Democratizing Advanced Attribution Analyses of Generative Language Models with the Inseq Toolkit

Gabriele Sarti[1,*], Nils Feldhus[2], Jirui Qi[1], Malvina Nissim[1] and Arianna Bisazza[1]

[1] *Center for Language and Cognition (CLCG), University of Groningen, Oude Kijk in 't Jatstraat 26*
Groningen, 9712EK, The Netherlands

[2] *German Research Center for Artificial Intelligence (DFKI), Alt-Moabit 91c, Berlin, 10559, Germany*

Abstract

Inseq[1] is a recent toolkit providing an intuitive and optimized interface to conduct feature attribution analyses of generative language models. In this work, we present the latest improvements to the library, including efforts to simplify the attribution of large language models on consumer hardware, additional attribution approaches, and a new client command to detect and attribute context usage in language model generations. We showcase an online demo using Inseq as an attribution backbone for context reliance analysis, and we highlight interesting contextual patterns in language model generations. Ultimately, this release furthers Inseq's mission of centralizing good interpretability practices and enabling fair and reproducible model evaluations.

Keywords

Natural Language Processing, Generative Language Models, Feature Attribution, Python Toolkit

1. Introduction

Feature attribution methods have been widely adopted in NLP to quantify the importance of input tokens in driving language models' (LMs) predictions [1]. While some works used feature attribution to analyze generative NLP models, focusing mainly on machine translation [2, 3, 4, i.a.], most analyses in this area focused on classification due to the initial popularity of BERT-based encoders [5] and the challenges of autoregressive generation [6]. Although several post-hoc interpretability tools are available, few support generative LMs [7, 8, 9], often requiring ad-hoc wrappers to enable interoperability with the popular Transformers library [10] commonly used by NLP practitioners.

Inseq [11] is a Python library offering native compatibility with Transformers and supporting advanced methods and customizations. Inseq centralizes access to a broad set of feature attribution methods, sourced in part from the Captum [12] framework, enabling fair comparisons

[1] Library: https://github.com/inseq-team/inseq, Docs: https://inseq.org. This paper refers to release `v0.6.0`.

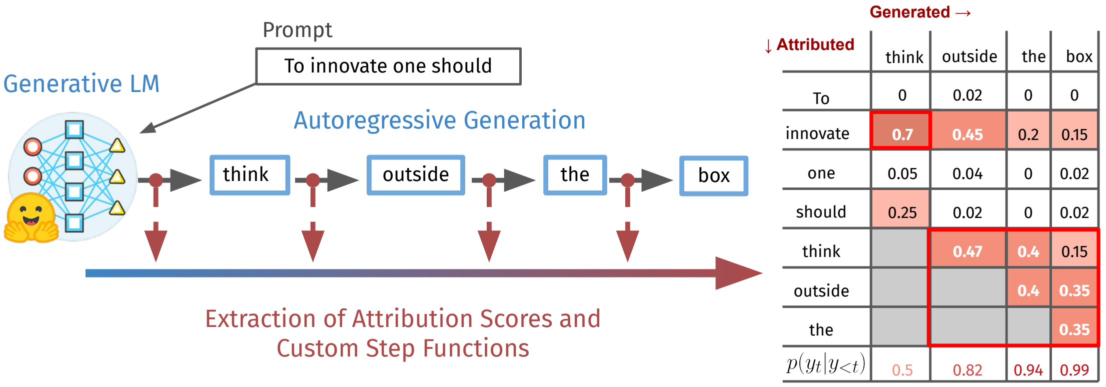
Late-breaking work, Demos and Doctoral Consortium, colocated with The 2nd World Conference on eXplainable Artificial Intelligence: July 17–19, 2024, Valletta, Malta

[*] Corresponding author.

g.sarti@rug.nl (G. Sarti); nils.feldhus@dfki.de (N. Feldhus); j.qi@rug.nl (J. Qi); m.nissim@rug.nl (M. Nissim); a.bisazza@rug.nl (A. Bisazza)

https://gsarti.com (G. Sarti); https://nfelnlp.github.io (N. Feldhus); https://betswish.github.io (J. Qi); https://cs.rug.nl/~bisazza (A. Bisazza)

