

RAG-E: Quantifying Retriever-Generator Alignment and Failure Modes

Korbinian Randl¹ Guido Rocchietti² Aron Henriksson¹ Ziawasch Abedjan² Tony Lindgren¹
John Pavlopoulos^{3,4,1}

Abstract

Retrieval-Augmented Generation (RAG) systems combine dense retrievers and language models to ground LLM outputs in retrieved documents. However, the opacity of how these components interact creates challenges for deployment in high-stakes domains. We present RAG-E, an end-to-end explainability framework that quantifies retriever-generator alignment through mathematically grounded attribution methods. Our approach adapts Integrated Gradients for retriever analysis, introduces PMCSHAP, a Monte Carlo-stabilized Shapley Value approximation, for generator attribution, and introduces the Weighted Attribution-Relevance Gap (WARG) metric to measure how well a generator’s document usage aligns with a retriever’s ranking. Empirical analysis on TREC CAsT and FoodSafeSum reveals critical misalignments: for 47.4% to 66.7% of queries, generators ignore the retriever’s top-ranked documents, while 48.1% to 65.9% rely on documents ranked as less relevant. These failure modes demonstrate that RAG output quality depends not solely on individual component performance but on their interplay, which can be audited via RAG-E.

1. Introduction

Retrieval Augmented Generation (Lewis et al., 2020, RAG) has become a standard in modern Question-Answering tasks, from the everyday use of ChatGPT to applications in critical domains like medicine and law (Amugongo et al., 2025; Brown et al., 2025). A standard RAG pipeline operates as a two-stage process: a *Retriever* (RET) is responsible for

¹Department of Computer and Systems Sciences, Stockholm University, Borgarfjordsgatan 12, Kista 164 07, Sweden
²BIFOLD, Technische Universität Berlin, Franklinstr. 28/29, 10587 Berlin, Germany
³Department of Informatics, Athens University of Economics and Business, Patision 76, Athens 104 34, Greece
⁴Archimedes, Athena Research Centre, Artemidos 1, Marousi 151 25, Greece. Correspondence to: Korbinian Randl <korbinian.randl@dsv.su.se>.

Preprint. January 30, 2026.

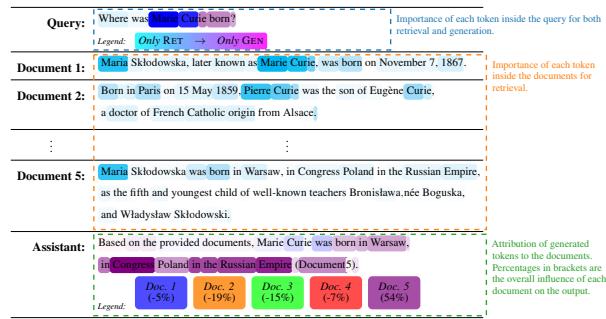


Figure 1. Visual example of the generated explanations. We detect important spans influencing both the retrieval and generation steps. This example was generated using Arctic Embed 2 and Llama 3.1.

ranking the documents by relevance to a user query. A *Generator* (GEN) synthesizes the information contained in the top- k ranked documents and generates an answer in natural language. While RAG reduces the opacity of Large Language Models (LLMs) by grounding its responses in specific sources, its internal reasoning is not inherently transparent. On the one hand, considering all the retrieved documents as sources leaves the uncertainty of which specific documents contain the relevant information. On the other hand, LLM-generated attribution is statistical in nature and not necessarily faithful (Randl et al., 2025). In this work, we address this uncertainty by introducing RAG-E, a mathematically grounded explainability framework for RAG. An example of the explanations provided by our framework is shown in Fig. 1, where we compute token saliency for RET input and GEN query input and document attribution. Using RAG-E, we seek to gain a better understanding of RAG:

RQ1: What do transformer-based and state-of-the-art RET and GEN models focus on?

Our results suggest that even neural network-based RET models select documents based on the existence of keywords also found in the query. Furthermore, we find that GEN statistically prefers documents early in the prompt over documents later on. This is expected as RAG explicitly leverages the ordered nature of the retrieved documents. Even so, the fact that the position of information in the prompt influences its probability of occurring in the output is also known in non-RAG LLMs (Liu et al., 2024) and can

become problematic if GEN ignores important sources in favour of higher-ranked sources.

An empirical analysis, conducted with LLama 3.1 8B and Gemma 3 13B on two datasets spanning open-domain and domain-specific queries, shows that for 60% of queries on domain-specific data, GEN ignores RET’s top-ranked document, while in 57% of cases it relies primarily on a document RET attributed with lower relevancy. This reflects two failure modes, which we term *wasted retrieval* and *noise distraction*, revealing that RAG output depends not solely on individual component performance but on their interplay. The lack of a single metric assessing this misalignment motivates our second research question.

RQ2: *How can the agreement between RET and GEN be quantified?*

As RAG systems proliferate in high-stakes domains, addressing misalignment between RET and GEN becomes increasingly important. In fact, a medical RAG system that ignores the best evidence or a legal system that focuses on marginal precedent might generate unreliable outputs with real consequences. Nonetheless, prior work on RAG systems typically focuses on explaining either RET or GEN in isolation. For example, Zhuang et al. (2021); Fernando et al. (2019) study retriever explanations, while Qi et al. (2024); Cohen-Wang et al. (2024) offer source attribution for the generator (more details in §6); however, there is no way to quantify how information flows across both components or how they diverge. Being able to do so can help not only by improving the transparency of RAG systems but also by increasing alignment between the two components and possibly reducing computational cost, e.g., by retrieving fewer documents if we know some of them will not be used by GEN. To address this gap, **we propose an end-to-end RAG auditing framework with three components:**

Attribution methods tailored to RAG: We introduce PMCSHAP, a Monte Carlo stabilized variant of KernelSHAP (Lundberg & Lee, 2017, KSHAP) that achieves significantly more accurate and reproducible approximations of Shapley Values (Shapley, 1953, SV) for autoregressive GENS. This addresses a fundamental limitation of KSHAP: its instability when applied to variable-length generation with dependent features (overlapping documents). Furthermore, we establish a baseline embedding for Integrated Gradients (Sundararajan et al., 2017, IG) on dense retrievers through systematic empirical analysis, showing that replacing non-special tokens with the [unk] embedding significantly outperforms baselines.

Diagnostic metrics for alignment: We propose Weighted Attribution-Relevance Gap (WARG), a novel metric based on Rank-Biased Overlap (Webber et al., 2010, RBO) that quantifies how well GEN’s use of documents aligns with

RET’s ranking. By sweeping a bias parameter $p \in (0, 1)$, it analyses whether misalignment is concentrated at the top (indicative of primacy bias) or diffuse across the ranking.

Empirical discovery of failure modes: Our framework shows that structural misalignment is prevalent and model-dependent. Llama 3.1 8B exhibits primacy bias (trusts prompt ranking even when shuffled), while Gemma 3 12B demonstrates a more semantic-driven behaviour and still relies on low-ranked documents in 57% of cases. We provide open-source tooling (RAG-E package) to enable community-wide RAG auditing and improvement.

The remainder of the paper is structured as follows. We provide important background information (§2) and details about the technical implementation of our framework (§3). Then, we present a comprehensive empirical analysis across two RET models, two GEN models, and two datasets (§4 and §5), showing failure modes and insights. Last, we present related work and discuss (§6), before concluding (§7).

2. Background

Since the original proposal of RAG by Lewis et al. (2020), several extensions have been proposed. Examples include Self-RAG (Asai et al., 2024), which introduces retrieval on demand, and ATLAS (Izacard et al., 2023), which leverages RAG to improve few-shot learning. We focus on the original framework, which serves as a conceptual basis for most RAG approaches. To assess information flow through the RET and GEN models in the RAG pipeline, we employ saliency-based local explainability techniques.

Definition 2.1 (Saliency-Based Local Explanation). Given a Machine Learning (ML) model $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, that maps input vectors $\mathbf{x} = [x_1, x_2, \dots, x_n]$ to output vectors $\mathbf{y} = [y_1, y_2, \dots, y_m]$, the matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$ is a local explanation for the specific input-output pair $\bar{\mathbf{y}} = f(\bar{\mathbf{x}})$, iff its elements $\beta_{i,j}$ describe the impact of feature \bar{x}_i on the output \bar{y}_j for all indices $i \in 1, 2, \dots, n$ and $j \in 1, 2, \dots, m$.

Since the Language Models (LMs) used in this paper do not map from or to \mathbb{R} but a set of tokens \mathcal{T} , they usually rely on an *embedding function* $\Phi : \mathcal{T}^n \rightarrow \mathbb{R}^n$ and/or a *decoder function* $\Omega : \mathbb{R}^n \rightarrow \mathcal{T}^n$ for these steps.

Generally, methods explaining ML models are commonly separated into *intrinsic* methods, deriving explanations based on the internal state of the model, and *extrinsic* or *model-agnostic* methods, which statistically infer explanations from input-to-output-relationships without considering model internals. We argue that intrinsic methods are preferable over extrinsic methods, as their output is directly tied to the explained models’ function, while extrinsic methods are only statistically correct. Nevertheless, extrinsic methods are independent of the model’s design and therefore more flexible. See Appendix A for extensive background on such

methods; we focus purely on *additive* methods in this paper.

