Inseq: An Interpretability Toolkit for Sequence Generation Models

Gabriele Sarti **Nils Feldhus** **Ludwig Sickert** **Oskar van der Wal**

University of Groningen University of Amsterdam

German Research Center for Artificial Intelligence (DFKI), Berlin

g.sarti@rug.nl nils.feldhus@dfki.de l.sickert@outlook.com o.d.vanderwal@uva.nl

Abstract

Past work in natural language processing interpretability focused mainly on popular classification tasks while largely overlooking generation settings, partly due to a lack of dedicated tools. In this work, we introduce Inseq[1], a Python library to democratize access to interpretability analyses of sequence generation models. Inseq enables intuitive and optimized extraction of models' internal information and feature importance scores for popular decoder-only and encoder-decoder Transformers architectures. We showcase its potential by adopting it to highlight gender biases in machine translation models and locate factual knowledge inside GPT-2. Thanks to its extensible interface supporting cutting-edge techniques such as contrastive feature attribution, Inseq can drive future advances in explainable natural language generation, centralizing good practices and enabling fair and reproducible model evaluations.

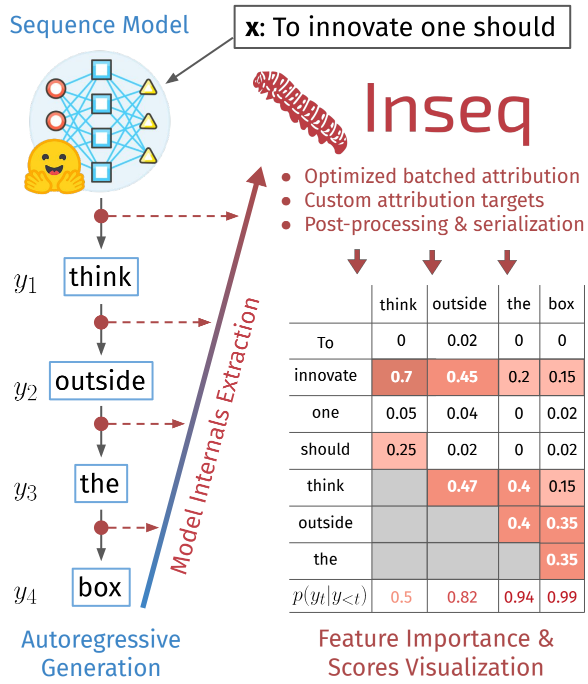

Figure 1: Feature importance and next-step probability extraction and visualization using Inseq with a 🤗 Transformers causal language model.

1 Introduction

Recent years saw an increase in studies and tools aimed at improving our behavioral or mechanistic understanding of neural language models (Belinkov and Glass, 2019). In particular, *feature attribution* methods became widely adopted to quantify the importance of input tokens in relation to models' inner processing and final predictions (Madsen et al., 2022b). Many studies applied such techniques to modern deep learning architectures, including Transformers (Vaswani et al., 2017), leveraging gradients (Baehrens et al., 2010; Sundararajan et al., 2017), attention patterns (Xu et al., 2015; Clark et al., 2019) and input perturbations (Zeiler and Fergus, 2014; Feng et al., 2018) to quantify input importance, often leading to controversial outcomes in terms of faithfulness, plausibility and overall usefulness of such explanations (Adebayo et al., 2018; Jain and Wallace, 2019; Jacovi and Goldberg, 2020; Zafar et al., 2021). However, feature attribution techniques have mainly been applied to classification settings (Atanasova et al., 2020; Wallace et al., 2020; Madsen et al., 2022a; Chrysostomou and Aletras, 2022), with relatively little interest in the more convoluted mechanisms underlying generation. Classification attribution is a single-step process resulting in one importance score per input token, often allowing for intuitive interpretations in relation to the predicted class. Sequential attribution[2] instead involves a computationally expensive multi-step iteration producing a matrix A_{ij} representing the importance of every input i in the prediction of every generation outcome j (Figure 1). Moreover, since previous

[1]Library: `https://github.com/inseq-team/inseq`
Documentation: `https://inseq.readthedocs.io`
This paper describes the Inseq `v0.4.0` release on PyPI.

[2]We use *sequence generation* to refer to all iterative tasks including (but not limited to) natural language generation.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 421–435
July 10-12, 2023 ©2023 Association for Computational Linguistics

generation steps causally influence following predictions, they must be dynamically incorporated into the set of attributed inputs throughout the process. Lastly, while classification usually involves a limited set of classes and simple output selection (e.g. argmax after softmax), generation routinely works with large vocabularies and non-trivial decoding strategies (Eikema and Aziz, 2020). These differences limited the use of feature attribution methods for generation settings, with relatively few works improving attribution efficiency (Vafa et al., 2021; Ferrando et al., 2022) and explanations' informativeness (Yin and Neubig, 2022).

In this work, we introduce **Inseq**, a Python library to democratize access to interpretability analyses of generative language models. Inseq centralizes access to a broad set of feature attribution methods, sourced in part from the Captum (Kokhlikyan et al., 2020) framework, enabling a fair comparison of different techniques for all sequence-to-sequence and decoder-only models in the popular 🤗 Transformers library (Wolf et al., 2020). Thanks to its intuitive interface, users can easily integrate interpretability analyses into sequence generation experiments with just 3 lines of code (Figure 2). Nevertheless, Inseq is also highly flexible, including cutting-edge attribution methods with built-in post-processing features (§ 4.1), supporting customizable attribution targets and enabling constrained decoding of arbitrary sequences (§ 4.2). In terms of usability, Inseq greatly simplifies access to local and global explanations with built-in support for a command line interface (CLI), optimized batching enabling dataset-wide attribution, and various methods to visualize, serialize and reload attribution outcomes and generated sequences (§ 4.3). Ultimately, Inseq's aims to make sequence models first-class citizens in interpretability research and drive future advances in interpretability for generative applications.

```python
import inseq

# Load HF Hub model and attribution method
model = inseq.load_model(
    "google/flan-t5-base",
    "integrated_gradients"
)
# Answer and attribute generation steps
attr_out = model.attribute(
    "Does 3 + 3 equal 6?",
    attribute_target=True
)
# Visualize the generated attribution,
# applying default token-level aggregation
attr_out.show()
```

Source Saliency

	_yes	</s>
_Does	0.264	0.153
_3	0.113	0.099
_+	0.096	0.086
_3	0.096	0.076
_equal	0.213	0.154
_6	0.11	0.108
?	0.106	0.114

Prefix Saliency

	_yes	</s>
_yes		0.21
</s>		

Figure 2: Computing and visualizing source and target-side attributions using Flan-T5 (Chung et al., 2022).