0000-0001-8715-2987 (G. Sarti); 0009-0009-0608-8203 (J. Qi); 0000-0001-5289-0971 (M. Nissim); 0000-0003-1270-3048 (A. Bisazza)


Prompt
To innovate one should
Generative LM
Autoregressive Generation
think
outside
the
box
Extraction of Attribution Scores and Custom Step Functions
Generated →
↓ Attributed

↓ Attributed	think	outside	the	box
To	0	0.02	0	0
innovate	0.7	0.45	0.2	0.15
one	0.05	0.04	0	0.02
should	0.25	0.02	0	0.02
think		0.47	0.4	0.15
outside			0.4	0.35
the				0.35
$p(y_t \vert y_{<t})$	0.5	0.82	0.94	0.99

Figure 1: Example of Inseq usage with a generative LM. Given a prompt, attribution scores and next-step probabilities are extracted from the model at every generation step, with a final visualization aggregating values at the token level. Highlighted areas in the output show that the model relies on the keyword "innovate" to begin the idiomatic expression "think outside the box" at relatively low confidence (`p = 0.5`). However, importance shifts to previous tokens in the idiom throughout the generation.

across various techniques for all encoder-decoder and decoder-only models supported by the Transformers library. The toolkit aims to democratize access to interpretability analyses of generative LMs with minimal setup, enabling reproducible evaluations. An example is provided in Figure 1. Thanks to its intuitive interface, users can easily integrate interpretability analyses into their text generation pipelines with just a few lines of code. Moreover, a command-line interface (CLI) and various utility methods to visualize, serialize, and reload attribution outcomes are provided to facilitate analysis at scale. Inseq is also highly flexible, including cutting-edge attribution methods with built-in post-processing features (Section 2), supporting customizable attribution targets and enabling the attribution of arbitrary sequences produced via forced decoding (Section 2.1).

In this paper, we summarize recent efforts in the development of the Inseq toolkit, focusing specifically on newly added usability features to support the attribution of large LMs (LLMs) (Section 2.2), and a new command to contrastively attribute context usage in LMs generations (Section 3). Finally, we present various applications of Inseq in recent research (Section 4).

2. The Inseq Toolkit

Inseq provides an easy-to-use interface to apply feature attribution methods, extending Captum [12] as attribution back-end to generative models from the Transformers library [10].

Table 1 (left) presents an updated list of supported attribution methods, categorized into three groups, *gradient-based*, *internals-based* and *perturbation-based*, depending on their underlying approach to importance quantification. Aside from popular model-agnostic methods, Inseq notably provides built-in support for attention weight attribution and a range of cutting-edge methods not supported in any other toolkit, such as Discretized Integrated Gradients [17], Sequential Integrated Gradients [18], Value Zeroing [22], and ReAGent [23], with many of those allowing for the importance attribution of custom intermediate model layers.

Among its notable features, Inseq offers flexible **source and target-side attribution** for

	Method	Source	$f(l)$
G	(Input $\times$) Gradient	Simonyan et al.	yes
	DeepLIFT	Shrikumar et al.	yes
	GradientSHAP	Lundberg and Lee	no
	Integrated Gradients	Sundararajan et al.	yes
	Discretized IG	Sanyal and Ren	no
	Sequential IG	Enguehard	no
I	Attention Weights	Bahdanau et al.	yes
P	Occlusion (Blank-out)	Zeiler and Fergus	no
	LIME	Ribeiro et al.	no
	Value Zeroing	Mohebbi et al.	yes
	ReAGent	Zhao and Shan	no

	Method	Source
S	(Log) Probability	-
	Softmax Entropy	-
	Target Cross-entropy	-
	Perplexity	-
	Contrastive logits Δ	Yin and Neubig
	Contrastive prob. Δ	
	μ MC Dropout Prob.	Gal and Ghahramani
	P-CXMI	Fernandes et al.
	KL divergence	-
	In-context P$\mathcal{V}$I	Lu et al.
	Top-p tokens	-

Table 1
Gradient (**G**), internals (**I**) and perturbation-based (**P**) attribution methods and built-in step functions (**S**) in Inseq. $f(l)$: supports intermediate layers attribution. New methods are **bolded**.