Definition 2.2 (Additivity). A linear feature attribution method adhering to Definition 2.1 is called **additive**, if, given a baseline input $\mathbf{x}^0 = [x_1^0, x_2^0, \dots, x_n^0]$ and its corresponding model output $\mathbf{y}^0 = [y_1^0, y_2^0, \dots, y_m^0]$, the sum of the attribution scores $\beta_{i,j}$ adds up to the difference of the model prediction \bar{y}_j and the baseline prediction y_j^0 :

$$\bar{y}_j - y_j^0 = \sum_{i=1}^n \beta_{i,j} \quad \forall j \in \{1, 2, \dots, m\}. \quad (1)$$

Methods matching Definition 2.2 therefore directly attribute a specific part of the output to each input feature. Some methods require explicitly specifying $(\mathbf{x}^0, \mathbf{y}^0)$ (Sundararajan et al., 2017; Shrikumar et al., 2017), while others assume them implicitly (Shapley, 1953; Lundberg & Lee, 2017). While in theory any saliency map can be normalized to fulfil the additivity attribute, requiring intrinsic theoretical additivity in all our methods is favourable in two respects: **(i)** it ensures comparability of all produced saliency maps both in scale and interpretation, and **(ii)** the error ratio $\frac{\sum_{i=1}^n \beta_{i,j}}{\bar{y}_j - y_j^0}$ can be used as a quality metric for the explanation **B**.

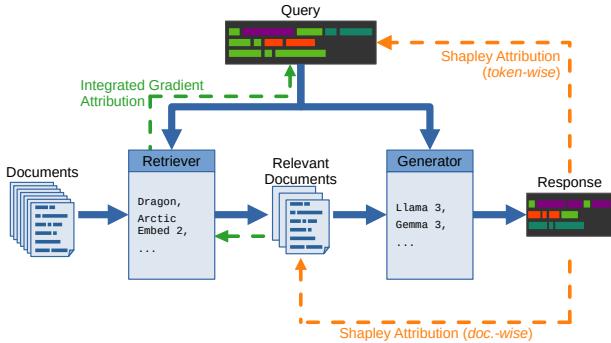


Figure 2. RAG-E overview. Explanations are based on intrinsic IG (—) for the RET and extrinsic Shapley for the GEN (—).

3. Method

To answer our research questions, we propose an explainability framework for RAG.¹ As can be seen in Fig. 2, the framework relies on separate methods to track saliency through RET and GEN. Specifically, in our approach, we compute IG (Sundararajan et al., 2017) attributions on the RET and SV (Shapley, 1953) based attributions on the GEN output. This twofold choice is necessary to optimize the trade-off between explanation faithfulness and runtime: as we focus on transformer-based encoder-only RETs in this work, which are comparably small neural network-based LMs, the

¹ The source code is available on GitHub under [k-rndl/Interpretable_RAG](#).

choice of an intrinsic method is possible. Furthermore, IG’s time-complexity does not scale with the number of input tokens (which can be high for the documents) as this is the case with comparable extrinsic methods (Shapley, 1953; Lundberg & Lee, 2017; Ribeiro et al., 2016). In the face of the increasing architectural variability of LLMs, we opt for an extrinsic method for GEN. Given the typically low number of query tokens and context documents, the previously mentioned time-complexity issue is less problematic here.

3.1. Retriever Explanations: Integrated Gradients

To explain RET, we adapt IG to RAG. RET encodes a query \mathbf{q} and document \mathbf{d} using transformer-based encoders $e^{\text{qry}}(\cdot)$ and $e^{\text{ctx}}(\cdot)$, ranking documents by their dot-product:

$$s^{\text{ret}}(\mathbf{q}, \mathbf{d}) = e^{\text{qry}}(\mathbf{q}) \cdot e^{\text{ctx}}(\mathbf{d}), \quad (2)$$

and retrieving the k documents for which the summed similarity is maximal. Both encoders are pre-trained transformers $f(\cdot)$ applied to token embeddings $\Phi^{\text{qry}}(\cdot)$ and $\Phi^{\text{ctx}}(\cdot)$:

$$e^{\text{qry}}(\mathbf{q}) = f^{\text{qry}}(\Phi^{\text{qry}}(\mathbf{q})), \quad e^{\text{ctx}}(\mathbf{d}) = f^{\text{ctx}}(\Phi^{\text{ctx}}(\mathbf{d})) \quad (3)$$

As the computation for IG is analogous for query and contexts, we omit the specifiers “qry” and “ctx” in the following, and refer to both \mathbf{q} and \mathbf{d} as \mathbf{x} . IG approximates a model $\mathbf{y} = f(\mathbf{x})$ by integrating its gradients with regard to each input feature x_i over x_i , starting from a chosen baseline of x_i^0 . Following IG, we compute the attributions $\beta_i^{\text{ret}, \mathbf{x}}$ based on the embeddings using Riemann integration with L steps. Since the retrieval pipeline receives multiple inputs, we calculate the saliency for the query and each of the retrieved documents separately, holding all other inputs fixed to avoid cross-effects. Given $[\phi_1, \dots, \phi_n] = \Phi(\mathbf{x})$ and a baseline embedding $[\phi_1^0, \dots, \phi_n^0] = \Phi^0(\mathbf{x})$, we compute saliency as:

$$\beta_i^{\text{ret}, \mathbf{x}} = (\phi_i - \phi_i^0) \cdot \sum_{l=0}^L \frac{\partial s}{\partial \phi_i} \frac{1}{L}. \quad (4)$$

Here $\mathbf{s} = s^{\text{ret}}(\delta(\mathbf{q}, l), \mathbf{d})$ for the query and $s^{\text{ret}}(\mathbf{q}, \delta(\mathbf{d}, l))$ for each document, with

$$\delta(\mathbf{x}, l) = f \left(\Phi(\mathbf{x}) + \frac{l}{L} \cdot (\Phi(\mathbf{x}) - \Phi^0(\mathbf{x})) \right). \quad (5)$$

We compare different choices for the baseline embedding $\Phi^0(\mathbf{x})$ in §4.1 and use the best candidate: replacing non-special tokens with the model’s [unk] token in the rest of the paper. Fig. 3 illustrates this process.

3.2. Generator Explanations: Shapley Style Attributions

In general, the GEN $\mathbf{y} = g(\mathbf{q}, \mathcal{D})$ can be split into two parts: **(i)** the function $[t_0^{\mathcal{D}}, t_1^{\mathcal{D}}, \dots, t_n^{\mathcal{D}}] = \text{create_prompt}(\mathbf{q}, \mathcal{D})$ combines the query \mathbf{q} and set of retrieved documents $\mathcal{D} =$

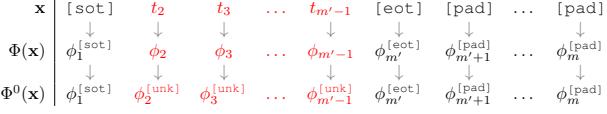


Figure 3. Baseline creation for IG. We replace the embeddings corresponding to non-special tokens with embeddings corresponding to the model’s `[unk]` token evaluated at the same input position. `[XXX]` denotes a special token and $\phi_i^{[XXX]}$ the corresponding embedding at input position i .

$\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k\}$ (ordered by descending relevance) to a single sequence of tokens. The function used in this paper is illustrated in Appendix B. (ii) An autoregressive (L)LM iteratively completes this prompt sequence:

$$t_{i+1}^{\mathcal{D}} = \text{LLM} \left(\underbrace{[t_0^{\mathcal{D}}, t_1^{\mathcal{D}}, \dots, t_n^{\mathcal{D}}]}_{\text{prompt } \mathbf{x}}, \underbrace{[t_{n+1}^{\mathcal{D}}, \dots, t_i^{\mathcal{D}}]}_{\text{previous generation}} \right) \quad (6)$$

The output of this (L)LM is the sequence of generated tokens excluding the prompt: $\mathbf{y} = [t_{n+1}^{\mathcal{D}}, \dots, t_m^{\mathcal{D}}]$.

Given \mathbf{q} , \mathcal{D} , and a GEN function $[t_0^{\mathcal{D}'}, \dots, t_n^{\mathcal{D}'}] = g(\mathbf{q}, \mathcal{D}')$ that returns the sequence of tokens based on the subset of documents $\mathcal{D}' \subseteq \mathcal{D}$, the SV for input document \mathbf{d}_i and output token $t_j^{\mathcal{D}'}$ averages the marginal contributions over all permutations (Shapley, 1953):

$$\beta_{i,j}^{\text{gen}} = \sum_{\mathcal{D}' \subseteq \mathcal{D} \setminus \{\mathbf{d}_i\}} \underbrace{\frac{|\mathcal{D}'|!(|\mathcal{D}| - |\mathcal{D}'| - 1)!}{|\mathcal{D}'|!}}_{\text{likelihood of } \mathcal{D}' \text{ appearing in a random permutation}} \underbrace{\left[t_j^{\mathcal{D}' \cup \{\mathbf{d}_i\}} - t_j^{\mathcal{D}'} \right]}_{\text{marginal contribution}} \quad (7)$$

Computing this precise SV has exponential time complexity $\mathcal{O}(2^{|\mathcal{D}|})$ as the GEN model $g(\cdot)$ needs to be called for each possible subset $\mathcal{D}' \subseteq \mathcal{D}$. This makes computation feasible only for small numbers of documents (e.g. $|\mathcal{D}| = 6$).