2 Related Work

Feature Attribution for Sequence Generation Work on feature attribution for sequence generation has mainly focused on machine translation (MT). Bahdanau et al. (2015) showed how attention weights of neural MT models encode interpretable alignment patterns. Alvarez-Melis and Jaakkola (2017) adopted a perturbation-based framework to highlight biases in MT systems. Ding et al. (2019); He et al. (2019); Voita et al. (2021a,b) *inter alia* conducted analyses on MT word alignments, coreference resolution and training dynamics with various gradient-based attribution methods. Vafa et al. (2021); Ferrando et al. (2022) developed approaches to efficiently compute sequential feature attributions without sacrificing accuracy. Yin and Neubig (2022) introduced contrastive feature attribution to disentangle factors influencing generation in language models. Attribution scores obtained from MT models were also used to detect hallucinatory behavior (Dale et al., 2022; Tang et al., 2022; Xu et al., 2023), providing a compelling practical use case for such explanations.

Tools for NLP Interpretability Although many post-hoc interpretability libraries were released recently, only a few support sequential feature attribution. Notably, LIT (Tenney et al., 2020), a structured framework for analyzing models across modalities, and Ecco (Alammar, 2021), a library specialized in interactive visualizations of model internals. LIT is an all-in-one GUI-based tool to analyze model behaviors on entire datasets. However, the library does not provide out-of-the-box support for 🤗 Transformers models, requiring the definition of custom wrappers to ensure compatibility. Moreover, it has a steep learning curve due to its advanced UI, which might be inconvenient when working on a small amount of examples. All these factors limit LIT usability for researchers working with custom models, needing access to extracted scores, or being less familiar with interpretability research. On the other hand, Ecco is closer to our work, being based on 🤗 Transformers and having started to support encoder-decoder models concurrently with Inseq development. Despite a marginal overlap in their functionalities, the two libraries provide orthogonal benefits: Inseq's flexible interface makes it especially suitable for methodical quantitative analyses involving repeated evaluations, while Ecco excels in qualitative analyses aimed at visualizing model internals. Other popular tools such as ERASER (DeYoung et al., 2020), Thermostat (Feldhus et al., 2021), transformers-interpret (Pierse, 2021) and ferret (Attanasio et al., 2022) do not support sequence models.

3 Design

Inseq combines sequence models sourced from 🤗 Transformers (Wolf et al., 2020) and attribution methods mainly sourced from Captum (Kokhlikyan et al., 2020). While only text-based tasks are currently supported, the library's modular design[3] would enable the inclusion of other modeling frameworks (e.g. fairseq (Ott et al., 2019)) and modalities (e.g. speech) without requiring substantial redesign. Optional dependencies include 🤗 Datasets (Lhoest et al., 2021) and Rich[4].

3.1 Guiding Principles

Research and Generation-oriented Inseq should support interpretability analyses of a broad set of sequence generation models without focusing narrowly on specific architectures or tasks. Moreover, the inclusion of new, cutting-edge methods should be prioritized to enable fair comparisons with well-established ones.

Scalable The library should provide an optimized interface to a wide range of use cases, models and setups, ranging from interactive attributions of individual examples using toy models to compiling statistics of large language models' predictions for entire datasets.

Beginner-friendly Inseq should provide built-in access to popular frameworks for sequence generation modeling and be fully usable by non-experts at a high level of abstraction, providing sensible defaults for supported attribution methods.

Extensible Inseq should support a high degree of customization for experienced users, with out-of-the-box support for user-defined solutions to enable future investigations into models' behaviors.

	Method	Source	$f(l)$
G	(Input $\times$) Gradient	Simonyan et al.	✓
	DeepLIFT	Shrikumar et al.	✓
	GradientSHAP	Lundberg and Lee	✗
	Integrated Gradients	Sundararajan et al.	✓
	Discretized IG	Sanyal and Ren	✗
I	Attention Weights	Bahdanau et al.	✓
P	Occlusion (Blank-out)	Zeiler and Fergus	✗
	LIME	Ribeiro et al.	✗
S	(Log) Probability	-	
	Softmax Entropy	-	
	Target Cross-entropy	-	
	Perplexity	-	
	Contrastive Prob. Δ	Yin and Neubig	
	μ MC Dropout Prob.	Gal and Ghahramani	

Table 1: Overview of gradient-based (**G**), internals-based (**I**) and perturbation-based (**P**) attribution methods and built-in step functions (**S**) available in Inseq. $f(l)$ marks methods allowing for attribution of arbitrary intermediate layers.

4 Modules and Functionalities

4.1 Feature Attribution and Post-processing

At its core, Inseq provides a simple interface to apply feature attribution techniques for sequence generation tasks. We categorize methods in three groups, *gradient-based*, *internals-based* and *perturbation-based*, depending on their underlying approach to importance quantification.[5] Table 1 presents the full list of supported methods. Aside from popular model-agnostic methods, Inseq notably provides built-in support for attention weight attribution and the cutting-edge Discretized Integrated Gradients method (Sanyal and Ren, 2021). Moreover, multiple methods allow for the importance attribution of custom intermediate model layers, simplifying studies on representational structures and information mixing in sequential models, such as our case study of Section 5.2.

