sampled is thus of size $2^{n_{units}}$. Obviously, the number of units n_{units} is much smaller for sentences than for words, since sentences are comprised of multiple words. Take for example the contexts from the SQuAD 2.0 (Rajpurkar et al., 2018) dataset: They contain on average 137.7 words in 5.1 sentences. The number of elements in the neighborhood are thus $2^{127.7} = 2.8 * 10^{41}$ and $2^{5.1} = 34.3$ for word-based and sentence-based alteration respectively. Since the number of samples are limited by the available computation power, the smaller sentence based neighborhood can be much better explored,

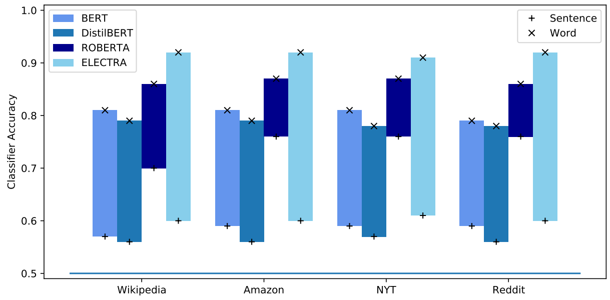

Figure 2: Comparing Distributional Shift Classifier Accuracy

resulting in a better estimation of the model's decision surface.

In the next section, we use sentence-based sampling, which we showed to be less affected by the problems of distributioinal shift and computational complexity, to propose the GUTEK framework for NLP interpretability.

3 GUTEK

Traditional post-hoc model agnostic black-box interpretability follows the same scheme, independent of input data type: For a sample to explain, we create a dataset of the local neighborhood by repeatedly perturbing parts of the input. Based on the insights from Section 2, we choose sentences as atomic units for explanations. In addition to sentence-based alteration, we employ a different methodology to select which parts to alter. For tabular data, Laugel et al. (2018) conclude that defining locality is a crucial issue for local black-box interpretability. We think that the same holds for text classification: Texts should be sampled such that small changes are more frequent than large changes. This is why we enumerate the most *local* neighborhood possible, that is the neighborhood with the fewest sentences removed. Since the dataset of the neighborhood reflects a truly local behaviour, sample weighting in the regression is not necessary.

We thus formulate the GUTEK[1] framework as a three stage approach:

1. **Segmentation**: Split the input text into sentences.

2. **Local Sampling/Dataset Generation**: Repeatedly remove some sentences in order to

[1] GUTEK stands for **G**enerating **U**nderstandable **T**ext **E**xplanations based on **K**ey segments. It is also a diminutive for "Gutenberg" in Polish.

create a dataset reflecting the local neighborhood.

3. **Modelling for Explanation**: Fit a linear regression to the dataset created in Step 2.

We will evaluate if the theoretical advancements of sentence-based sampling materialize in better explanations in the next section.

4 Experiments and Analysis

In Section 2 we point out the main reasons for proposing sentence-based interpretability: computational complexity and distributional shift. While we give theoretical arguments why these are important drawbacks of word-based methods to address, we ultimately want to give *better* explanations. Defining what a *good* explanation constitutes is still an open question in interpretability research, but we identify 2 main properties a *good* explanation should possess: Firstly, we want high **fidelity**. This means that the given explanation well reflects the reasoning of the underlying classifier. Secondly, we want the explanations to be **understandable** to humans. The explanation should be simple in order for humans to understand and reason based on.

4.1 Fidelity Evaluation with QUACKIE

In order to assess if GUTEK correctly explains the classifier's reasoning, we test if it is able to detect which parts of the text were important for the prediction. We use QUACKIE (Rychener et al., 2020), which uses a specific classification task for which interpretability ground-truths arise directly without additional human annotation. We compare our approach to LIME, which represents the current best-performing Black-Box method in the benchmark. We use LIME with *sum* aggregation, since it outperforms *max* aggregation in the primary metrics IoU and HPD, representing performance of *correctly identifying the important sentence* and *highly ranking the important sentence* respectively. We report the results for the SQuAD 2.0 dataset in Table 1, other results are given in Appendix D but show the same behaviour. We outperform the previous method by a substantial margin in both IoU and HPD for both classifiers. Notably in IoU, our approaches scores are more than double LIME's scores with the same number of samples, which implies that we find the most important sentence twice as often as the word-based approach. Only in the SNR metric, representing *selectiveness*, LIME is outperforming GUTEK. When allowing LIME 10 times as many samples as our approach (100 samples vs. 10 samples for GUTEK) it gets closer our performance without matching it. Obviously, drawing 10 times as many samples also results in a 10 fold increase in required computation power and thus a roughly 10 fold increase in computation time. Using 100 samples with the sentence-based approach results in a minor improvement of about 3 percentage points in the primary metrics IoU and HPD, suggesting that the neighborhood is already sufficiently well explored with 10 samples. Overall, the explanations from the sentence-based approach thus better represent the model's reasoning.