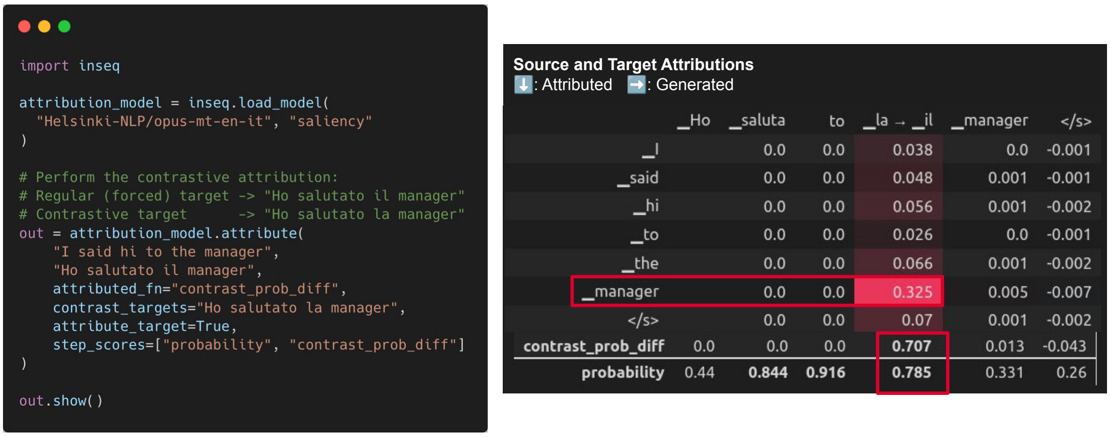
encoder-decoder systems, alongside several `Aggregator` classes to aggregate attribution scores across various dimensions (e.g. at the token level), and `AggregatorPipeline` for chaining various aggregation steps (e.g. extract the weight of the i-th attention head at the n-th layer).

2.1. Customizing generation and attribution

At every generation step, in addition to computing attribution scores, Inseq can also use models' information to compute functions of the output distributions or intermediate representations, which we collectively refer to as **step functions** (Table 1, **S**). For example, the resulting scores can provide additional insights into the generation process for uncertainty quantification or outlier detection. Inseq provides access to several built-in step functions and allows users to create and register custom ones. Step scores are computed alongside attribution and visualized in the same matrix of attribution scores (e.g. $p(y_t|y_{<t})$ in Figure 1).

Various attribution methods rely on model outputs to predict input importance, using functions of the model's output logits or token probabilities [27]. Yin and Neubig [6] propose contrastive metrics to help disentangle how various factors contribute to the prediction. For example, the gradient $\nabla(p(\text{barking}) - p(\text{crying}))$ given the prompt *"Can you stop the dog from ___"* will highlight the role of the entity *dog* in selecting *barking*, disentangling the semantic component from grammatical correctness by providing a *crying* as grammatically valid choice. Figure 2 illustrates an example. Inseq users can leverage any built-in or custom-defined step function as an **attribution target**, enabling advanced use cases like contrastive comparisons.