Source and target-side attribution When using encoder-decoder architectures, users can set the

[3]More details are available in Appendix B.
[4]https://github.com/Textualize/rich

[5]We distinguish between gradient- and internals-based methods to account for their difference in scores' granularity.

`attribute_target` parameter to include or exclude the generated prefix in the attributed inputs. In most cases, this should be desirable to account for recently generated tokens when explaining model behaviors, such as when to terminate the generation (e.g. relying on the presence `▁yes` in the target prefix to predict `</s>` in Figure 2, bottom-right matrix). However, attributing the source side separately could prove useful, for example, to derive word alignments from importance scores.

Post-processing of attribution outputs Aggregation is a fundamental but often overlooked step in attribution-based analyses since most methods produce neuron-level or subword-level importance scores that would otherwise be difficult to interpret. Inseq includes several `Aggregator` classes to perform attribution aggregation across various dimensions. For example, the input word “Explanation” could be tokenized in two subword tokens “Expl” and “anation”, and each token would receive N importance scores, with N being the model embedding dimension. In this case, aggregators could first merge subword-level scores into word-level scores, and then merge granular embedding-level scores to obtain a single token-level score that is easier to interpret. Moreover, aggregation could prove especially helpful for long-form generation tasks such as summarization, where word-level importance scores could be aggregated to obtain a measure of sentence-level relevance. Notably, Inseq allows chaining multiple aggregators like in the example above using the `AggregatorPipeline` class, and provides a `PairAggregator` to aggregate different attribution maps, simplifying the conduction of contrastive analyses as in Section 5.1.[6]

4.2 Customizing generation and attribution

During attribution, Inseq first generates target tokens using 🤗 Transformers and then attributes them step by step. If a custom target string is specified alongside model inputs, the generation step is instead skipped, and the provided text is attributed by constraining the decoding of its tokens[7]. Constrained attribution can be used, among other things, for contrastive comparisons of minimal pairs and to obtain model justifications for desired outputs.

Custom step functions At every attribution step, Inseq can use models' internal information to extract scores of interest (e.g. probabilities, entropy) that can be useful, among other things, to quantify model uncertainty (e.g. how likely the generated `▁yes` token was given the context in Figure 2). Inseq provides access to multiple built-in step functions (Table 1, **S**) enabling the computation of these scores, and allows users to create and register new custom ones. Step scores are computed together with the attribution, returned as separate sequences in the output, and visualized alongside importance scores (e.g. the $p(y_t|y_{<t})$ row in Figure 1).

Step functions as attribution targets For methods relying on model outputs to predict input importance (gradient and perturbation-based), feature attributions are commonly obtained from the model's output logits or class probabilities (Bastings et al., 2022). However, recent work showed the effectiveness of using targets such as the probability difference of a contrastive output pair to answer interesting questions like “What inputs drive the prediction of y rather than $\hat{y}$?” (Yin and Neubig, 2022). In light of these advances, Inseq users can leverage any built-in or custom-defined step function as an attribution target, enabling advanced use cases like contrastive comparisons and uncertainty-weighted attribution using MC Dropout (Gal and Ghahramani, 2016).

4.3 Usability Features

Batched and span-focused attributions The library provides built-in batching capabilities, enabling users to go beyond single sentences and attribute even entire datasets in a single function call. When the attribution of a specific span of interest is needed, Inseq also allows specifying a start and end position for the attribution process. This functionality greatly accelerates the attribution process for studies on localized phenomena (e.g. pronoun coreference in MT models).

CLI, Serialization and Visualization The Inseq library offers an API to attribute single examples or entire 🤗 Datasets from the command line and save resulting outputs and visualizations to a file. Attribution outputs can be saved and loaded in JSON format with their respective metadata to easily identify the provenance of contents. Attributions can be visualized in the command line or IPython notebooks and exported as HTML files.

[6] See Appendix C for an example.

[7] Constrained decoding users should be aware of its limitations in the presence of a high distributional discrepancy with natural model outputs (Vamvas and Sennrich, 2021).

Quantized Model Attribution Supporting the attribution of large models is critical given recent scaling tendencies (Kaplan et al., 2020). All models allowing for quantization using `bitsandbytes` (Dettmers et al., 2022) can be loaded in 8-bit directly from 🤗 Transformers, and their attributions can be computed normally using Inseq.[8] A minimal manual evaluation of 8-bit attribution outputs for Section 5.2 study shows minimal discrepancies compared to full-precision results.

5 Case Studies

5.1 Gender Bias in Machine Translation

In the first case study, we use Inseq to investigate gender bias in MT models. Studying social biases embedded in these models is crucial to understand and mitigate the representational and allocative harms they might engender (Blodgett et al., 2020). Savoldi et al. (2021) note that the study of bias in MT could benefit from explainability techniques to identify spurious cues exploited by the model and the interaction of different features that can lead to intersectional bias.

Synthetic Setup: Turkish to English The Turkish language uses the gender-neutral pronoun *o*, which can be translated into English as either "he", "she", or "it", making it interesting to study gender bias in MT when associated with a language such as English for which models will tend to choose a gendered pronoun form. Previous works leveraged translations from gender-neutral languages to show gender bias present in translation systems (Cho et al., 2019; Prates et al., 2020; Farkas and Németh, 2022). We repeat this simple setup using a Turkish-to-English MarianMT model (Tiedemann, 2020) and compute different metrics to quantify gender bias using Inseq.