Method	Classif IoU	Classif HPD	Classif SNR	QA IoU	QA HPD	QA SNR
GUTEK 10	**88.55**	**90.75**	**39.48**	**90.53**	**92.37**	37.37
LIME 10	37.70	50.29	39.23	38.47	50.83	38.20
LIME 100	58.04	66.50	39.30	69.90	75.98	**40.91**

Table 1: Results on QUACKIE (SQuAD)

4.2 Qualitative Evaluation of Understandability

We resort to a simple qualitative evaluation by the authors to assess the understandability of the results. It suggests that sentence-based explanations are easier to understand, as they include context. It seems that the provided context in the sentence allows for reasoning based on the explanation. For example, by giving the whole sentence in a movie review, we know **what** was the worst (the villain), **what** the movie needed much more of (energy and spunk) and **which** role was poorly played out (phantom). The explanations can be used beyond debugging and troubleshooting for reasoning and understanding. The specific Example is given in Figure 3 in Appendix C, which also contains further examples.

5 Related Work and Discussion

Our post-hoc interpretability approach allows to explain any existing classifier. This is contrasted by the numerous developments in inherently interpretable classification models based on rationale extraction, for example by Lei et al. (2016), Jain et al. (2020) and Chang et al. (2019), which work in two steps: *Extract a Rationale* and *Make the prediction based on the Rationale*. While this approach can be argued to be inherently interpretable (the classifier only has access to the rationale), we are not aware of this method being applicable to all possible, potentially pretrained classifiers, while the black-box approach allows to use any classifier, thus potentially harnessing the most powerful classifiers for prediction.

Compared to other, word-based black-box post-hoc NLP interpretability methods like LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017), we have a much smaller search space (Section 2.2). Further, our experiments suggest that no distributional shift occurs when removing sentences (Section 2.1). We hypothesize that these are the two reasons contributing to improved fidelity at comparable computational complexity (Section 4.1) compared to LIME and SHAP.

In addition to better fidelity, the explanations are easier to understand (Section 4.2). The context provided with the explanation further opens the door for knowledge discovery, for example by comparing the topics of positive and negative sentences in reviews.

6 Conclusion

In this work, we illustrated limits of current state of the art model-agnostic interpretability methods based on word sampling (*e.g.* LIME, SHAP) when it comes to more complex classifiers like BERT. We showed that using sentences as elementary units can address the problems arising with computational complexity and distributional shift, resulting in better fidelity and human understandability. One could also imagine using other, more sophisticated methods to extract atomic units for explanation. We leave this analysis into segmentation methods for interpretability for further work. Further, we hope that this work sparks attention to the importance of the perturbation and sampling of any agnostic post-hoc interpretability method.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Leila Arras, Franziska Horn, and Grégoire Montavon. 2016. Explaining predictions of non-linear classifiers in nlp. *ACL 2016*, page 1.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2019. A game theoretic approach to class-wise selective rationalization. In *Advances in Neural Information Processing Systems*, pages 10055–10065.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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.

Yannis Dimopoulos, Paul Bourret, and Sovan Lek. 1995. Use of some sensitivity criteria for choosing networks with good generalization ability. *Neural Processing Letters*, 2(6):1–4.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C Wallace. 2020. Learning to faithfully rationalize by construction. *arXiv preprint arXiv:2005.00115*.

Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. 2018. Defining locality for surrogates in post-hoc interpretablity. *arXiv preprint arXiv:1806.07498*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. *arXiv preprint arXiv:2004.14444*.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773.