A well-established approximation of SVs for higher $|\mathcal{D}|$ is kSHAP: Lundberg & Lee (2017) show that a linear surrogate model $g'(\cdot)$, trained by minimizing the loss function

$$\sum_{\mathcal{D}' \subseteq \mathcal{D}} \left(\frac{|\mathcal{D}|}{|\mathcal{D}'|} \right)^{-1} \frac{|\mathcal{D}| - 1}{|\mathcal{D}'| \cdot (|\mathcal{D}| - |\mathcal{D}'|)} \cdot (g(\mathbf{q}, \mathcal{D}') - g'(\mathbf{q}, \mathcal{D}'))^2 \quad (8)$$

produces coefficients that approximate SVs and are consistent with their mathematical properties. This trades time complexity for faithfulness, as a lower number $N \leq 2^{|\mathcal{D}|}$ of training samples \mathcal{D}' can be used. Note that, contrary to precise SVs, kSHAP assumes independent documents. This assumption is unrealistic, as documents collected to answer a single query are prone to have overlapping content.

Monte-Carlo (MC) stabilization of kSHAP attributions:

As a solution, we propose and evaluate independent sampling strategies of perturbed input-output pairs for kSHAP.

Specifically, we compare native kSHAP and repetitive sampling of kSHAP in an MC fashion (referred to as MCSHAP). We also try complementary sampling (i.e. the sampling of opposed input pairs $(\mathcal{D}'^1, \mathcal{D}'^2)$, where $\mathcal{D}'^1 \cap \mathcal{D}'^2 = \emptyset$ and $\mathcal{D}'^1 \cup \mathcal{D}'^2 = \mathcal{D}$), proposed by Covert & Lee (2021), for kSHAP. For MCSHAP, we try both paired (i.e. complementary input-output pairs in each MC sample) and random Monte-Carlo sampling. We refer to the paired method as PMCSHAP. Algorithm 1 shows the precise procedure. Limiting the number of LLM calls to a fixed number N guarantees that the runtime of this procedure stays comparable to native kSHAP. As shown in §4.1, PMCSHAP leads to a significant improvement of the approximation’s accuracy at an acceptable improvement of reproducibility.

Algorithm 1 (P)MCSHAP

```

Require: Query  $\mathbf{q}$ 
Require: Set of context documents  $\mathcal{D}$ 
Require: Number of perturbations  $N \leq 2^{|\mathcal{D}|}$ 
Require: Number of MC samples  $M$ 
Require: Size of MC samples  $N' < N$ 
 $\mathcal{P} \Leftarrow \{\} \quad \{ \text{create perturbations} \}$ 
while  $|\mathcal{P}| < N$  do
     $\mathcal{D}' \Leftarrow \text{take sample} \subseteq \mathcal{D} \quad \{ \text{paired (PMCSHAP) or random (MCSHAP)} \}$ 
     $\mathbf{x} \Leftarrow \text{create\_prompt}(\mathbf{q}, \mathcal{D}')$ 
     $\mathbf{y} \Leftarrow \text{LLM}(\mathbf{x})$ 
     $\mathcal{P} \Leftarrow \{(\mathbf{x}, \mathbf{y})\} \cup \mathcal{P}$ 
end while
 $\mathcal{A} \Leftarrow \{\} \quad \{ \text{sample attributions} \}$ 
while  $|\mathcal{A}| < M$  do
     $\mathcal{A} \Leftarrow \{\text{kernel\_shap}(\mathcal{P}' \subseteq \mathcal{P} \mid |\mathcal{P}'| = N')\} \cup \mathcal{A}$ 
end while
return  $\frac{1}{M} \sum \mathcal{A} \quad \{ \text{return average attributions} \}$ 

```

Constrained token generation: Both SV and kSHAP were developed for classification scenarios where a single call to the ML model produces a single output \mathbf{y} based on a single input \mathbf{x} (i.e. $\mathbf{y} = f(\mathbf{x})$). However, as mentioned earlier, GEN models in RAG are often autoregressive (L)LMs that iteratively complete a sequence of tokens starting from an initial prompt $\mathbf{x} = [t_0, t_1, \dots, t_n]$. In order to keep the GEN output comparable for different \mathcal{D}' , we first generate the output for the unperturbed set of documents \mathcal{D}

$$t_{i+1}^{\mathcal{D}} = \text{LLM}([t_1^{\mathcal{D}}, \dots, t_n^{\mathcal{D}}, t_{n+1}^{\mathcal{D}}, \dots, t_i^{\mathcal{D}}]) \quad (9)$$

and then constrain the generation output for perturbed sets of documents \mathcal{D}' on the previous *original* output combined with the prompt based on \mathcal{D}' :

$$t_{i+1}^{\mathcal{D}'} = \text{LLM}([t_1^{\mathcal{D}'}, \dots, t_{n'}^{\mathcal{D}'}, t_{n'+1}^{\mathcal{D}'}, \dots, t_i^{\mathcal{D}'}]). \quad (10)$$

This constrained generation is a proven approach in literature (Cohen-Wang et al., 2024; Qi et al., 2024).

3.3. Quantifying Retriever-Generator Agreement

To answer RQ2, we propose and test the *Weighted Attribution-Relevance Gap* (WARG) metric quantifying the agreement between RET and GEN. Our metric is defined on the token level for the query and on the document level for the context documents. In the latter case, recall that \mathcal{D} is the *ordered* set of retrieved documents. The ranking of documents according to RET importance is therefore trivially $\mathcal{R}^{\text{ret}} = \mathcal{D}$. Overall document importance for the GEN is the mean over all output token attributions. The GEN’s ranking is then \mathcal{D} , ordered by descending importance:

$$\mathcal{R}^{\text{gen}} = \text{argsort} \left(\left[\frac{1}{m} \sum_{j=1}^m \beta_{i,j}^{\text{gen}} \mid \forall i \in \{1, 2, \dots, |\mathcal{D}| \} \right] \right) \quad (11)$$

We ground our metric in RBO (Webber et al., 2010), a top-weighted similarity measure used extensively in information retrieval. Unlike correlation coefficients, which are sensitive to conjoint disjointness, RBO provides a “persistence” parameter p controlling weight decay down the ranking list.

Definition 3.1 (Weighted Attribution-Relevance Gap). We define WARG as the complement of the RBO of RET ranking \mathcal{R}^{ret} and GEN ranking \mathcal{R}^{gen} :

$$\begin{aligned} \text{WARG}(p) &= 1 - \text{RBO}(\mathcal{R}^{\text{ret}}, \mathcal{R}^{\text{gen}}; p) \\ &= 1 - (1-p) \sum_{d=1}^k p^{d-1} \cdot \frac{|\mathcal{R}_{1:d}^{\text{ret}} \cap \mathcal{R}_{1:d}^{\text{gen}}|}{d} \end{aligned} \quad (12)$$

where $p \in (0, 1)$ controls the steepness of the weighting.

Selecting $p = 0.5$ implies a strong top-heaviness. In fact, the first rank carries $\approx 50\%$ of the total weight. This is ideal for detecting primacy bias, as rank inversions at the very top (e.g., the GEN attending to a document at rank 1 that the RET placed at rank 3) result in a massive penalty. On the other hand, $p = 0.9$ implies a moderate decay. This is useful for general conformity. By sweeping p , we can perform a sensitivity analysis: if WARG is high at low p but low at high p , the misalignment is concentrated at the very top of the list. This is a hallmark of primacy bias.

4. Experimental Setup

To evaluate the impact of our framework on different RAG architectures, we consider two encoder-only dense RET models and two open-weight GEN models representing the current state-of-the-art in our experiments. For the **retrieval phase**, we use *DRAGON* (Lin et al., 2023), a bi-encoder model built upon the BERT-base architecture (110M parameters), and *Snowflake Arctic Embed 2* (Yu et al., 2025), a single encoder model fine-tuned from the multilingual XLM-R Large (568M parameters). For the **generation phase**, we employ *Llama 3.1 8B* (Grattafiori et al., 2024) and *Gemma 3 12B* (Kamath et al., 2025), which offer a balance between

computational efficiency and reasoning depth. To save resources, we compress the GEN models to `bfloat16`.

4.1. Analysis of Design Choices

To verify the faithfulness of our approach and the validity of our design choices, we carry out a number of small experiments. These are performed on 200 randomly selected samples from the MS-Marco v2.1 Q&A dataset (Nguyen et al., 2016), considering 5 random context documents for each query. The experiments are performed on 8 NVIDIA RTX A5500 GPUs, each with 24GB of memory.

We measure faithfulness as the Area Inside the Perturbation Curves (AIPC) via input perturbation. Since the query texts are naturally short, and perturbing a single token can easily distort the meaning of the whole text independent of the token’s impact on the decision, we test faithfulness only on the context documents. However, as the applied methods are analogous for both contexts and queries, we argue that the results are transferable. Appendix D details the process.

4.2. Experiments

We perform our main analysis on the following two datasets:

TREC CAsT 2019 (TC): a conversational IR benchmark (38,636,520 texts) composed of MS-MARCO (Nguyen et al., 2016), TREC CAR (Dietz et al., 2017), and WAPO, with evaluation topics and human relevance judgments.

FoodSafeSum (FSS): a dataset in the food safety domain with 124k documents and 133 evaluation topics with human-annotated document relevancy; this dataset is not publicly available due to copyright constraints and is therefore a good effort to test the LLMs on previously unseen data. A subset of the dataset was analysed by Bakagianni et al. (2025).

For each dataset, we construct flat FAISS (Douze et al., 2024) indexes over document embeddings and retrieve the top-10 documents per query. The documents are provided to the GEN under two prompt configurations: **(i)** preserving retrieval rank order, and **(ii)** randomly shuffling documents, enabling analysis of sensitivity to document ordering. The experiments conducted on these datasets are performed on an Intel(R) Xeon(R) Platinum 8480CL with 8 NVIDIA H100, each with 80GB of memory.