We select 49 Turkish occupation terms verified by a native speaker (see Appendix E) and use them to infill the template sentence "*O bir* __" (He/She is a(n) __). For each translation, we compute attribution scores for source Turkish pronoun (x_{pron}) and occupation (x_{occ}) tokens[9] when generating the target English pronoun (y_{pron}) using Integrated Gradients (IG), Gradients (∇), and Input $\times$ Gradient (I$\times$G),[10]. We also collect target pronoun probabilities ($p(y_{\text{pron}})$), rank the 49 occupation terms using these metrics, and finally compute Kendall's τ correlation with the percentage of women working in the respective fields, using U.S. labor statistics as in previous works (e.g., Caliskan et al., 2017; Rudinger et al., 2018). Table 2 presents our results.

	Base		♀ → ♂	
	x_{pron}	x_{occ}	x_{pron}	x_{occ}
$p(y_{\text{pron}})$	0.01		-0.44*	
∇	-0.16	0.25*	0.23*	-0.00
IG	-0.08	0.09	0.11	0.17
I×G	-0.11	0.22*	0.22*	-0.01

Table 2: **Gender Bias in Turkish-to-English MT:** Kendall's τ correlation of MT model metrics with U.S. labor statistics. $*$ = Significant correlation ($p < .05$).

In the **base case**, we correlate the different metrics with how much the gender distribution deviates from an equal distribution ($50-50\%$) for each occupation (i.e., the gender bias irrespective of the direction). We observe a strong gender bias, with "she" being chosen only for 5 out of 49 translations and gender-neutral variants never being produced by the MT model. We find a low correlation between pronoun probability and the degree of gender stereotype associated with the occupation. Moreover, we note a weaker correlation for IG compared to the other two methods. For those, attribution scores for x_{occ} show significant correlations with labor statistics, supporting the intuition that the MT model will accord higher importance to source occupation terms associated to gender-stereotypical occupations when predicting the gendered target pronoun.

In the **gender-swap case** (♀ → ♂), we use the `PairAggregator` class to contrastively compare attribution scores and probabilities when translating the pronoun as "She" or "He".[11] We correlate resulting scores with the % of women working in the respective occupation and find strong correlations for $p(y_{\text{pron}})$, supporting the validity of contrastive approaches in uncovering gender bias.

Qualitative Example: English to Dutch We qualitatively analyze biased MT outputs, showing how attributions can help develop hypotheses about models' behavior. Table 3 (top) shows the I $\times$ G attributions for English-to-Dutch translation using M2M-100 (418M, Fan et al., 2021). The model

[8] `bitsandbytes 0.37.0` required for backward method, see Appendix D for an example.

[9] For multi-token occupation terms, e.g., *bilim insanı* (scientist), the attribution score of the first token was used.

[10] We set approx. steps to ensure convergence $\Delta < 0.05$ for IG. All methods use the L2 norm to obtain token-level attributions.

[11] An example is provided in Appendix C.

Source	De	leraar	verliest	zijn	baan
The	0.10	0.08	0.04	0.03	0.02
teacher	0.11	0.20	0.06	0.03	0.05
loses	0.11	0.09	0.25	0.07	0.07
her	0.15	0.09	0.10	0.21	0.07
job	0.10	0.08	0.08	0.10	0.24
Target	De	leraar	verliest	zijn	baan
De		0.23	0.05	0.06	0.04
leraar			0.17	0.13	0.03
verliest				0.18	0.08
zijn					0.26
$p(y_t)$	0.69	0.28	0.35	0.65	0.29

Source	De	♂ → ○	verliest	haar	baan
The	0.00	-0.02	0.00	0.00	0.00
teacher	0.00	-0.05	-0.01	-0.01	-0.01
loses	0.00	-0.02	-0.01	-0.02	-0.01
her	0.00	-0.01	-0.01	-0.10	0.01
job	0.00	-0.02	-0.01	-0.02	-0.02
Target	De	♂ → ○	verliest	haar	baan
De		-0.07	-0.01	0.01	-0.01
♂ → ○			0.09	0.18	0.02
verliest				-0.03	0.00
haar					0.00
$\Delta p(y_t)$	0.00	-0.23	0.13	0.20	0.00

Table 3: **Top:** Attribution of pronoun gender mistranslation using M2M-100. **Bottom:** Target attribution difference when swapping the target noun gender (♂ → ○) from *leraar* (male) to *leerkracht* (gender-neutral).

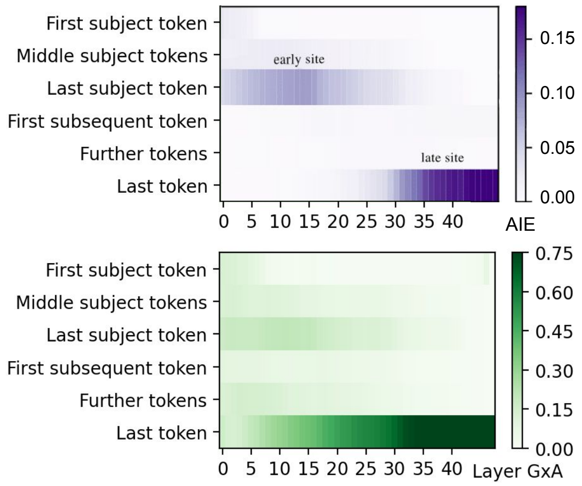

Figure 3: **Top:** Estimated causal importance of GPT-2 XL layers for predicting factual associations, as reported by Meng et al. (2022). **Bottom:** Average GPT-2 XL Gradient × Layer Activation scores obtained with Inseq using contrastive factual pairs as attribution targets.

mistranslates the pronoun "her" into the masculine form *zijn* (his). We find that the wrongly translated pronoun exhibits high probability but does not associate substantial importance to the source occupation term "teacher". Instead, we find good relative importance for the preceding word and *leraar* (male teacher). This suggests a strong prior bias for masculine variants, shown by the pronoun *zijn* and the noun *leraar*, as a possible cause for this mistranslation. When considering the contrastive example obtained by swapping *leraar* with its gender-neutral variant *leerkracht* (Table 3, bottom), we find increased importance of the target occupation in determining the correctly-gendered target pronoun *haar* (her). Our results highlight the tendency of MT models to attend inputs sequentially rather than relying on context, hinting at the known benefits of context-aware models for pronoun translation (Voita et al., 2018).