Nikita Nangia and Samuel Bowman. 2019. Human vs. muppet: A conservative estimate of human performance on the glue benchmark. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4566–4575.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Yves Rychener, Xavier Renard, Djamé Seddah, Pascal Frossard, and Xavier Detyniecki. 2020. Quackie: A nlp classification task with ground truth explanations.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

A Reproducibility

To ensure reproducibility, we give the implementation details of our experiments. Direct implementations can also be found directly on our Github[2].

A.1 The Case Against Word-Based Black-Box Interpretability

A.1.1 Distributional Shift

We use the last embedding of the classification token as representation of the whole text. We use base uncased BERT (Devlin et al., 2019). For the visualisation experiment, we directly use this embedding to calculate Wasserstein distance. To visualize, we use t-SNE on the combined dataset (word removed + sentence removed + original) with PCA initialisation and a perplexity of 100. The algorithm is given a maximum of 5000 iterations, for other parameters we used SKLearn (Pedregosa et al., 2011) defaults.

For evaluating distributional shift with classifier accuracy, we use base uncased BERT (Devlin et al., 2019), base RoBERTa (Liu et al., 2019), base uncased DistilBERT (Sanh et al., 2019) and the small ELECTRA (Clark et al., 2020) discriminator. The text embeddings are pairwise used to create a classification problem, which uses a random 75-25 train test split. We train a Random Forest Classifier using default SKLearn parameters, controlling for complexity using the maximum depth with options 2, 5, 7, 10, 15 and 20. The best choice is selected using out-of-bag accuracy. Results in Figure 2 and Table 2 represents performance on the test-set.

A.1.2 Computational Complexity

In order to have normal flowing text, we use text from Wikipedia, notably contexts from SQuAD 2.0 (Rajpurkar et al., 2018). We compare the number of sentences and the number of words, obtained using NLTK (Bird et al., 2009) *sent_tokenize* and *word_tokenize* respectively.

A.2 Experiments and Analysis

A.2.1 Fidelity Evaluation with QUACKIE

We use code provided with QUACKIE (Rychener et al., 2020) to test GUTEK. In our implementation of GUTEK, we use NLTK *sent_tokenize* to split the text into sentences and use the SKLearn implementation of the Linear Regression as surrogate. The coefficients of the linear regression are used as sentence scores.

[2]https://github.com/axa-rev-research/gutek

DATASET	LM	WORD	SENTENCE
WIKIPEDIA	BERT	0.81	0.57
	DISTILBERT	0.79	0.56
	ROBERTA	0.86	0.70
	ELECTRA	0.92	0.60
AMAZON	BERT	0.81	0.59
	DISTILBERT	0.79	0.56
	ROBERTA	0.87	0.76
	ELECTRA	0.92	0.60
NYT	BERT	0.81	0.59
	DISTILBERT	0.78	0.57
	ROBERTA	0.87	0.76
	ELECTRA	0.91	0.61
REDDIT	BERT	0.79	0.59
	DISTILBERT	0.78	0.56
	ROBERTA	0.86	0.76
	ELECTRA	0.92	0.60

Table 2: OOD Classification Results in Tabular Form

A.2.2 Human Understandability

As underlying classifier, we use a combination of TF-IDF and Random Forest Classifier. For TF-IDF, we use English stopwords, an n-gram range from 1 to 3, no maximum number of features, maximum document frequency of 0.8, minimum token occurrence of 4, using L1 normalization and sublinear term-frequency. Random Forest uses default parameters. Both implementations are by SKLearn. We use the standard LIME (Ribeiro et al., 2016) package and our implementation of GUTEK described in Appendix A.2.1. For both interpreters, probability outputs are used.

B Tabular Results for OOD Classification

In addition to plotting, we give the results from Figure 2 in Table 2.

C Further Examples for Qualitative Evaluation

In addition to the example from Figure 3, we provide further examples for qualitative evaluation in Figures 4, 5 and 6. Note that we manually tweaked the text size and figure size in order for the text to not overlap in visualisation. Also, no post-processing (like filtering important parts only) was done after GUTEK.

D Complete QUACKIE results

We also give results for all datasets in QUACKIE and report the scores for all other methods currently in QUACKIE in Tables 3, 4 and 5.