5. Results

We report the results of our experiments in the following three sections: §5.1 motivates the design choices of our proposed RAG-E framework; §5.2 presents an exploratory analysis of RAG explanations using RAG-E (RQ1); and §5.3 empirically evaluates the utility of our WARG metric (RQ2).

Table 1. Faithfulness in terms of AIPC (*higher is better*) for different baselines Φ^0 of IG (with $L=100$). Bold scores mark the most faithful baseline per RET. 95% confidence intervals (over 1000 bootstrap samples) are reported as [lower, upper].

RET	Baselines (Φ^0)			
	0	[mask]	[unk]	[pad]
DRAGON	0.45 [0.44, 0.46]	0.46 [0.45, 0.47]	0.50 [0.49, 0.51]	0.41 [0.39, 0.42]
Arctic Embed 2	0.68 [0.66, 0.71]	0.61 [0.59, 0.64]	0.73 [0.70, 0.76]	0.67 [0.65, 0.69]

5.1. Empirical Analysis of Design Choices

Retriever Design Choices: To select a useful baseline for the IG-based saliency values for the RET, we compare replacing the embeddings of non-special tokens of the transformer input with the following values (see Fig. 3): **(i)** zeros (discarding the positional embeddings), **(ii)** the [mask] token, **(iii)** the [pad] token (suggested by the IG paper), and **(iv)** the [unk] token, all embedded at the input position. As shown in Tab. 1, a baseline replacing non-special tokens with the model’s [unk] token clearly outperforms the other choices. Further tests of IG on the RET component are reported in Appendix D. Specifically, we find that using $L = 100$ integration steps sufficiently approximates the integral and that IG is more faithful than other explainability methods. Based on these findings, we apply IG with a [unk] baseline and $L = 100$ on RET in this paper.

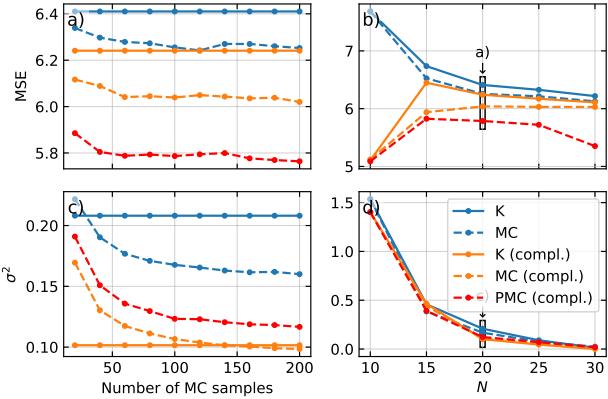


Figure 4. MSE compared to precise SV [plots a) & b)] and variance σ^2 over 10 repetitions [plots c) & d)] of KSHAP, MCSHAP, and PMCSHAP for $|\mathcal{D}| = 5$ (Right column: results for $N = 20$; Left column: results for 200 MC samples).

Generator Design Choices: Fig. 4 presents the approximation error (i.e., MSE), and reproducibility (i.e., variance σ^2) of native KSHAP, MCSHAP, and our proposed PMCSHAP. While the highest stability is achieved by KSHAP with complementary sampling, we find PMCSHAP the better choice, as it achieves the best approximation of SV while displaying sufficiently low variance, especially for high numbers of MC samples and a medium N .

Further experiments reported in Appendix C show that

(P)MCSHAP approximations are statistically significantly closer to the true SV than native KSHAP under identical sampling conditions. Nevertheless, none of the tested approaches outperforms the others in terms of faithfulness (see Appendix D). In conclusion, we apply PMCSHAP with 200 MC samples to explain GEN.

5.2. Exploratory Analysis of RAG Attributions

To answer **RQ1**, we conduct an exploratory analysis of RET and GEN attributions. Overall, we find that even dense RET rely on keyword-matching to retrieve documents while GEN ones show an inherent primacy bias.

Retrieval Analysis: For the retrieval phase, we analyse token-level attributions to understand which parts of the query and documents drive the retrieval score. We aggregated these IG scores across all queries to identify global trends in feature importance. Given that this publication is primarily concerned with explainability, we only report retrieval performance in Appendix E for completeness.

We conduct a Part-of-Speech (POS)-based grammatical analysis of queries and documents to examine how different parts of speech influence the retrieval phase. Our analysis reveals that the retrieval mechanism relies heavily on content-bearing Nouns (NOUN) and Proper Nouns (PROPN), which together make up more than a third of the top-30% attributed words. In comparison, other POS tags make up around 10% or less each (see Appendix G). This indicates that RET matches entities and key concepts rather than structural or function words. This is also supported by qualitative inspection of the attributions (see Appendix I).

We also investigate the role of exact term matching. Our analysis reveals a strong overlap of top-attributed tokens in the retrieved documents and the query terms (see Appendix I). We observed that for the TC dataset, between 65.23% (Arctic Embed 2) and 78.23% (DRAGON) of the query tokens appear among the top 50% attributed tokens in the documents. For the FSS dataset, we observe similarly high overlaps. When calculating the attributions using DRAGON, we have a 78.18% overlap, while for Arctic Embed v2, we observe 71.06% overlapping tokens. These results reinforce our claim that despite the use of dense embeddings, the models still strongly rely on keyword overlap between queries and documents.

Generation Analysis: We test generation with our framework using *Llama 3.1 8B* and *Gemma 3 12B*. We focus on stability of document attribution with respect to ordering.

We compare document attributions under ranked (according to RET) and shuffled document orders in the prompt. Under a generic prompting setup, a document’s importance should depend on its content rather than its position. Note that in

our case, the prompt explicitly specifies the ranked nature of the provided documents, which is expected to influence the attributions assigned by the models (see Appendix B). To measure stability, we compute the mean attribution score for each document included in the prompt across all tested configurations: ranked (according to RET), deduplicated (for FSS), and shuffled document order.

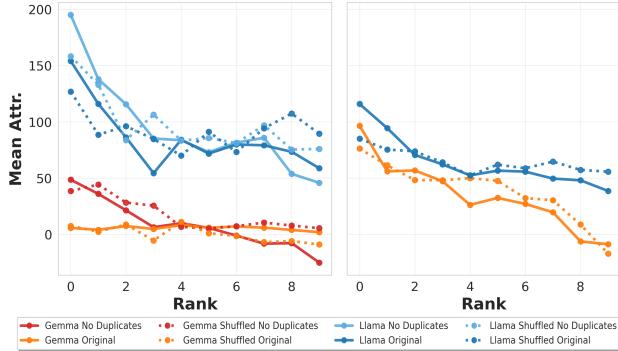


Figure 5. Attribution Instability on FSS (left) and TC (right).

As shown in Fig. 5, both models tend to attribute more to the first documents. While Gemma is more stable on the FSS dataset, Llama exhibits stronger attributions for the top-ranked documents even after shuffling them. Notably, we still see this primacy bias in a small experiment using a prompt not stating a specific ordering of documents (see Appendix B). This suggests an inherent preference of information based on its position in the prompt. When observing the stability in the TC dataset, we have similar curves for the original case and the case with shuffled documents, indicating that preference is not tied to the actual content of the documents. As before, both models tend to prioritize the prompt information regarding the ranked nature of the documents over the actual content, even though attribution decreases on the first documents when shuffled.

Failure Mode Analysis (Wasted Retrieval and Noise Distraction): To analyse consequences of the different inherent behaviours of RET and GEN, indicated by the above results, we quantify two specific failure modes: **(i) Wasted Retrieval** occurs when the top-ranked retrieved document (RET Rank 0) receives GEN Rank > 2 , indicating that the RET’s most relevant document was largely ignored during generation. **(ii) Noise Distraction** occurs when GEN assigns its highest importance (GEN Rank 0) to a document with RET Rank > 2 , suggesting that the model was distracted by content deemed less relevant by RET. The results are shown in Table 2. Specifically, we see *Wasted Retrieval* in 47.4% (Llama on TC) to 66.7% (Gemma on FSS) of the cases, and *Noise Distraction* in 48.1% (Llama on FSS) to 65.9% (Gemma on FSS) of the cases. While shuffling the documents before generation increases these values, removing duplicates reduces them as it focuses attribution to fewer

Table 2. Failure rates quantifying disagreement: “Wasted Retrieval” (ignoring rank 0) and “Noise Distraction” (focusing on rank > 2).

Model	Dataset	Condition	Wasted Ret. (%)	Noise Dist. (%)
Gemma	FSS	Orig	66.7	65.9
Gemma	FSS	ShuffOrig	63.9	75.2
Gemma	FSS	NoDup	64.7	71.4
Gemma	FSS	ShuffNoDup	69.9	69.9
Gemma	TC	Orig	51.4	56.1
Gemma	TC	ShuffOrig	61.8	63.0
Llama	FSS	Orig	53.4	48.1
Llama	FSS	ShuffOrig	64.7	61.7
Llama	FSS	NoDup	48.9	49.6
Llama	FSS	ShuffNoDup	53.4	59.4
Llama	TC	Orig	47.4	49.1
Llama	TC	ShuffOrig	55.5	68.8

documents. These high values motivate further analysis into quantification of the disagreement using WARG.

5.3. Verification of the WARG Metric

To assess the capability of our proposed WARG to quantify the alignment between the RET’s and the GEN’s relevance signals (**RQ2**), we compare it to standard Spearman correlation. As can be seen in Table 3, the alignment between the RET and the GEN is low across all models and datasets. In the standard *Original* condition, Spearman correlation peaks at modest values (0.255 for Gemma on TC) and drops to zero for Gemma on the FSS dataset. This aligns with the high WARG values, indicating high distance between the order provided by RET and attributions of GEN.