5.2 Locating Factual Knowledge inside GPT-2 with Contrastive Attribution Tracing

For our second case study, we experiment with a novel attribution-based technique to locate factual knowledge encoded in the layers of GPT-2 1.5B (Radford et al., 2019). Specifically, we aim to reproduce the results of Meng et al. (2022), showing the influence of intermediate layers in mediating the recall of factual statements such as *'The Eiffel Tower is located in the city of → Paris'*. Meng et al. (2022) estimate the effect of network components in the prediction of factual statements as the difference in probability of a correct target (e.g. *Paris*), given a corrupted subject embedding (e.g. for *Eiffel Tower*), before and after restoring clean activations for some input tokens at different layers of the network. Apart from the obvious importance of final token states in terminal layers, their results highlight the presence of an early site associated with the last subject token playing an important role in recalling the network's factual knowledge (Figure 3, top).

To verify such results, we propose a novel knowledge location method, which we name Contrastive Attribution Tracing (CAT), adopting the contrastive attribution paradigm of Yin and Neubig (2022) to locate relevant network components by attributing minimal pairs of correct and wrong factual targets (e.g. *Paris* vs. *Rome* for the example above). To perform the contrastive attribution, we use the Layer Gradient × Activation method, a layer-specific variant of Input × Gradient, to propagate gradients up to intermediate network activations instead of reaching input tokens. The resulting attribution scores hence answer the question "How important are layer L activations for prefix token t in predicting the correct factual target over a wrong one?". We compute attribution scores for 1000 statements taken from the Counterfact Statement dataset (Meng et al., 2022) and present averaged results in Figure 3 (bottom).[12] Our results closely match those of the original authors, providing further evidence of how attribution methods can be used to identify salient network components and guide model editing, as shown by Dai et al. (2022) and Nanda (2023).

To our best knowledge, the proposed CAT method is the most efficient knowledge location technique to date, requiring only a single forward and backward pass of the attributed model. Patching-based approaches such as causal mediation (Meng et al., 2022), on the other hand, provide causal guarantees of feature importance at the price of being more computationally intensive. Despite lacking the causal guarantees of such methods, CAT can provide an approximation of feature importance and greatly simplify the study of knowledge encoded in large language model representations thanks to its efficiency.

6 Conclusion

We introduced Inseq, an easy-to-use but versatile toolkit for interpreting sequence generation models. With many libraries focused on the study of classification models, Inseq is the first tool explicitly aimed at analyzing systems for tasks such as machine translation, code synthesis, and dialogue generation. Researchers can easily add interpretability evaluations to their studies using our library to identify unwanted biases and interesting phenomena in their models' predictions. We plan to provide continued support and explore developments for Inseq,[13] to provide simple and centralized access to a comprehensive set of thoroughly-tested implementations for the interpretability community. In conclusion, we believe that Inseq has the potential to drive real progress in explainable language generation by accelerating the development of new analysis techniques, and we encourage members of this research field to join our development efforts.

Acknowledgments

We thank Ece Takmaz for verifying the Turkish word list used in Section 5.1. GS and AB acknowledge the support of the Dutch Research Council (NWO) as part of the project InDeep (NWA.1292.19.399). NF is supported by the German Federal Ministry of Education and Research as part of the project XAINES (01IW20005). OvdW's contributions are financed by the NWO as part of the project "The biased reality of online media – Using stereotypes to make media manipulation visible" (406.DI.19.059).

Broader Impact and Ethics Statement

Reliability of Attribution Methods The plausibility and faithfulness of attribution methods supported by Inseq is an active matter of debate in the research community, without clear-cut guarantees in identifying specific model behaviors, and prone to users' own biases (Jacovi and Goldberg, 2020). We emphasize that explanations produced with Inseq should not be adopted in high-risk and user-facing contexts. We encourage Inseq users to critically approach results obtained from our toolkit and validate them on a case-by-case basis.

Technical Limitations and Contributions While Inseq greatly simplifies comparisons across different attribution methods to ensure their mutual consistency, it does not provide explicit ways of evaluating the quality of produced attributions in terms of faithfulness or plausibility. Moreover, many recent methods still need to be included due to the rapid pace of interpretability research in natural language processing and the small size of our development team. To foster an open and inclusive development environment, we encourage all interested users and new methods' authors to contribute to the development of Inseq by adding their interpretability methods of interest.

Gender Bias Case Study The case study of Section 5.1 assumes a simplified concept of binary gender to allow for a more straightforward evaluation of the results. However, we encourage other researchers to consider non-binary gender and different marginalized groups in future bias studies. We acknowledge that measuring bias in language models is complex and that care must be taken in its conceptualization and validation (Blodgett et al., 2020; van der Wal et al., 2022; Bommasani and Liang, 2022), even more so in multilingual settings (Talat et al., 2022). For this reason, we do not claim to provide a definite bias analysis of these MT models – especially in light of the aforementioned attributions' faithfulness issues. The study's primary purpose is to demonstrate how attribution methods could be used for exploring social biases in sequence-to-sequence models and showcase the related Inseq functionalities.

[12] Figure 6 of Appendix D presents some examples.

[13] Planned developments available in Appendix F.

References

Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Y. Zou. 2019. Gradio: Hassle-free sharing and testing of ML models in the wild. *ArXiv*, abs/1906.02569.

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, pages 9505–9515, Montréal, Canada. Curran Associates, Inc.

J Alammar. 2021. Ecco: An open source library for the explainability of transformer language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 249–257, Online. Association for Computational Linguistics.