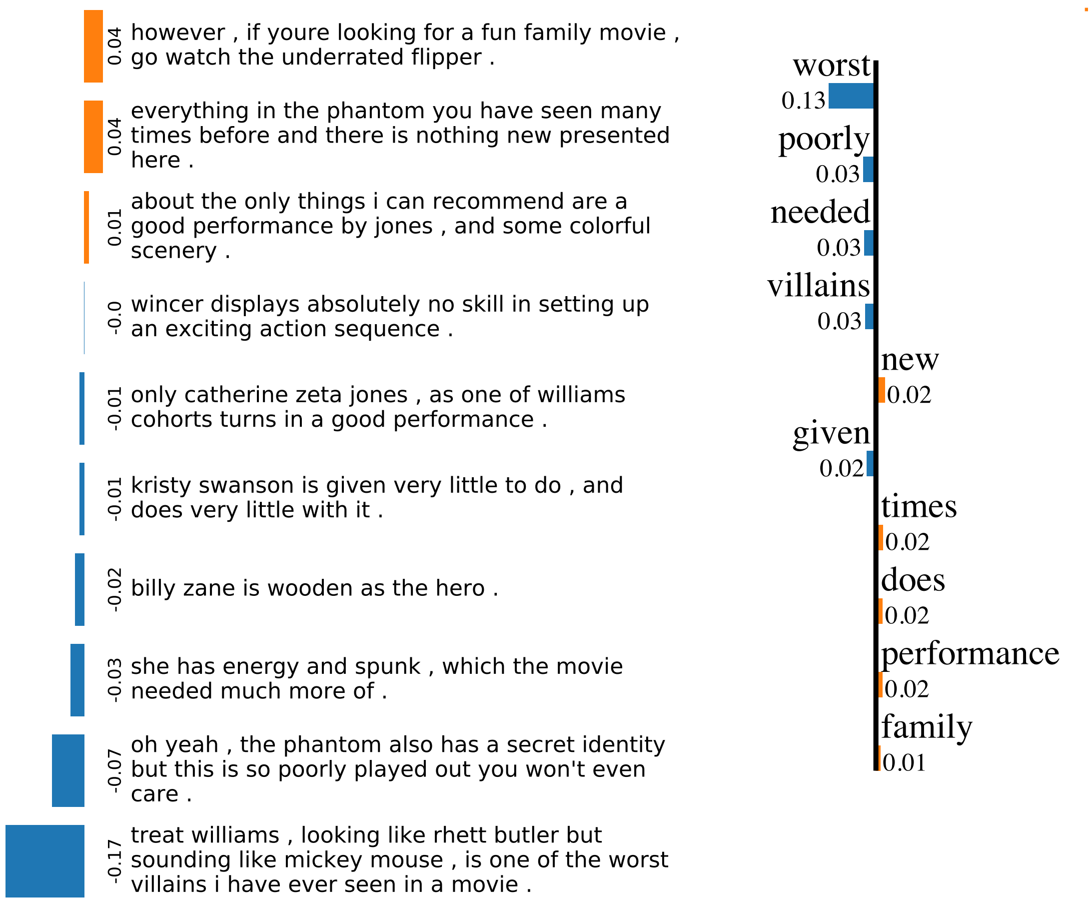

Figure 3: Comparison of Explanations for TFIDF movie sentiment classifier, GUTEK (left) vs LIME (right) (sample id 332)