We compare *Original* and *No Duplicates* for the FSS dataset. Since this data is composed of regulatory data that may be identical or nearly identical, removing duplicate documents should reduce noise, allowing the model to better focus on important documents. This assumption is supported by the WARG values, which show lower disagreement for both models when compared with the *Original* retrieved documents. Conversely, one would expect that shuffling the order of the documents increases the gap between RET and GEN. This is also visible in Table 3, taking into account values of p that privilege the top documents.

Llama consistently exhibits higher agreement scores with RET than Gemma, with a Spearman correlation of 0.241 against 0.003, obtained on the FSS dataset with no duplicates. This suggests that Llama may be slightly more capable of identifying and attending to relevant content within the context window, whereas Gemma’s attention is either more diffuse or more strictly driven by positional heuristics. This assumption aligns with the fact that while Llama keeps higher WARG values for the shuffled version compared to the correctly ordered documents when varying p , Gemma shows closer values when focusing more on the deep list of retrieved documents, i.e., higher p values.

Table 3. RET-GEN agreement metrics. We report the values with the associated std for the Arctic Embed 2 RET.

Model	Condition	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	Spearman
Gemma (FSS)	No Duplicates	0.774	0.734	0.687	0.655	0.702	0.003
Gemma (FSS)	Original	0.813	0.767	0.711	0.665	0.699	-0.008
Gemma (FSS)	Shuffled No Duplicates	0.784	0.739	0.689	0.655	0.702	0.010
Gemma (FSS)	Shuffled Original	0.809	0.762	0.706	0.661	0.696	-0.010
Gemma (TC)	Original	0.642	0.611	0.575	0.557	0.634	0.255
Gemma (TC)	Shuffled Original	0.724	0.685	0.638	0.606	0.662	0.183
Llama (FSS)	No Duplicates	0.623	0.595	0.565	0.558	0.644	0.241
Llama (FSS)	Original	0.665	0.636	0.602	0.584	0.653	0.117
Llama (FSS)	Shuffled No Duplicates	0.700	0.667	0.631	0.612	0.679	0.075
Llama (FSS)	Shuffled Original	0.750	0.714	0.670	0.637	0.685	0.016
Llama (TC)	Original	0.659	0.626	0.589	0.570	0.642	0.214
Llama (TC)	Shuffled Original	0.764	0.719	0.667	0.628	0.676	0.114

In Appendix H, we report a comparison with the state-of-the-art: ContextCite (Cohen-Wang et al., 2024).

6. Related Work & Discussion

Apart from a short paper by Sudhi et al. (2025), there is, to the best of our knowledge, no peer-reviewed work on end-to-end RAG explainability. The authors propose and test an extrinsic method comparing RET and GEN outputs after leave-one-out perturbations of the inputs and find their explanations plausible in different user studies. However, it remains unclear how exactly the comparison of outputs is done, while evaluation of faithfulness is missing.

Explaining the RET component in isolation has been studied more extensively in literature. Zhuang et al. (2021) propose an *interpretable-by-design* ranking model that could be applied in a RAG context. We argue, however, that limiting oneself to a single RET architecture is not future-proof. Fernando et al. (2019) evaluate intrinsic and extrinsic explainability methods on a single neural retriever, analyzing the resulting top-attributed terms and observing substantial variability across methods, but without assessing their individual faithfulness. In contrast, our RAG-E adopts attribution methods with explicit theoretical guarantees (IG for RET and SV for GEN). Moreover, we improve the reproducibility of SV approximations by proposing PMCSHAP.

Recent work on explaining RAG outputs has largely focused on the GEN component, proposing algorithmic approaches to source attribution rather than relying on model-generated citations alone. Cohen-Wang et al. (2024) introduce ContextCite, which attributes generation to context using a linear surrogate model and can be seen as an extension of LIME (Ribeiro et al., 2016) to RAG. MIRAGE (Qi et al., 2024) identifies context-sensitive tokens by measuring changes in generation probabilities under document removal and applies contrastive attribution scores (Yin & Neubig, 2022). We ground our GEN attribution directly in SV and their kSHAP approximations. For a small number of context documents, this allows us to compute exact, theoretically

founded attributions with manageable computational cost (e.g., one batch of size 64 for $|\mathcal{D}| = 6$). Although attribution precision degrades with larger context sizes (as also the case for ContextCite) existing methods often trade theoretical grounding, and therefore attribution quality, for usability (Lundberg & Lee, 2017).

In summary, existing explainability approaches for RAG systems typically focus on either retrieval or generation in isolation. Our work differs in explicitly targeting the full RAG pipeline and in providing tools to analyse/quantify alignment and information flow between RET and GEN. Specifically, our WARG is, to the best of our knowledge, the first metric assessing RET-GEN alignment enabling diagnosis of whether retrieved evidence is actually used downstream, rather than merely made available.

7. Conclusions

We presented RAG-E, a novel framework for end-to-end explainability and diagnosis of RAG pipelines. We adapted gradient-based IG to explain RETs, establishing suitable baseline embeddings, and provided PMCSHAP, a stabilized variant of KSHAP that enables reliable attribution for autoregressive GEN models. Using RAG-E, we performed exploratory analyses on two RET models, *DRAGON* and *Arctic Embed 2*, and two GEN models, *Llama 3 8B* and *Gemma 3 12B*, to understand which parts of the inputs and outputs are most influential for RET and GEN (RQ1). We observed that while both GEN models exhibit inherent primacy bias, Llama is more influenced by the user prompt when attributing to the individual documents. Gemma tends to value the actual content over the order of the documents provided in the prompt. A grammatical analysis revealed that dense RET models primarily rely on content-bearing nouns, with limited sensitivity to adpositions and syntactic connectors, resulting in behaviour that closely resembles keyword matching between queries and documents. Beyond these component-level insights, our analyses exposed a systematic misalignment between RET and GEN models. This suggests that RAG performance should be understood less as a property of individual components and more as an alignment problem between RET and GEN. To quantify this disagreement, we proposed and evaluated WARG, a metric grounded in the assumption that top-ranked documents best reflect user-relevant information (RQ2).

Overall, RAG-E provides a principled framework for explaining and auditing RAG pipelines by making information flow across components explicit and measurable. Future directions include scalability and extending RAG-E and WARG to include rerankers. Finally, we plan to investigate the use of WARG as an optimization signal, for example, in query rewriting or reinforcement learning-based extensions of RAG systems.

Acknowledgements

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program. Funding for this research has also been provided by the European Union’s Horizon Europe research and innovation programme EFRA (Grant Agreement Number 101093026). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or European Commission-EU. Neither the European Union nor the granting authority can be held responsible for them. 

Impact Statement

RAG systems are increasingly deployed in applications that support decision-making in high-stakes domains such as healthcare and law (Amugongo et al., 2025; Pipitone & Alami, 2024). This work contributes tools for auditing and explaining RAG pipelines by making the interaction between retrieval and generation explicit and measurable. By quantifying retriever-generator misalignment and identifying systematic failure modes, our framework can support more reliable system design, post-hoc analysis, and informed human oversight.

At the same time, we are aware that the methods proposed in this paper rely on a certain level of technical understanding to interpret. Untrained users may misinterpret results, and, for example, confuse low WARG with a direct indication of factual correctness or truthfulness. Misuse of such metrics could lead to overconfidence in aligned but incorrect systems. Moreover, attribution-based explanations may be misunderstood as fully causal by non-expert users, underscoring the importance of careful interpretation and complementary evaluation.

Overall, we view this work as enabling more transparent and accountable use of RAG systems, while recognizing that explainability tools must be applied judiciously and in conjunction with domain expertise and external validation.

References

- Abnar, S. and Zuidema, W. Quantifying attention flow in transformers. In *Proceedings of ACL*, pp. 4190–4197. ACL, 2020.
- Amugongo, L. M., Mascheroni, P., Brooks, S., Doering, S., and Seidel, J. Retrieval augmented generation for large language models in healthcare: A systematic review. *PLOS Digital Health*, 4(6):1–33, 06 2025.
- Angiulli, F., De Luca, F., Fassetti, F., and Nisticò, S. LLiMe: enhancing text classifier explanations with large language models. *Machine Learning*, 114(12):271, 2025.
- Asai, A., Wu, Z., Wang, Y., et al. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of ICLR*, 2024.
- Bakagianni, J., Randl, K., Rocchietti, G., et al. FoodSafe-Sum: Enabling natural language processing applications for food safety document summarization and analysis. In *Findings of EMNLP*, pp. 16786–16804. ACL, 2025.
- Brown, A., Roman, M., and Devereux, B. A systematic literature review of retrieval-augmented generation: Techniques, metrics, and challenges. *Big Data and Cognitive Computing*, 9(12), 2025. ISSN 2504-2289.
- Cohen-Wang, B., Shah, H., Georgiev, K., and Madry, A. ContextCite: attributing model generation to context. In *Advances in NeurIPS*. Curran Associates Inc., 2024.
- Covert, I. and Lee, S.-I. Improving kernelshap: Practical shapley value estimation using linear regression. In *Proceedings of Machine Learning Research*, volume 130, pp. 3457–3465. PMLR, 2021.
- Dietz, L., Verma, M., Radlinski, F., and Craswell, N. TREC complex answer retrieval overview. In *Proceedings of TREC*, pp. 13, 2017.
- Douze, M., Guzhva, A., Deng, C., et al. The faiss library, 2024. Preprint at <https://arxiv.org/abs/2401.08281>.
- Edin, J., Motzfeldt, A. G., Christensen, C. L., et al. Normalized AOPC: Fixing misleading faithfulness metrics for feature attributions explainability. In *Proceedings of ACL*, pp. 1715–1730. ACL, 2025.
- Fernando, Z. T., Singh, J., and Anand, A. A study on the interpretability of neural retrieval models using deepshap. In *Proceedings of SIGIR*, pp. 1005–1008. ACM, 2019.
- Grattafiori, A., Dubey, A., Jauhri, A., et al. The llama 3 herd of models, 2024. Preprint at <https://arxiv.org/abs/2407.21783>. Huggingface model: meta-llama/Llama-3.1-8B.
- Izacard, G., Lewis, P. S. H., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24:251:1–251:43, 2023.
- Jain, S. and Wallace, B. C. Attention is not Explanation. In *Proceedings of NACL*, pp. 3543–3556. ACL, 2019.