The new version of Inseq supports customizable **word alignments**, i.e. indices aligning tokens in the original and contrastive generated texts, to support contrastive comparisons between texts of different lengths, including automatic alignments using the multilingual LaBSE encoder [28] to streamline their application.



```python
import inseq

attribution_model = inseq.load_model(
  "Helsinki-NLP/opus-mt-en-it", "saliency"
)

# Perform the contrastive attribution:
# Regular (forced) target -> "Ho salutato il manager"
# Contrastive target      -> "Ho salutato la manager"
out = attribution_model.attribute(
    "I said hi to the manager",
    "Ho salutato il manager",
    attributed_fn="contrast_prob_diff",
    contrast_targets="Ho salutato la manager",
    attribute_target=True,
    step_scores=["probability", "contrast_prob_diff"]
)

out.show()
```

Source and Target Attributions
⬇️: Attributed ➡️: Generated

	_Ho	_saluta	to	_la → _il	_manager	</s>
_I		0.0	0.0	0.038	0.0	-0.001
_said		0.0	0.0	0.048	0.001	-0.001
_hi		0.0	0.0	0.056	0.001	-0.002
_to		0.0	0.0	0.026	0.0	-0.001
_the		0.0	0.0	0.066	0.001	-0.002
_manager		0.0	0.0	0.325	0.005	-0.007
</s>		0.0	0.0	0.07	0.001	-0.002
contrast_prob_diff	0.0	0.0	0.0	**0.707**	0.013	-0.043
probability	0.44	**0.844**	**0.916**	**0.785**	0.331	0.26

Figure 2: Left: Code using Inseq to compute contrastive attributions for an English-to-Italian machine translation model using raw gradient attribution. **Right:** Source-to-target attributions aggregated at token-level, indicating the importance of the stereotypical noun "manager" to generate the Italian masculine pronoun "il" (original) over the feminine "la" (contrastive case).

2.2. Usability Features

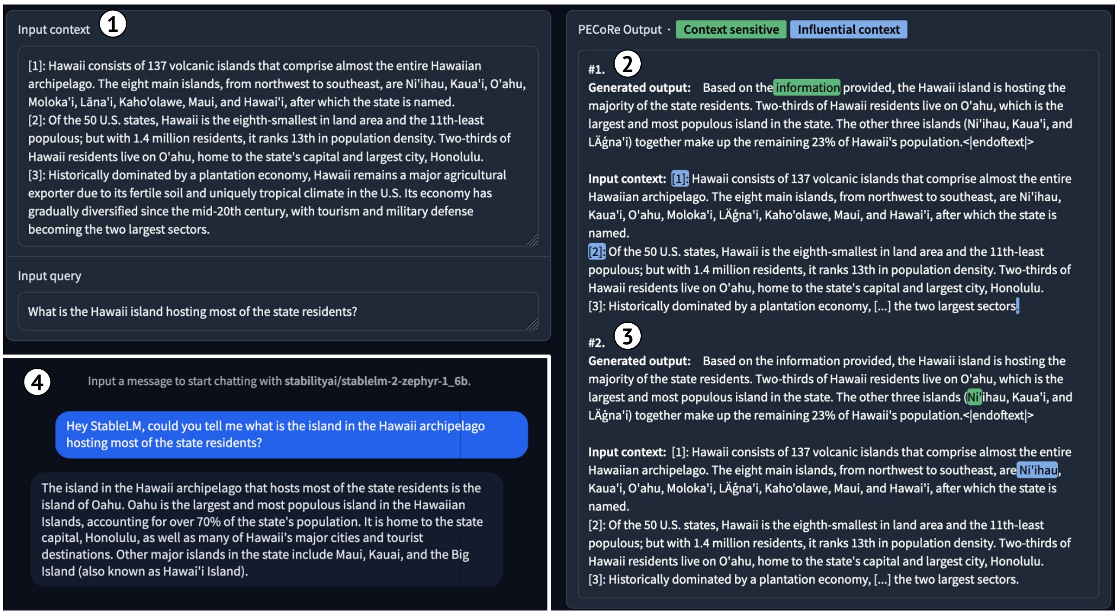
Inseq supports batching to simplify analysis at scale and customizable start/end positions to accelerate the attribution process for studies on localized phenomena (e.g., pronoun coreference). Moreover, it offers a CLI to attribute single examples or entire Datasets from the command line, storing resulting outputs and visualizations. Attributions can be saved in JSON format with metadata to identify their provenance, allowing for easy reloading and visualization.