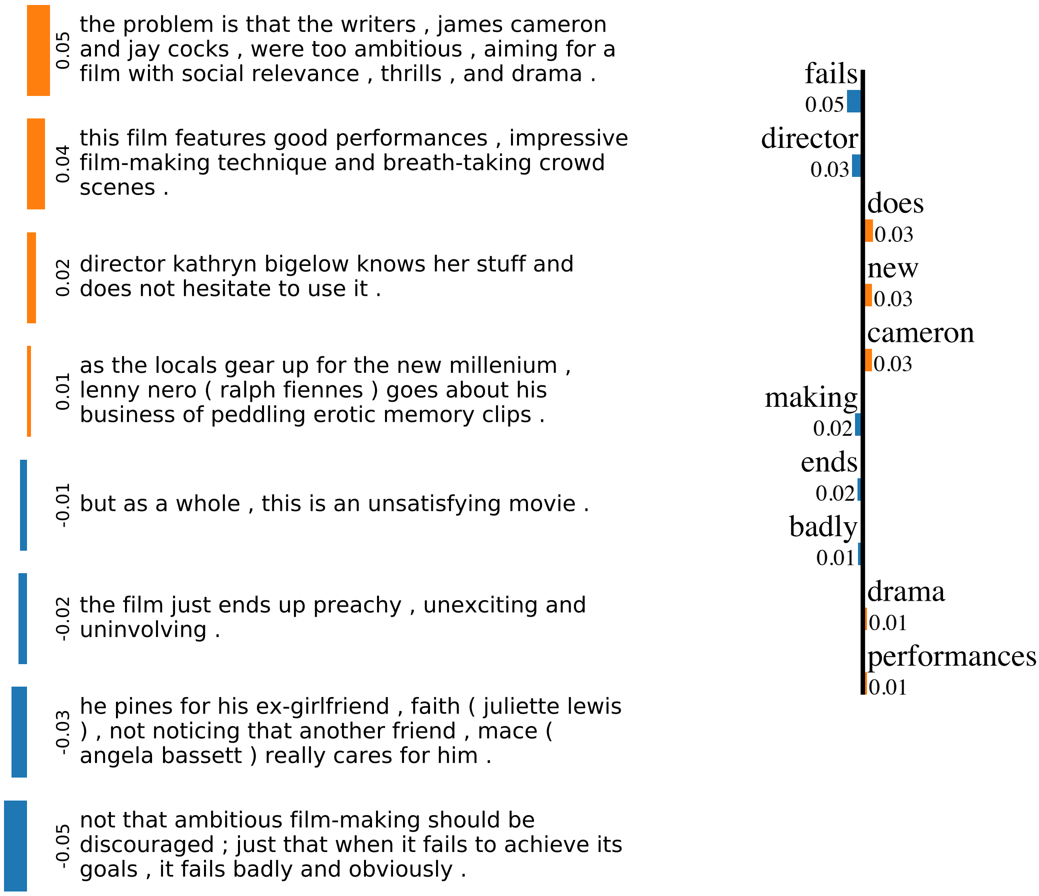

Figure 4: Comparison of Explanations for TFIDF movie sentiment classifier, GUTEK (left) vs LIME (right) (sample id 195)