- Kamath, A., Ferret, J., Pathak, S., et al. Gemma 3 technical report, 2025. Preprint at <https://arxiv.org/abs/2503.19786>. Huggingface model: google/gemma-3-12b-bit.
- Kuratomi, A., Lee, Z., Miliou, I., Lindgren, T., and Papapetrou, P. ORANGE: Opposite-label soRting for tANGent Explanations in heterogeneous spaces. In *Proceedings of DSAA*, pp. 1–10. IEEE, 2023.
- Lewis, P., Perez, E., Piktus, A., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in NeurIPS*, volume 33, pp. 9459–9474. Curran Associates Inc., 2020.
- Lin, S.-C., Asai, A., Li, M., et al. How to train your DRAGON: Diverse augmentation towards generalizable dense retrieval, 2023. Preprint at <https://arxiv.org/abs/2302.07452>. Huggingface model: facebook/dragon-plus-[context|query]-encoder.
- Liu, N. F., Lin, K., Hewitt, J., et al. Lost in the middle: How language models use long contexts. *TACL*, 12:157–173, 2024.
- Liu, S., Le, F., Chakraborty, S., and Abdelzaher, T. On exploring attention-based explanation for transformer models in text classification. In *Proceedings of Big Data*, pp. 1193–1203, 2021.
- Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in NeurIPS*, volume 30, pp. 4765–4774. Curran Associates, Inc., 2017.
- Mitchell, R., Frank, E., and Holmes, G. GPUtreeShap: massively parallel exact calculation of SHAP scores for tree ensembles. *PeerJ Computer Science*, 8:e880, 2022. doi: 10.7717/peerj-cs.880.
- Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. MS MARCO: A human generated machine reading comprehension dataset. In *CEUR Workshop Proceedings*, volume 1773. CEUR-WS.org, 2016.
- Pipitone, N. and Alami, G. H. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain, 2024. Preprint at <https://arxiv.org/abs/2408.10343>.
- Qi, J., Sarti, G., Fernández, R., and Bisazza, A. Model Internals-based answer attribution for trustworthy Retrieval-Augmented GEneration. In *Proceedings of EMNLP*, pp. 6037–6053. ACL, 2024.
- Rndl, K., Pavlopoulos, J., Henriksson, A., and Lindgren, T. Mind the gap: from plausible to valid self-explanations in large language models. *Machine Learning*, 114(10): 220, 2025.
- Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of KDD*, pp. 1135–1144, 2016.
- Shapley, L. S. 17. a value for n-person games. In Kuhn, H. W. and Tucker, A. W. (eds.), *Contributions to the Theory of Games, Volume II*, pp. 307–318. Princeton University Press, 1953.
- Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *Proceedings of ICML*, volume 70, pp. 3145–3153, 2017.
- Sudhi, V., Bhat, S. R., Rudat, M., et al. Towards end-to-end model-agnostic explanations for rag systems, 2025. Preprint at <https://arxiv.org/abs/2509.07620>.
- Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of ICML*, volume 70, pp. 3319–3328, 2017.
- Tan, Z., Tian, Y., and Li, J. Glime: General, stable and local lime explanation. In *Advances in NeurIPS*, volume 36, pp. 36250–36277. Curran Associates, Inc., 2023.
- Webber, W., Moffat, A., and Zobel, J. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4), 2010.
- Yin, K. and Neubig, G. Interpreting language models with contrastive explanations. In *Proceedings of EMNLP*, pp. 184–198. ACL, 2022.
- Yu, P., Merrick, L., Nuti, G., and Campos, D. F. Arctic-embed 2.0: Multilingual retrieval without compromise. In *Proceedings of COLM*, October 2025. Huggingface model: Snowflake/snowflake-arctic-embed-l-v2.0.
- Zhuang, H., Wang, X., Bendersky, M., et al. Interpretable ranking with generalized additive models. In *Proceedings of WSDM*, 2021.

A. Extended Background

Generally, methods explaining ML models are commonly separated into *intrinsic* methods, deriving explanations based on the internal state of the model, and *extrinsic* or *model agnostic* methods, which statistically infer explanations from input-to-output-relationships without considering model internals.

Early **intrinsic methods** tailored to transformer models were either *attention-based* or *gradient-based* methods. **Attention-based methods** use the fact that the self-attention weights of transformers can be interpreted as weights of how a specific input token impacts the output. They range from simply using the raw attention weights of the last layer as an explanation, to more holistic approaches tracking attention through the whole transformer (Abnar & Zuidema, 2020). Finally, there are also hybrid approaches that consider the gradient of the output with regard to the attention weight (Liu et al., 2021). Nevertheless, Jain & Wallace (2019) criticise at least the use of raw attention weights as explanations to overestimate their correlation to the output. **Gradient-based methods** evaluate the gradient $\frac{\partial y_j}{\partial x_i} \Big|_{x_i=\bar{x}_i}$ to get a linear approximation of the impact of the i^{th} element of the input on the j^{th} element of the output. Simple methods include directly using the gradient as an explanation or multiplying it with the input (Shrikumar et al., 2017). Nevertheless, while optimal in the immediate neighbourhood of \bar{x}_i , the raw gradient is not necessarily a good approximation of the global function learned by the ML model (Shrikumar et al., 2017; Sundararajan et al., 2017, see Figure 6). Sundararajan et al. (2017) solve this problem by integrating gradients over the input dimension by proposing IG.

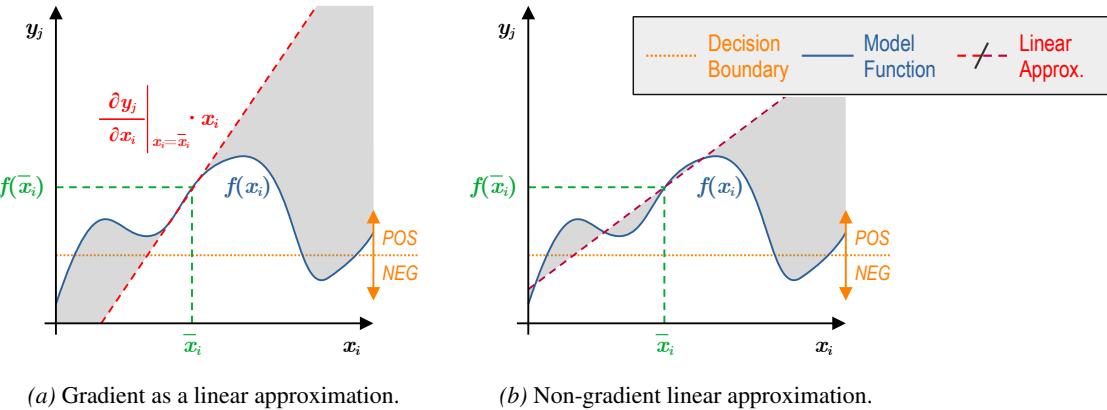


Figure 6. Gradient-based saliency and its caveats. While the gradient is an optimal local linear approximation in the point of the input \bar{x}_i , this is not necessarily the case globally or with regard to the intersect of model function and decision boundary.

Among the most widely used **extrinsic methods** for LMs are Local Interpretable Model-agnostic Explanations (LIME) by Ribeiro et al. (2016) and SHapley Additive exPlanations (SHAP) by Lundberg & Lee (2017). LIME trains a surrogate Logistic Regression classifier on tuples (\bar{x}', \bar{y}') , with \bar{x}' randomly sampled in the close vicinity of \bar{x} , and $\bar{y}' = f(\bar{x}')$ by perturbing elements in \bar{x} . The coefficients of this classifier are used as linear attribution scores. Recent extensions of this method focus mostly on improving the generation \bar{x}' for different types of data (Kuratomi et al., 2023; Tan et al., 2023; Angiulli et al., 2025). κSHAP (Lundberg & Lee, 2017) adapts the LIME approach to approximate SV from game theory (Shapley, 1953) by deriving a loss function for training the surrogate model which satisfies the *additivity*, *missingness*, and *consistency* constraints of SVs. Extensions of the SHAP approach usually focus on creating faster, intrinsic methods that also satisfy these constraints for more specific classifiers (Mitchell et al., 2022). Nevertheless, (Covert & Lee, 2021) test different methods for sampling the input-output pairs necessary to train the surrogate model. They find that sampling complementary pairs of feature sets at the input and their respective outputs improves the stability of κSHAP.

B. Generation Prompts

Fig. 7 shows the formats for the prompts used to generate answers in our experiments and ablations. These correspond to the “create_prompt(q, \mathcal{D})” functions referenced in the main text. In Llama we include the instructions in the system prompt and provide query and context documents in the user prompt. Since Gemma does not support system prompts, we include everything in the user prompt.

SYSTEM: Use the following retrieved documents, ranked from highest to lowest relevance, to answer the user's query.
Be thorough and accurate, and cite documents when useful.
Keep the answer under 200 words.