David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Giuseppe Attanasio, Eliana Pastor, Chiara Di Bonaventura, and Debora Nozza. 2022. ferret: a framework for benchmarking explainers on transformers. *ArXiv*, abs/2208.01575.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. "will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

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.

Rishi Bommasani and Percy Liang. 2022. Trustworthy social bias measurement. *ArXiv*, abs/2212.11672.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Won Ik Cho, Ji Won Kim, Seok Min Kim, and Nam Soo Kim. 2019. On measuring gender bias in translation of gender-neutral pronouns. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 173–181, Florence, Italy. Association for Computational Linguistics.

George Chrysostomou and Nikolaos Aletras. 2022. An empirical study on explanations in out-of-domain settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6920–6938, Dublin, Ireland. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

David Dale, Elena Voita, Loïc Barrault, and Marta Ruiz Costa-jussà. 2022. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better. *ArXiv*, abs/2212.08597.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

Anna Farkas and Renáta Németh. 2022. How to measure gender bias in machine translation: Real-world oriented machine translators, multiple reference points. *Social Sciences & Humanities Open*, 5(1):100239.

Nils Feldhus, Robert Schwarzenberg, and Sebastian Möller. 2021. Thermostat: A large collection of NLP model explanations and analysis tools. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 87–95, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8756–8769, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Javier Ferrando, Gerard I. G'allego, Ioannis Tsiamas, and Marta Ruiz Costa-jussà. 2023. Explaining how transformers use context to build predictions. *ArXiv*, abs/2305.12535.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, NY, USA. Proceedings of Machine Learning Research (PLMR).

Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. 2019. Towards understanding neural machine translation with word importance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 953–962, Hong Kong, China. Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Sarthak Jain, Varun Manjunatha, Byron Wallace, and Ani Nenkova. 2022. Influence functions for sequence tagging models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 824–839, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhiying Jiang, Raphael Tang, Ji Xin, and Jimmy Lin. 2020. Inserting Information Bottlenecks for Attribution in Transformers. In *Findings of the Association* for Computational Linguistics: EMNLP 2020*, pages 3850–3857, Online. Association for Computational Linguistics.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. 2021. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5050–5058.

Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv*, abs/2001.08361.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2021. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4547–4568, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Feed-forward blocks control contextualization in masked language models. *ArXiv*, abs/2302.00456.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch. *ArXiv*, abs/2009.07896.

Tsz Kin Lam, Eva Hasler, and Felix Hieber. 2022. Analyzing the use of influence functions for instance-specific data filtering in neural machine translation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 295–309, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Shahar Levy, Koren Lazar, and Gabriel Stanovsky. 2021. Collecting a large-scale gender bias dataset for coreference resolution and machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2470–2480, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, page 4768–4777, Long Beach, California, USA. Curran Associates Inc.

Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. 2022a. Evaluating the faithfulness of importance measures in NLP by recursively masking allegedly important tokens and retraining. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1731–1751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022b. Post-hoc interpretability for neural nlp: A survey. *ACM Comput. Surv.*, 55(8).

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States. Association for Computational Linguistics.

Hosein Mohebbi, Willem H. Zuidema, Grzegorz Chrupała, and A. Alishahi. 2023. Quantifying context mixing in transformers. *ArXiv*, abs/2301.12971.

Neel Nanda. 2023. Attribution patching: Activation patching at industrial scale. Blog post.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Charles Pierse. 2021. Transformers interpret. Python library.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. 2020. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, 32:6363–6381.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

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*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Gender bias in machine translation. *Transactions of the Association for Computational Linguistics*, 9:845–874.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, page 3319–3328. Journal of Machine Learning Research (JMLR).

Zeerak Talat, Aurélie Névéol, Stella Biderman, Miruna Clinciu, Manan Dey, Shayne Longpre, Sasha Luccioni, Maraim Masoud, Margaret Mitchell, Dragomir Radev, Shanya Sharma, Arjun Subramonian, Jaesung Tae, Samson Tan, Deepak Tunuguntla, and Oskar Van Der Wal. 2022. You reap what you sow: On the challenges of bias evaluation under multilingual settings. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 26–41, virtual+Dublin. Association for Computational Linguistics.

Joel Tang, M. Fomicheva, and Lucia Specia. 2022. Reducing hallucinations in neural machine translation with feature attribution. *ArXiv*, abs/2211.09878.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 107–118, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Keyon Vafa, Yuntian Deng, David Blei, and Alexander Rush. 2021. Rationales for sequential predictions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10314–10332, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2021. On the limits of minimal pairs in contrastive evaluation. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 58–68, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Oskar van der Wal, Dominik Bachmann, Alina Leidinger, Leendert van Maanen, Willem Zuidema, and Katrin Schulz. 2022. Undesirable biases in nlp: Averting a crisis of measurement. *ArXiv*, abs/2211.13709.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021a. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1126–1140, Online. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021b. Language modeling, lexical translation, reordering: The training process of NMT through the lens of classical SMT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8478–8491, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Eric Wallace, Matt Gardner, and Sameer Singh. 2020. Interpreting predictions of NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 20–23, 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.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.

Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J. Martindale, and Marine Carpuat. 2023. Understanding and detecting hallucinations in neural machine translation via model introspection. *ArXiv*, abs/2301.07779.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Muhammad Bilal Zafar, Michele Donini, Dylan Slack, Cedric Archambeau, Sanjiv Das, and Krishnaram Kenthapadi. 2021. On the lack of robust interpretability of neural text classifiers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3730–3740, Online. Association for Computational Linguistics.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham. Springer International Publishing.

A Authors' Contributions

Authors jointly contributed to the writing and revision of the paper.

Gabriele Sarti Organized and led the project, developed the first public release of the Inseq library, conducted the case study of Section 5.2.

Nils Feldhus Implemented the perturbation-based methods in Inseq and contributed to the validation of the case study of Section 5.2.