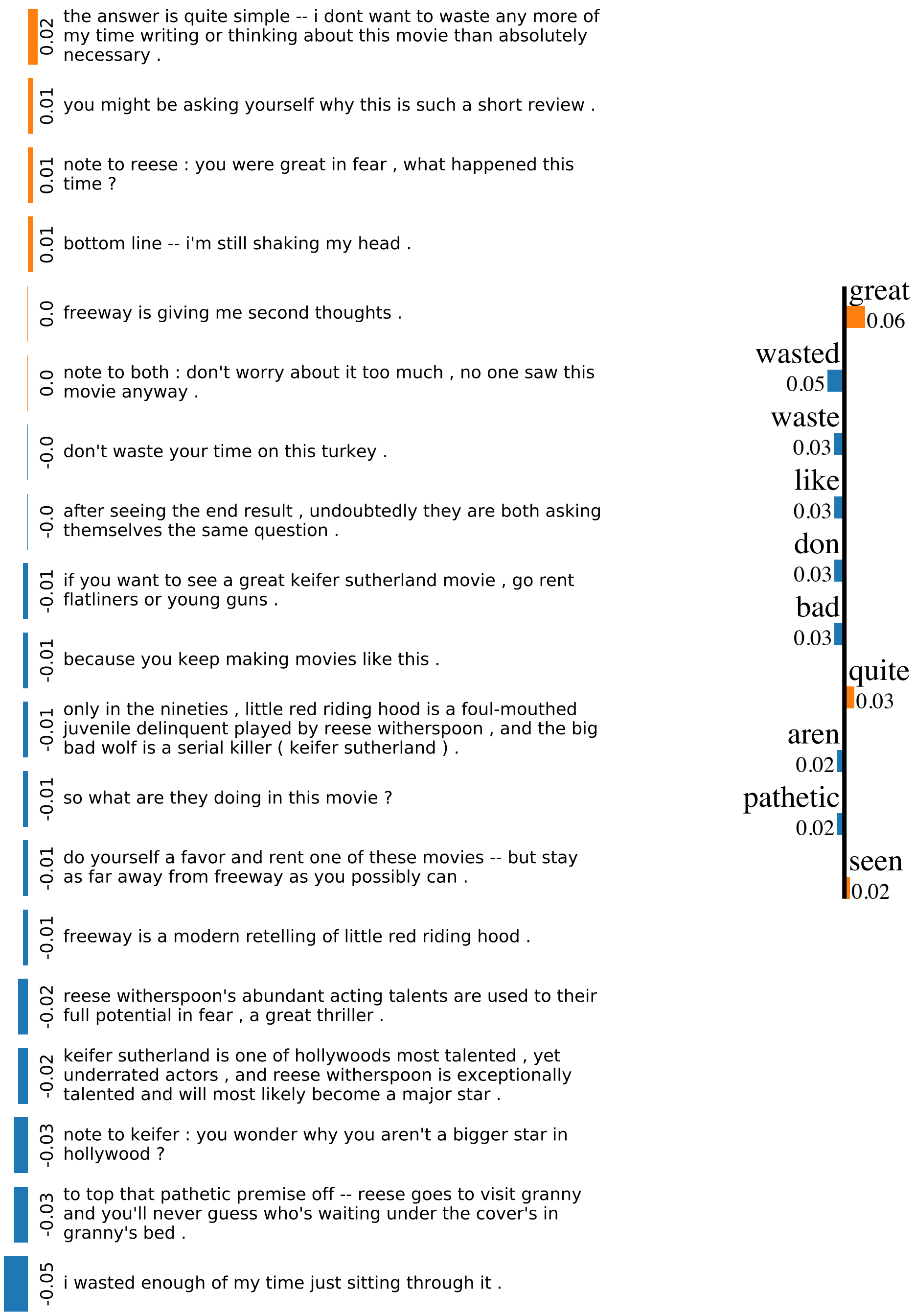

Figure 5: Comparison of Explanations for TFIDF movie sentiment classifier, GUTEK (left) vs LIME (right) (sample id 370)