USER: Document 1: [...]
Document 2: [...]
Document 3: [...]
Document 4: [...]
Document 5: [...]

Query: [...]

MODEL: ...

(a) Llama

USER: Use the following retrieved documents, ranked from highest to lowest relevance, to answer the user's query.
Be thorough and accurate, and cite documents when useful.
Keep the answer under 200 words.

Document 1: [...]
Document 2: [...]
Document 3: [...]
Document 4: [...]
Document 5: [...]

Query: [...]

MODEL: ...

(b) Gemma

Figure 7. Prompt formats for the GEN models. Grayed sections “[...]” are replaced by the respective content.

For the sake of completeness, we selected a subset of the two datasets of around 40% of the queries and tested a generic prompt: *Use the following documents to answer the user's query. Be thorough and accurate, and cite documents when useful. Keep the answer under 200 words.* In this case, we completely removed the information regarding the ordered nature of the documents to observe whether the models would distribute their attribution the same way or not. In Figure 8, we report the attribution instability across the documents.

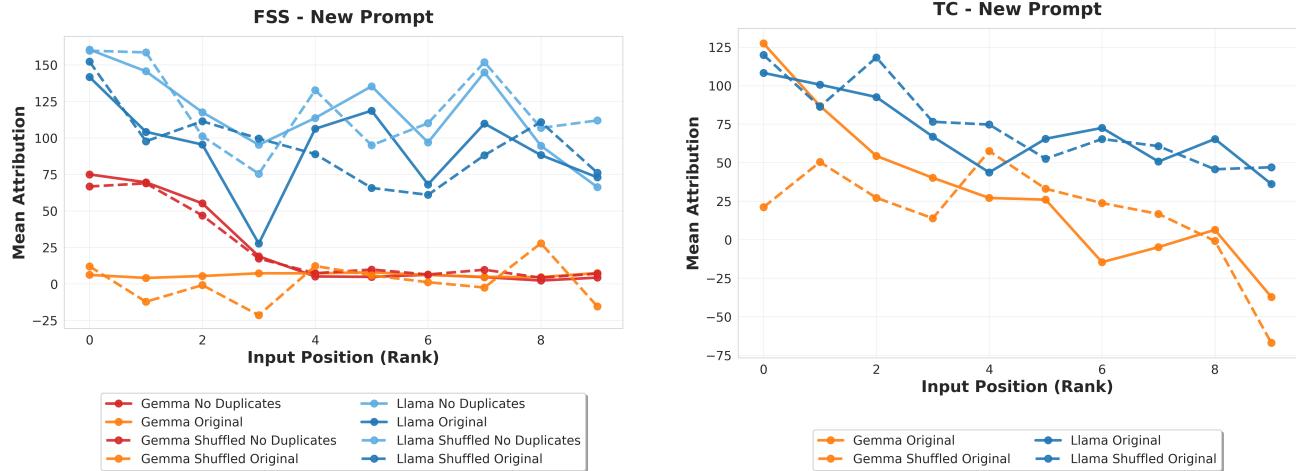


Figure 8. Attribution Instability on the FSS (left) and TC (right) Dataset with the *generic* prompt.

In contrast to the Attribution Instability shown in Figure 5, we can observe here that the attributions given by the GEN are less stable than they were before. Regarding the FSS dataset, we can see that the curve for Gemma with the *original* and *no duplicates* against their shuffled version is very similar, suggesting some kind of primacy bias for the *original* setup, while in the case of *no duplicates*, it seems that the GEN ignores the suggestion given by the RET. When observing Llama, the curves are more unstable. When considering TC, we notice that Gemma has a flatter curve when shuffling the documents, compared to a decreasing curve for the *original* setup. The results obtained by Llama suggest a higher primacy bias since the differences between the curves for the *original* and *shuffled* setups are minimal, with the first being slightly more stable than the second one.

C. Monte-Carlo stabilization for Kernel SHAP

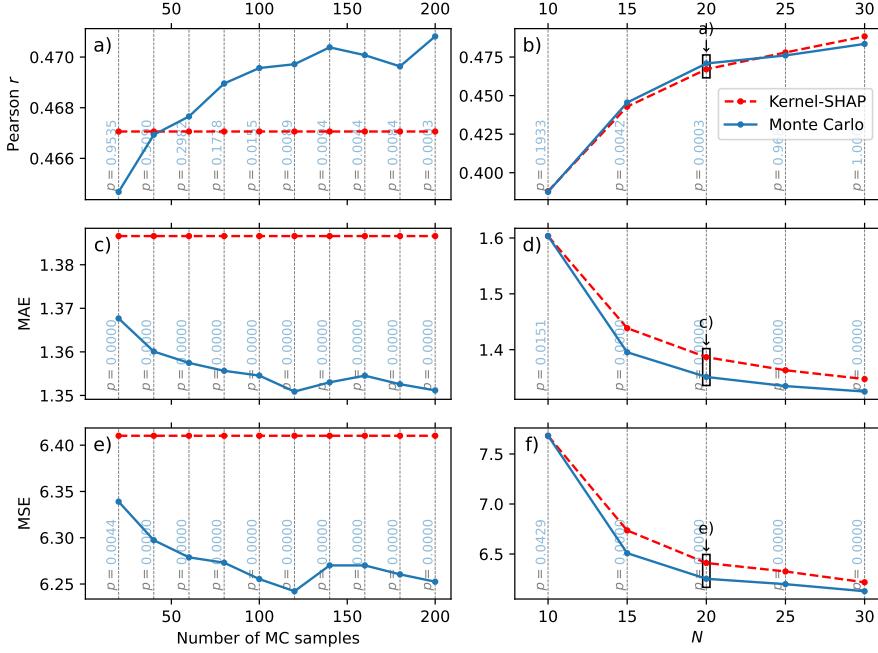


Figure 9. Mean Pearson r , MAE, and MSE comparing kSHAP and MCSHAP to precise SV over 200 random samples of MS-Marco. Computed for $|\mathcal{D}| = 5$ and native **uniform sampling**. The blue vertical numbers are the p -values for a paired Wilcoxon Signed Rank test with alternate hypothesis that the MCSHAP metric is greater than (for Pearson- r) or less than (MAE & MSE) the kSHAP metric (*Right column*: results for $N = 20$; *Left column*: results for 200 MC samples).

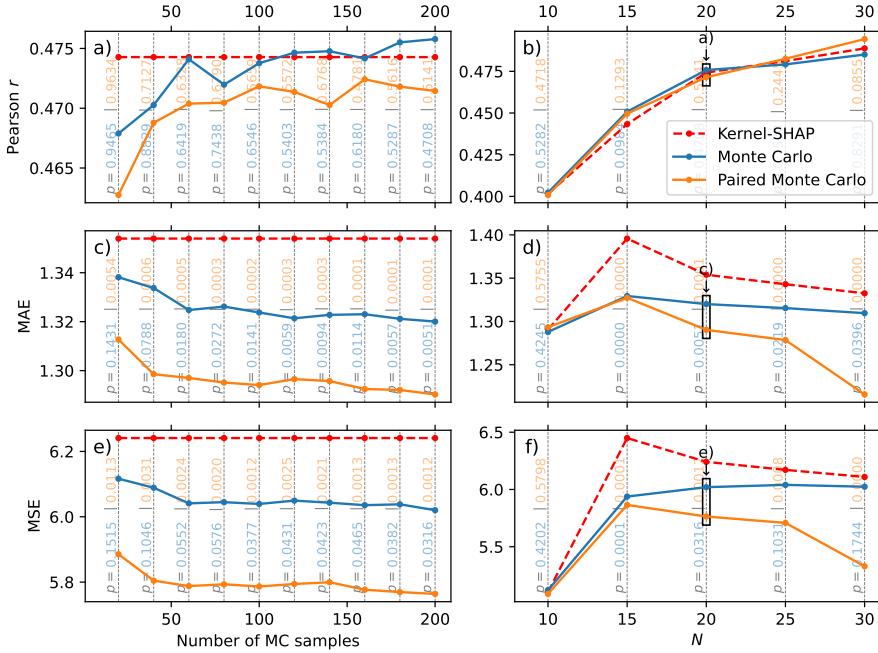


Figure 10. Pearson r , MAE, and MSE comparing KSHAP, MCSHAP, and PMCSHAP over 200 random samples of MS-Marco. Computed for precise SV for $|\mathcal{D}| = 5$ and **complementary sampling**. The blue/orange vertical numbers are the p -values for a paired Wilcoxon Signed Rank test with alternate hypothesis that the (P)MCSHAP metric is greater than (for Pearson- r) or less than (MAE & MSE) the KSHAP metric (*Right column*: results for $N = 20$; *Left column*: results for 200 MC samples).

Figures 9 and 10 show comparisons of kSHAP and (P)MCSHAP methods for the GEN. We present comparisons for both native uniform sampling (Fig. 9) and complementary sampling (Fig. 10). A non-parametric Wilcoxon Signed Rank test shows a significant improvement of uniform MCSHAP and complementary PMCSHAP over kSHAP under identical sampling for ≥ 40 MC samples at $N = 20$ or $15 \leq N \leq 25$ at 200 MC samples. For uniform sampling, we also see significantly better correlation of SV and MCSHAP for these values.

D. Faithfulness Metric

We measure faithfulness as the Area Inside the Perturbation Curves (AIPC) via input perturbation. Concretely, we repeatedly call the model while gradually masking input tokens according to their attributed importance until the input is completely obscured. We perturb the input in both directions, removing tokens Most Relevant First (MoRF) and Least Relevant First (LeRF). A *faithful* explanation should trigger an early change in the MoRF setting, since highly influential tokens are masked first, and only a late change in the LeRF setting, because initially unimportant tokens are removed and should not affect the prediction much.