Quantization and distributed attribution All models allowing for **quantization** using `bitsandbytes` [29] can be loaded in 4-bit or 8-bit precision directly from Transformers, and their attributions can be computed normally using Inseq at a fraction of the cost. Similarly, Inseq is compatible with the Petals library [30], supporting gradient-based attribution across language models whose computation is distributed across several machines. This can alleviate the need for high-end GPUs to run LLMs, enabling the distributed computation of attribution scores.[1]

3. Case study: Attributing Context Influence using PECoRe

The PECoRe framework [31] was proposed to identify and attribute context usage in language models, and further adapted by Qi et al. [32] to produce model internals-based citations for LLM generations. First, contrastive functions such as KL divergence select generated tokens sensitive to context ablation. Then, contrastive feature attribution is used to identify context tokens driving the contextual prediction. Inseq provides an ad-hoc CLI command (`attribute-context`) for PECoRe usage, supporting all contrastive step functions and attribution methods. Figure 3 provides an example output in a GUI built on top of the Inseq API.[2] In the example, an LLM[3] is

[1] A tutorial for distributed attribution is available here: https://inseq.org/en/latest/examples/petals.html
[2] The presented demo is available here: https://huggingface.co/spaces/gsarti/pecore
[3] We used StableLM 2 Zephyr 1.6B: https://huggingface.co/stabilityai/stablelm-2-zephyr-1_6b



Input context

[1]: Hawaii consists of 137 volcanic islands that comprise almost the entire Hawaiian archipelago. The eight main islands, from northwest to southeast, are Ni'ihau, Kaua'i, O'ahu, Moloka'i, Lāna'i, Kaho'olawe, Maui, and Hawai'i, after which the state is named.
[2]: Of the 50 U.S. states, Hawaii is the eighth-smallest in land area and the 11th-least populous; but with 1.4 million residents, it ranks 13th in population density. Two-thirds of Hawaii residents live on O'ahu, home to the state's capital and largest city, Honolulu.
[3]: Historically dominated by a plantation economy, Hawaii remains a major agricultural exporter due to its fertile soil and uniquely tropical climate in the U.S. Its economy has gradually diversified since the mid-20th century, with tourism and military defense becoming the two largest sectors.

Input query

What is the Hawaii island hosting most of the state residents?

Input a message to start chatting with **stabilityai/stablelm-2-zephyr-1_6b.**

Hey StableLM, could you tell me what is the island in the Hawaii archipelago hosting most of the state residents?

The island in the Hawaii archipelago that hosts most of the state residents is the island of Oahu. Oahu is the largest and most populous island in the Hawaiian Islands, accounting for over 70% of the state's population. It is home to the state capital, Honolulu, as well as many of Hawaii's major cities and tourist destinations. Other major islands in the state include Maui, Kauai, and the Big Island (also known as Hawai'i Island).

PECoRe Output · Context sensitive Influential context

#1.

Generated output: Based on the information provided, the Hawaii island is hosting the majority of the state residents. Two-thirds of Hawaii residents live on O'ahu, which is the largest and most populous island in the state. The other three islands (Ni'ihau, Kaua'i, and LĂģna'i) together make up the remaining 23% of Hawaii's population.<|endoftext|>

Input context: [1]: Hawaii consists of 137 volcanic islands that comprise almost the entire Hawaiian archipelago. The eight main islands, from northwest to southeast, are Ni'ihau, Kaua'i, O'ahu, Moloka'i, LĂģna'i, Kaho'olawe, Maui, and Hawai'i, after which the state is named.
[2]: Of the 50 U.S. states, Hawaii is the eighth-smallest in land area and the 11th-least populous; but with 1.4 million residents, it ranks 13th in population density. Two-thirds of Hawaii residents live on O'ahu, home to the state's capital and largest city, Honolulu.
[3]: Historically dominated by a plantation economy, [...] the two largest sectors.

#2.