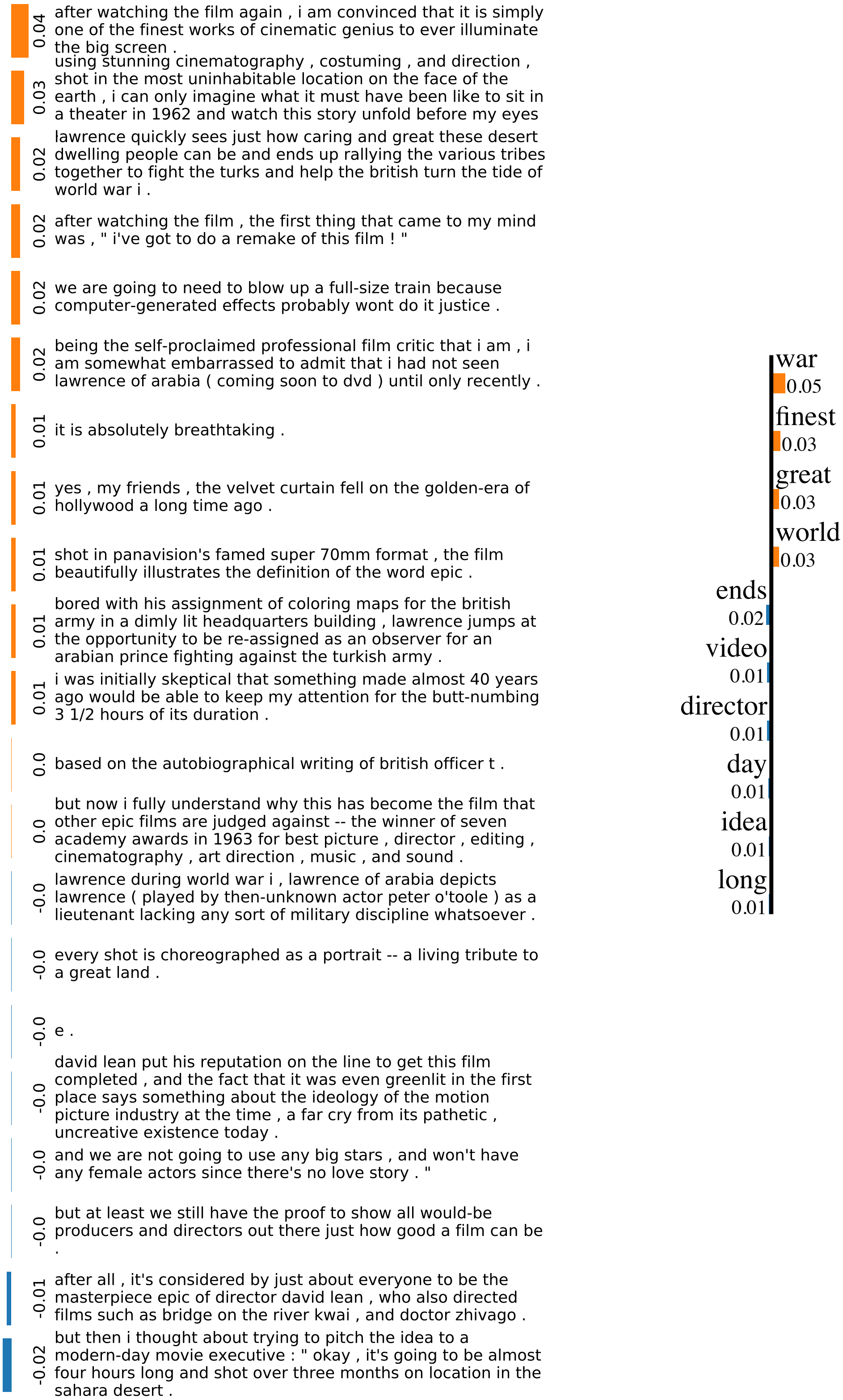

Figure 6: Comparison of Explanations for TFIDF movie sentiment classifier, GUTEK (left) vs LIME (right) (sample id 70)

Interpreter	Aggregation	Samples	SQuAD Classif	SQuAD QA	New Wiki Classif	New Wiki QA	NYT Classif	NYT QA	Reddit Classif	Reddit QA	Amazon Classif	Amazon QA
GUTEK	-	10	88.55	90.53	87.7	89.54	87.66	88.04	71.62	76.86	79.09	78.95
		100	91.38	90.53	90.83	90.53	91.56	91.84	80.62	86.98	84.77	86.14
LIME†	sum	10	37.70	38.47	40.72	41.40	40.22	41.98	31.35	35.34	32.69	35.27
		100	58.04	69.90	60.74	70.82	62.02	73.50	50.33	69.02	53.57	67.58
	max	10	34.06	35.36	36.98	37.77	36.19	36.43	26.52	28.89	29.80	32.12
		100	57.86	68.30	59.65	68.43	61.57	72.00	48.22	65.23	53.09	66.24
SHAP†	sum	10	30.48	32.90	31.66	32.84	29.26	31.03	22.13	23.75	24.59	25.43
		100	54.85	65.92	57.53	65.79	56.38	67.70	49.35	65.02	54.03	67.68
	max	10	29.69	30.81	30.72	31.68	28.32	30.00	21.17	22.58	22.72	23.84
		100	52.45	62.34	54.56	63.18	53.19	64.78	45.79	60.03	49.54	63.35
Saliency	sum	-	74.74	91.12	72.19	91.18	68.87	88.46	57.57	85.26	64.82	85.91
	max	-	66.27	80.79	65.04	80.78	58.95	76.07	48.41	77.33	59.52	79.70
Integrated Gradients	sum	50	66.73	85.93	65.00	85.93	65.44	85.20	51.62	79.73	51.96	78.21
	max	50	62.73	87.05	60.70	86.73	61.63	85.92	50.24	82.35	49.09	82.45
SmoothGrad	sum	5	60.98	91.28	60.29	90.56	60.25	88.29	50.32	84.51	52.34	84.40
	max	5	59.48	82.16	61.45	82.38	56.93	78.33	45.95	77.72	53.03	78.26
Random	-	-	24.64	25.38	26.86	27.39	24.53	24.36	16.51	16.09	18.71	19.17