Ludwig Sickert Implemented the attention-based attribution method in Inseq.

Oskar van der Wal Conducted the experiments in the gender bias case study of Section 5.1.

Malvina Nissim and **Arianna Bisazza** ensured the soundness of the overall process and provided valuable inputs for the initial design of the toolkit.

B Additional Design Details

Figure 4 presents the Inseq hierarchy of models and attribution methods. The model-method connection enables out-of-the-box attribution using the selected method. Framework-specific and architecture-specific classes enable extending Inseq to new modeling architectures and frameworks.

C Example of Pair Aggregation for Contrastive MT Comparison

An example of gender translation pair using the synthetic template of Section 5.1 is show in Figure 5, highlighting a large drop in probability when switching the gendered pronoun for highly gender-stereotypical professions, similar to Table 2 results.

D Example of Quantized Contrastive Attribution of Factual Knowledge

Figure 6 presents code used in Section 5.2 case study, with visualized attribution scores for contrastive examples in the evaluated dataset.

E Gender Bias in Machine Translation

Table 4 shows the list of occupation terms used in the gender bias case study (Section 5.1). We correlate the ranking of occupations based on the selected attribution metrics and probabilities with U.S. labor statistics[14] (`bls_pct_female` column). Table 3 example was taken from the BUG dataset (Levy et al., 2021).

[14]https://github.com/rudinger/winogender-schemas

Turkish	English	Turkish	English
teknisyen	technician	memur	officer
muhasebeci	accountant	patolog	pathologist
süpervizör	supervisor	öğretmen	teacher
mühendis	engineer	avukat	lawyer
işçi	worker	planlamacı	planner
eğitimci	educator	yönetici	practitioner
katip	clerk	tesisatçı	plumber
danışman	consultant	eğitmen	instructor
müfettiş	inspector	cerrah	surgeon
tamirci	mechanic	veteriner	veterinarian
müdür	manager	kimyager	chemist
terapist	therapist	makinist	machinist
resepsiyonist	receptionist	mimar	architect
kütüphaneci	librarian	kuaför	hairdresser
ressam	painter	fırıncı	baker
eczacı	pharmacist	programlamacı	programmer
kapıcı	janitor	itfaiyeci	firefighter
psikolog	psychologist	bilim insanı	scientist
doktor	physician	sevk memuru	dispatcher
marangoz	carpenter	kasiyer	cashier
hemşire	nurse	komisyoncu	broker
araştırmacı	investigator	şef	chef
barmen	bartender	doktor	doctor
uzman	specialist	sekreter	secretary
elektrikçi	electrician		

Table 4: List of the 49 Turkish occupation terms and their English translations used in the gender bias case study (Section 5.1).

	Method	Source
G	Guided Integrated Gradients	Kapishnikov et al.
	LRP	Bach et al.
I	Attention Rollout & Flow	Abnar and Zuidema
	Attention × Vector Norm	Kobayashi et al.
	Attention × Attn. Block Norm	Kobayashi et al.
	GlobEnc	Modarressi et al.
	ALTI+	Ferrando et al.
	Attention × Trans. Block Norm	Kobayashi et al.
	ALTI-Logit	Ferrando et al.
P	Information Bottlenecks	Jiang et al.
	Value Zeroing	Mohebbi et al.
	Input Reduction	Feng et al.
	Activation Patching	Meng et al.

Table 5: Gradient-based (**G**), internals-based (**I**) and perturbation-based (**P**) attribution methods for which we plan to include support in future Inseq releases.

F Planned Developments and Next Steps

We plan to continuously expand the core functionality of the library by adding support for a wider range of attribution methods. Table 5 shows a subset of methods we consider including in future releases. Besides new methods, we also intend to significantly improve result visualization using an interactive interface backed by Gradio Blocks (Abid et al., 2019), work on interoperability features together with ferret developers (Attanasio et al., 2022) to simplify the evaluation of sequence attributions, and include sequential instance attribution methods (Lam et al., 2022; Jain et al., 2022) for training data attribution.