Generated output: Based on the information provided, the Hawaii island is hosting the majority of the state residents. Two-thirds of Hawaii residents live on O'ahu, which is the largest and most populous island in the state. The other three islands (Ni'ihau, Kaua'i, and LĂģna'i) together make up the remaining 23% of Hawaii's population.<|endoftext|>

Input context: [1]: Hawaii consists of 137 volcanic islands that comprise almost the entire Hawaiian archipelago. The eight main islands, from northwest to southeast, are Ni'ihau, Kaua'i, O'ahu, Moloka'i, LĂģna'i, Kaho'olawe, Maui, and Hawai'i, after which the state is named.
[2]: Of the 50 U.S. states, Hawaii is the eighth-smallest in land area and the 11th-least populous; but with 1.4 million residents, it ranks 13th in population density. Two-thirds of Hawaii residents live on O'ahu, home to the state's capital and largest city, Honolulu.
[3]: Historically dominated by a plantation economy, [...] the two largest sectors.

Figure 3: Context attribution for retrieval-augmented long-form QA using the Inseq-powered demo.

prompted with contexts retrieved from Wikipedia to provide a long-form answer to a query (1). When referring to context information (2), PECoRe shows that the indices of the two documents containing relevant information are salient. On the other hand, the names of other Hawaiian islands are important when the model produces an additional remark on their population (3). We observe that the context is not salient for answering the question, suggesting the model might have memorized the answer. We test this by prompting the model in a closed-book setting, finding that the model can indeed respond correctly without context (4).

4. Related Work using Inseq

Since its first release, Inseq was adopted to conduct several feature attribution analyses of generative LMs. In the conversational domain, its Integrated Gradients implementation was used to study longitudinal dialogues with conversational models for Italian [33]. Inseq was also used to measure agreement between attribution scores and a new metric of LLMs' factual reliability [34], and to analyze the context repetition in dialogues [35]. In machine translation, Inseq attribution methods were used to select salient in-context examples with the aim to mitigate gender bias in translated sentences [36] and to evaluate the usage of source and target-side information in character-level machine translation systems across several languages [37].

Inseq was integrated into several tools and methods, including the LLMCheckup interface [38], using Inseq for producing attributions for fact-checking and conversational question answering (QA), and the PECoRe framework [31] for detecting and attributing context usage in language models. Finally, Inseq methods were also used as baselines to compare proposed new feature attribution approaches [23], and to probe the contextual influence in affixal negation [39].

Acknowledgments

GS and AB acknowledge the support from the Dutch Research Council (NWO) for the InDeep project (NWA.1292.19.399). JQ and AB acknowledge NWO support for the Lessen project (NWA.1389.20.183). GS acknowledges support of the Netherlands eScience Center Fellowship program. NF acknowledges support from the German Federal Ministry of Education and Research (BMBF) as part of the project XAINES (01IW20005).

References

[1] A. Madsen, S. Reddy, S. Chandar, Post-hoc interpretability for neural nlp: A survey, ACM Comput. Surv. 55 (2022). doi:10.1145/3546577.

[2] D. Alvarez-Melis, T. Jaakkola, A causal framework for explaining the predictions of black-box sequence-to-sequence models, in: EMNLP 2017, 2017, pp. 412–421. doi:10.18653/v1/D17-1042.

[3] S. Ding, H. Xu, P. Koehn, Saliency-driven word alignment interpretation for neural machine translation, in: WMT 2019 (Volume 1: Research Papers), ACL, 2019, pp. 1–12. doi:10.18653/v1/W19-5201.

[4] J. Ferrando, G. I. Gállego, B. Alastruey, C. Escolano, M. R. Costa-jussà, Towards opening the black box of neural machine translation: Source and target interpretations of the transformer, in: EMNLP, 2022, pp. 8756–8769. doi:10.18653/v1/2022.emnlp-main.599.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[6] K. Yin, G. Neubig, Interpreting language models with contrastive explanations, in: EMNLP 2022, ACL, 2022, pp. 184–198. doi:10.18653/v1/2022.emnlp-main.14.