Table 3: IoU Results

Interpreter	Aggregation	Samples	SQuAD Classif	SQuAD QA	New Wiki Classif	New Wiki QA	NYT Classif	NYT QA	Reddit Classif	Reddit QA	Amazon Classif	Amazon QA
GUTEK	-	10	90.75	92.37	90.17	91.68	89.64	90.02	74.93	79.5	81.76	81.71
		100	93.02	92.37	92.72	92.37	93.02	93.36	83.04	88.66	86.84	88.17
LIME†	sum	10	50.29	50.83	53.32	53.76	51.62	53.12	39.99	43.56	42.31	44.64
		100	66.50	75.98	68.93	76.93	69.17	78.58	56.60	72.97	60.12	72.25
	max	10	45.12	46.19	47.85	48.62	46.60	47.11	34.47	36.74	38.43	40.63
		100	63.74	71.33	65.47	71.89	67.28	75.25	53.23	67.65	58.41	69.21
SHAP†	sum	10	41.22	44.09	42.87	44.57	39.06	41.38	28.94	31.26	32.97	34.63
		100	63.93	72.75	66.18	72.91	64.39	73.74	55.59	69.44	60.48	72.29
	max	10	37.74	39.28	39.29	40.68	36.40	38.54	27.30	29.13	30.24	32.05
		100	59.85	67.47	61.80	68.35	60.80	69.98	51.19	63.41	55.32	66.86
Saliency	sum	-	79.91	93.01	78.20	93.06	74.97	90.71	63.05	87.21	69.96	88.01
	max	-	72.99	84.83	72.40	84.81	66.86	80.94	55.10	80.32	65.31	82.74
Integrated Gradients	sum	50	73.52	88.85	72.39	88.93	71.99	88.00	57.84	82.46	58.93	81.56
	max	50	70.15	89.67	68.93	89.48	68.73	88.52	56.51	84.70	56.35	85.10
SmoothGrad	sum	5	69.03	93.08	68.92	92.59	68.05	90.52	56.77	86.58	59.36	86.72
	max	5	67.83	85.77	69.64	86.12	65.32	82.65	53.00	80.64	59.84	81.43
Random	-	-	40.28	40.71	42.66	42.92	39.23	39.18	27.41	27.17	30.79	31.34

Table 4: HPD Results

Interpreter	Aggregation	Samples	SQuAD Classif	SQuAD QA	New Wiki Classif	New Wiki QA	NYT Classif	NYT QA	Reddit Classif	Reddit QA	Amazon Classif	Amazon QA
GUTEK	-	10	39.48	37.37	42.63	37.86	32.21	30.68	19.12	17.69	26.11	22.22
		100	35.49	37.37	39.5	37.37	30.38	33.13	18.22	19.48	20.9	22.92
LIME	sum	10	39.23	38.20	41.82	40.66	36.98	37.26	25.89	22.84	27.87	27.08
		100	39.30	40.91	42.30	43.96	39.41	39.38	32.90	46.71	27.42	32.52
	max	10	91.76	91.54	94.38	88.83	93.24	85.01	107.89	110.55	93.77	95.83
		100	125.98	176.07	124.96	162.51	133.54	184.66	171.42	305.21	151.52	232.94
SHAP	sum	10	73.24	67.24	74.17	67.85	71.42	68.34	91.28	83.95	68.02	60.68
		100	42.27	42.09	44.80	45.60	37.31	43.69	34.30	40.05	28.76	34.64
	max	10	99.16	102.31	97.42	101.44	97.10	92.66	130.98	127.01	99.87	102.16
		100	107.51	137.77	102.01	132.57	94.43	132.64	149.95	240.47	135.62	207.62
Saliency	sum	-	37.29	39.92	40.88	40.14	35.14	34.23	19.10	19.77	22.90	23.34
	max	-	35.58	38.20	39.75	39.81	34.08	36.35	19.15	20.04	22.17	23.78
Integrated Gradients	sum	50	37.28	37.32	39.16	40.19	33.30	33.04	18.96	20.60	22.50	24.60
	max	50	35.71	34.99	38.74	38.09	32.55	32.80	18.41	20.57	21.80	23.69
SmoothGrad	sum	5	38.13	37.22	41.29	40.15	35.16	33.04	19.40	20.33	23.34	23.11
	max	5	37.55	36.61	40.29	40.04	34.85	35.98	19.23	19.62	22.45	22.69
Random	-	-	37.70	37.34	40.63	40.52	34.87	35.06	19.24	19.79	23.21	23.70

Table 5: SNR Results (Examples for which noise cannot be estimated are omitted)