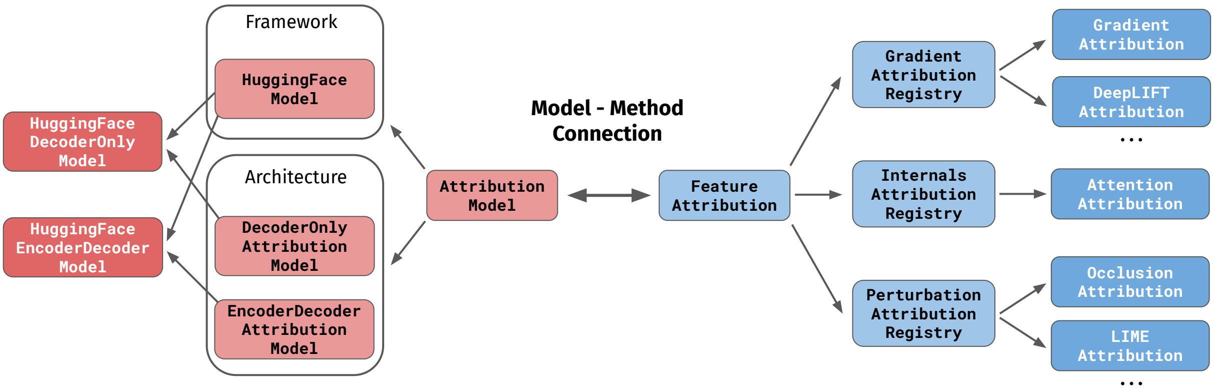

Figure 4: Inseq models and attribution methods. Concrete classes combine abstract framework and architecture attribution models classes, and are derived from abstract attribution methods' categories.

```python
import inseq
from inseq.data.aggregator import AggregatorPipeline, SubwordAggregator,
    SequenceAttributionAggregator, PairAggregator

# Load the TR-EN translation model and attach the IG method
model = inseq.load_model("Helsinki-NLP/opus-mt-tr-en", "integrated_gradients")

# Batch attribute with forced decoding. Return probabilities, no target attr.
out = model.attribute(
    ["O bir teknisyen", "O bir teknisyen"],
    ["She is a technician.","He is a technician."],
    step_scores=["probability"],
    # The following attributes are specific to the IG method
    internal_batch_size=100,
    n_steps=300
)

# Aggregation pipeline composed by two steps:
# 1. Aggregate subword tokens across all dimensions: [l1, l2, dim] -> [l3, l4, dim]
# 2. Aggregate hidden size to produce token-level attributions: [l1, l2, dim] -> [l1, l2]
subw_aggregator = AggregatorPipeline([SubwordAggregator, SequenceAttributionAggregator])

# Aggregate attributions using the pipeline
masculine = out.sequence_attributions[0].aggregate(aggregator=subw_aggregator)
feminine = out.sequence_attributions[1].aggregate(aggregator=subw_aggregator)

# Take the diff of the scores of the two attributions, show it and return the HTML
html = masculine.show(aggregator=PairAggregator, paired_attr=feminine, return_html=True)
```

Source Saliency Heatmap
x: Generated tokens, y: Attributed tokens

	_She → _He	_is	_a	_technician.	</s>
_O	0.115	-0.004	0.011	0.003	0.014
_bir	0.069	-0.023	-0.019	-0.006	-0.015
_teknisyen	-0.184	0.027	0.008	0.003	0.001
</s>	0.0	0.0	0.0	0.0	0.0
probability	0.46	0.004	0.003	-0.014	0.001

Figure 5: Comparing attributions for a synthetic Turkish-to-English translation example with underspecified source pronoun gender using a MarianMT Turkish-to-English translation model (Tiedemann, 2020). Values in the visualized attribution matrix show a 46% higher probability of producing the masculine pronoun in the translation and a relative decrease of 18.4% in the importance of the Turkish occupation term compared to the feminine pronoun case.

```python
import inseq
from datasets import load_dataset
from transformers import AutoModelForCausalLM, AutoTokenizer

# The model is loaded in 8-bit on available GPUs
model = AutoModelForCausalLM.from_pretrained("gpt2-xl", load_in_8bit=True, device_map="auto")
tokenizer = AutoTokenizer.from_pretrained("gpt2-xl")
# Counterfact datasets used by Meng et al. (2022)
data = load_dataset("NeelNanda/counterfact-tracing")["train"]

# GPT-2 XL is a Transformer model with 48 layers
for layer in range(48):
    attrib_model = inseq.load_model(
        model,
        "layer_gradient_x_activation",
        tokenizer="gpt2-xl",
        target_layer=model.transformer.h[layer].mlp,
    )
    for i, ex in data:
        # e.g. "The capital of Second Spanish Republic is"
        prompt = ex["relation"].format{ex["subject"]}
        # e.g. "The capital of Second Spanish Republic is Madrid"
        true_answer = prompt + ex["target_true"]
        # e.g. "The capital of Second Spanish Republic is Paris"
        false_answer = prompt + ex["target_false"]
        contrast = attrib_model.encode(false_answer)
        # Contrastive attribution of true vs false answer
        out = attrib_model.attribute(
            prompt,
            true_answer,
            attributed_fn="contrast_prob_diff",
            contrast_ids=contrast.input_ids,
            contrast_attention_mask=contrast.attention_mask,
            step_scores=["contrast_prob_diff"],
            show_progress=False,
        )
    # Aggregation and plotting omitted for brevity
    ...
```

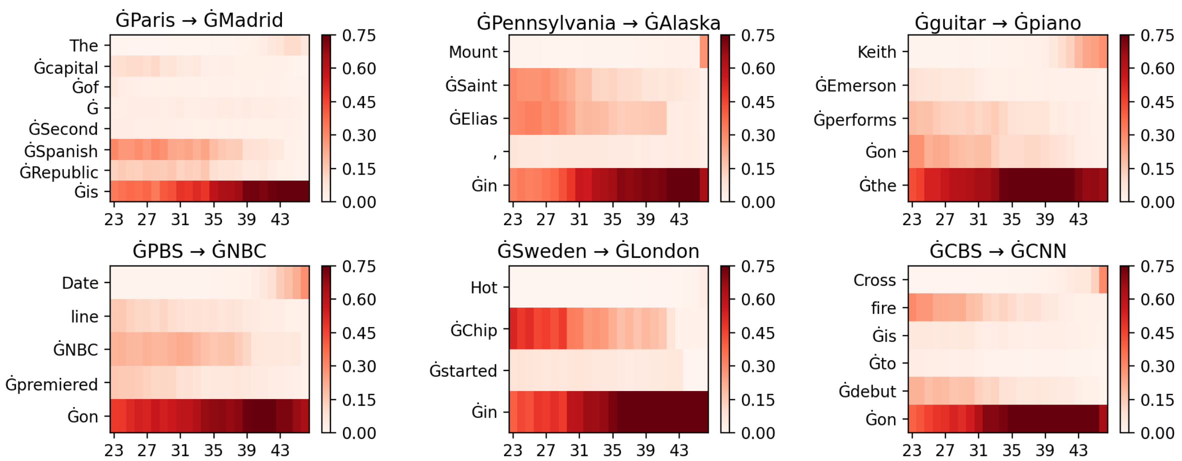

Figure 6: **Top:** Example code to contrastively attribute factual statements from the Counterfact Tracing dataset, using Layer Gradient × Activation to compute importance scores until intermediate layers of the GPT2-XL model. **Bottom:** Visualization of contrastive attribution scores on a subset of layers (23 to 48) for some selected dataset examples. Plot labels show the contrastive pairs of false → true answer used as attribution targets.