[7] I. Tenney, J. Wexler, J. Bastings, T. Bolukbasi, A. Coenen, S. Gehrmann, E. Jiang, M. Pushkarna, C. Radebaugh, E. Reif, A. Yuan, The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models, in: EMNLP 2020: System Demonstrations, ACL, 2020, pp. 107–118. doi:10.18653/v1/2020.emnlp-demos.15.

[8] J. Alammar, Ecco: An open source library for the explainability of transformer language models, in: ACL-IJCNLP 2021: System Demonstrations, ACL, 2021, pp. 249–257. doi:10.18653/v1/2021.acl-demo.30.

[9] V. Miglani, A. Yang, A. Markosyan, D. Garcia-Olano, N. Kokhlikyan, Using captum to explain generative language models, in: NLP-OSS 2023, ACL, 2023, pp. 165–173. doi:10.18653/v1/2023.nlposs-1.19.

[10] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: EMNLP 2020: System Demonstrations, ACL, 2020, pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6.

[11] G. Sarti, N. Feldhus, L. Sickert, O. van der Wal, M. Nissim, A. Bisazza, Inseq: An in-

terpretability toolkit for sequence generation models, in: ACL 2023 (Volume 3: System Demonstrations), ACL, 2023, pp. 421–435. URL: https://aclanthology.org/2023.acl-demo.40.

[12] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, O. Reblitz-Richardson, Captum: A unified and generic model interpretability library for PyTorch, arXiv abs/2009.07896 (2020). URL: https://arxiv.org/abs/2009.07896.

[13] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: The Second International Conference on Learning Representations, 2014. URL: http://arxiv.org/abs/1312.6034.

[14] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: ICML 2017, Proceedings of Machine Learning Research, PMLR, 2017, pp. 3145–3153. URL: https://proceedings.mlr.press/v70/shrikumar17a.html.

[15] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: NeurIPS 2017, volume 30, Curran Associates Inc., 2017, p. 4768–4777. URL: https://dl.acm.org/doi/10.5555/3295222.3295230.

[16] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning (ICML), volume 70, Journal of Machine Learning Research (JMLR), 2017, p. 3319–3328.

[17] S. Sanyal, X. Ren, Discretized integrated gradients for explaining language models, in: EMNLP 2021, ACL, 2021, pp. 10285–10299. doi:10.18653/v1/2021.emnlp-main.805.

[18] J. Enguehard, Sequential integrated gradients: a simple but effective method for explaining language models, in: Findings of ACL 2023, ACL, 2023, pp. 7555–7565. doi:10.18653/v1/2023.findings-acl.477.

[19] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: ICLR 2015, 2015. URL: http://arxiv.org/abs/1409.0473.

[20] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 818–833. doi:10.1007/978-3-319-10590-1_53.

[21] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in: KDD 2016, Association for Computing Machinery, 2016, p. 1135–1144. doi:10.1145/2939672.2939778.

[22] H. Mohebbi, W. Zuidema, G. Chrupała, A. Alishahi, Quantifying context mixing in transformers, in: EACL 2023, 2023, pp. 3378–3400. doi:10.18653/v1/2023.eacl-main.245.

[23] Z. Zhao, B. Shan, ReAGent: A model-agnostic feature attribution method for generative language models, in: AAAI Workshop on Responsible Language Models, 2024. URL: https://arxiv.org/abs/2402.00794.

[24] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: ICML 2016, volume 48 of *Proceedings of Machine Learning Research*, Proceedings of Machine Learning Research (PLMR), 2016, pp. 1050–1059. URL: https://proceedings.mlr.press/v48/gal16.html.

[25] P. Fernandes, K. Yin, E. Liu, A. Martins, G. Neubig, When does translation require context? a data-driven, multilingual exploration, in: ACL 2023 (Volume 1: Long Papers), ACL, 2023, pp. 606–626. doi:10.18653/v1/2023.acl-long.36.

[26] S. Lu, S. Chen, Y. Li, D. Bitterman, G. Savova, I. Gurevych, Measuring pointwise $\mathcal{V}$-usable information in-context-ly, in: Findings of EMNLP 2023, ACL, 2023, pp. 15739–15756. doi:10.18653/v1/2023.findings-emnlp.1054.

[27] J. Bastings, S. Ebert, P. Zablotskaia, A. Sandholm, K. Filippova, "will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification, in: EMNLP 2022, ACL, 2022, pp. 976–991. doi:10.18653/v1/2022.emnlp-main.64.

[28] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, W. Wang, Language-agnostic BERT sentence embedding, in: ACL 2022 (Volume 1: Long Papers), ACL, 2022, pp. 878–891. doi:10.18653/v1/2022.acl-long.62.

[29] T. Dettmers, M. Lewis, Y. Belkada, L. Zettlemoyer, GPT3.int8(): 8-bit matrix multiplication for transformers at scale, in: NeurIPS 2022, 2022. URL: https://openreview.net/forum?id=dXiGWqBoxaD.

[30] A. Borzunov, D. Baranchuk, T. Dettmers, M. Riabinin, Y. Belkada, A. Chumachenko, P. Samygin, C. Raffel, Petals: Collaborative inference and fine-tuning of large models, in: ACL 2023 (Volume 3: System Demonstrations), ACL, 2023, pp. 558–568. doi:10.18653/v1/2023.acl-demo.54.

[31] G. Sarti, G. Chrupała, M. Nissim, A. Bisazza, Quantifying the plausibility of context reliance in neural machine translation, in: ICLR 2024, OpenReview, 2024. URL: https://openreview.net/forum?id=XTHfNGI3zT.

[32] J. Qi, G. Sarti, R. Fernández, A. Bisazza, Model internals-based answer attribution for trustworthy retrieval-augmented generation, 2024. URL: https://arxiv.org/abs/2406.13663. arXiv:2406.13663.

[33] S. M. Mousavi, S. Caldarella, G. Riccardi, Response generation in longitudinal dialogues: Which knowledge representation helps?, in: NLP4ConvAI 2023, ACL, 2023, pp. 1–11. doi:10.18653/v1/2023.nlp4convai-1.1.

[34] W. Wang, B. Haddow, A. Birch, W. Peng, Assessing factual reliability of large language model knowledge, in: NAACL, 2024, pp. 805–819. doi:10.18653/v1/2024.naacl-long.46.

[35] A. Molnar, J. Jumelet, M. Giulianelli, A. Sinclair, Attribution and alignment: Effects of local context repetition on utterance production and comprehension in dialogue, in: CoNLL 2023, ACL, 2023, pp. 254–273. doi:10.18653/v1/2023.conll-1.18.

[36] G. Attanasio, F. M. Plaza del Arco, D. Nozza, A. Lauscher, A tale of pronouns: Interpretability informs gender bias mitigation for fairer instruction-tuned machine translation, in: EMNLP 2023, ACL, 2023, pp. 3996–4014. doi:10.18653/v1/2023.emnlp-main.243.

[37] L. Edman, G. Sarti, A. Toral, G. v. Noord, A. Bisazza, Are Character-level Translations Worth the Wait? Comparing ByT5 and mT5 for Machine Translation, Transactions of the Association for Computational Linguistics 12 (2024) 392–410. doi:10.1162/tacl_a_00651.

[38] Q. Wang, T. Anikina, N. Feldhus, J. Genabith, L. Hennig, S. Möller, LLMCheckup: Conversational examination of large language models via interpretability tools and self-explanations, in: HCI-NLP 2024, 2024, pp. 89–104. doi:10.18653/v1/2024.hcinlp-1.9.

[39] T. H. Truong, Y. Otmakhova, K. Verspoor, T. Cohn, T. Baldwin, Revisiting subword tokenization: A case study on affixal negation in large language models To appear in NAACL 2024 (2024). URL: https://arxiv.org/abs/2404.02421.