This metric is widely applied in literature but not standardized (Liu et al., 2021; Edin et al., 2025; Randl et al., 2025). We compute AIPC per input sequence \mathbf{x} , masking it from 0 to all n tokens, one token at a time. Let $\text{mask}_{\text{dir}}(\mathbf{x}, i)$ denote the input \mathbf{x} with i tokens masked according to direction “dir”. We measure the area in between the perturbation curves

$$\begin{aligned} \text{AIPC}_{\text{qry}} = & \left| \int_{i=0}^n g(\text{mask}_{\text{MoRF}}(\mathbf{q}, i), \mathbf{d}) - g(\mathbf{q}, \mathbf{d}) \, di \right| - \\ & \left| \int_{i=0}^n g(\text{mask}_{\text{LeRF}}(\mathbf{q}, i), \mathbf{d}) - g(\mathbf{q}, \mathbf{d}) \, di \right| \end{aligned} \quad (13)$$

for the query and

$$\begin{aligned} \text{AIPC}_{\text{ctx}} = & \left| \int_{i=0}^n g(\mathbf{q}, \mathbf{d}) - g(\mathbf{d}, \text{mask}_{\text{MoRF}}(\mathbf{d}, i)) \, di \right| - \\ & \left| \int_{i=0}^n g(\mathbf{q}, \mathbf{d}) - g(\mathbf{d}, \text{mask}_{\text{LeRF}}(\mathbf{d}, i)) \, di \right| \end{aligned} \quad (14)$$

for context documents. For the RET we choose $g(\mathbf{q}, \mathbf{d}) = s_{\text{ret}}(\mathbf{q}, \mathbf{d})$ and for the GEN $g(\mathbf{q}, \mathbf{d})$ is the model output. We then min-max-normalize per sample and report the mean AIPC over all inputs. Consecutively, AIPC values theoretically range between 0 (not faithful) and 1 (maximally faithful). Note that the realistic maximum is model dependent and always less than 1.

RET Explanations: We compare the faithfulness of our IG RET explanations to the raw gradient $(\text{Grad} : \beta_i = \frac{\partial f(\mathbf{x})}{\partial x_i})$, gradient \times input $(\text{GradIn} : \beta_i = \frac{\partial f(\mathbf{x})}{\partial x_i} \cdot x_i)$ (Shrikumar et al., 2017), and Attention Gradient (AGrad)(Liu et al., 2021) as baselines. As both $e^{\text{qry}}(\cdot)$ and $e^{\text{ctx}}(\cdot)$ are encoder-only transformers $\mathbf{y} = f(\Phi(\mathbf{x}))$ atop an embedding $\Phi(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{p}$ (where \mathbf{p} is a positional embedding), this means we have to also calculate the gradient over $\Phi(\mathbf{x})$ which is not supported by PyTorch’s autograd. We therefore calculate the gradient to the input manually:

$$\frac{\partial e(\mathbf{x})}{\partial \mathbf{x}} = \underbrace{\frac{\partial f(\Phi(\mathbf{x}))}{\partial \Phi(\mathbf{x})}}_{\text{computed by autograd}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} \cdot \mathbf{W} . \quad (15)$$

Since the query texts are naturally short in nature, and perturbing a single token can easily distort the meaning of the whole text, independent of the tokens’ impact on the decision, we test faithfulness only on the context documents. However, as the applied IG method is identical for contexts and query, we argue that the results are transferable. The results of this test are shown in Tab. 4. For both tested RETs, IG performs best. Although, surprisingly, AGrad is equally faithful as IG for the DRAGON RET, IG remains the better choice overall.

To test the design choices we use for the IG explanations, we ablated the number of Riemann integration steps L in terms of faithfulness and additivity ratio. Our results, presented in Tab. 5, suggest that using $L = 100$ steps sufficiently approximates the integral, as the attributions sum up to sufficiently more than 90% of the output, and faithfulness on the context attributions plateaus.

Table 4. RET faithfulness in terms of AIPC (*higher is better*) for different explanation methods. Bold scores mark the most faithful explainability method per RET model. 95% confidence intervals (computed over 1000 bootstrap samples) are reported as [lower, upper].

RET	Methods				
	RND	Grad	GradIn	AGrad	IG
DRAGON	0.00 [-0.01, 0.01]	0.21 [0.19, 0.22]	0.10 [0.09, 0.12]	0.50 [0.49, 0.51]	0.50 [0.49, 0.51]
	0.00 [-0.01, 0.02]	-0.18 [-0.21, -0.15]	0.23 [0.21, 0.25]	0.64 [0.62, 0.66]	0.73 [0.70, 0.76]
Arctic Embed 2	0.00 [-0.01, 0.02]	-0.18 [-0.21, -0.15]	0.23 [0.21, 0.25]	0.64 [0.62, 0.66]	0.73 [0.70, 0.76]

Table 5. Faithfulness in terms of AIPC (*higher is better*) for different integration steps L of IG (Φ^0 fixed to `[pad]`), with additivity ratios reported for query and context. Bold scores mark the most faithful IG setting per RET. 95% confidence intervals (computed over 1000 bootstrap samples) are reported as [lower, upper].

RET / Metric		Integration Steps (L)		
		$L=10$	$L=50$	$L=100$
DRAGON	AIPC	0.40 [0.39, 0.41]	0.41 [0.39, 0.42]	0.41 [0.39, 0.42]
	$\frac{\sum_{i=1}^n \beta_{i,j}}{\bar{y}_j - \bar{y}_j^0}$ query	0.84 [0.02, 1.99]	1.00 [0.93, 1.08]	0.94 [0.88, 0.99]
	context	0.89 [0.88, 0.91]	0.98 [0.97, 0.98]	0.99 [0.99, 0.99]
Arctic	AIPC	0.65 [0.62, 0.67]	0.67 [0.65, 0.69]	0.67 [0.65, 0.69]
Embed 2	$\frac{\sum_{i=1}^n \beta_{i,j}}{\bar{y}_j - \bar{y}_j^0}$ query	0.91 [0.87, 0.96]	0.98 [0.98, 0.99]	0.99 [0.99, 0.99]
	context	0.77 [0.65, 0.86]	0.99 [0.90, 1.12]	0.98 [0.95, 1.01]

GEN Explanations: In order to verify the faithfulness of our GEN explanations, we compute AIPC for native kSHAP, MCSHAP, and our proposed PMCSHAP for native uniform sampling as well as complementary sampling (Covert & Lee, 2021) over different *sample sizes* N and report the results in Tab. 6. The results show comparable faithfulness overall.

Table 6. Faithfulness in terms of AIPC (*higher is better*) for different kSHAP extensions and $|\mathcal{D}| = 5$. 95% confidence intervals (computed over 1000 bootstrap samples) are reported as [lower, upper]. The columns “Random” and “Precise” show the faithfulness of randomized and precise SV attributions, and signify theoretical bounds for faithfulness.

Method	Sampling	Random	AIPC					Precise
			$N=10$	$N=15$	$N=20$	$N=25$	$N=30$	
kSHAP	unif.		0.51 [0.49, 0.53]	0.55 [0.53, 0.57]	0.56 [0.54, 0.59]	0.57 [0.55, 0.60]	0.58 [0.56, 0.60]	
MCSHAP	unif.		0.51 [0.49, 0.53]	0.55 [0.53, 0.57]	0.56 [0.54, 0.59]	0.57 [0.55, 0.60]	0.58 [0.56, 0.60]	
kSHAP	compl.	0.00 [-0.01, 0.01]	0.49 [0.47, 0.51]	0.55 [0.52, 0.57]	0.57 [0.55, 0.60]	0.58 [0.55, 0.60]	0.58 [0.56, 0.61]	0.58 [0.56, 0.60]
MCSHAP	compl.		0.50 [0.48, 0.52]	0.54 [0.52, 0.57]	0.57 [0.55, 0.59]	0.57 [0.55, 0.60]	0.58 [0.56, 0.60]	
PMCSHAP	compl.		0.49 [0.47, 0.51]	0.55 [0.53, 0.57]	0.57 [0.54, 0.59]	0.57 [0.55, 0.60]	0.58 [0.56, 0.60]	

E. Retrieval Performance

To validate our retrieval component, we evaluated the performance of the *DRAGON* and *Arctic Embed 2* embedders on the FSS and TC datasets. In Tab. 7 we report the NDCG@3, and Precision@1.

Table 7. Retrieval Performances. We report here Precision@1 (P@1) and Normalize Discounted Cumulative Gain for the first three documents (NDCG@3)

Configuration	P@1	NDCG@3
FSS (Arctic Embed 2)	0.221	0.155
FSS (DRAGON)	0.183	0.116
TC (Arctic Embed 2)	0.723	0.475
TC (DRAGON)	0.699	0.441

In this phase, we report only fine-grained metrics given the fact that our goal is not to evaluate the retrieval phase but to assess the alignment between the RET and the GEN.

F. Results with DRAGON RET

Table 8. WARG-Rank Sensitivity to p and Spearman Correlation (Dragon Retriever)

Model	Condition	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	Spearman
Gemma (FSS)	No Duplicates	0.778	0.733	0.681	0.641	0.685	0.055
Gemma (FSS)	Original	0.804	0.761	0.708	0.664	0.700	-0.024
Gemma (TREC)	Original	0.655	0.617	0.576	0.554	0.630	0.292
Gemma (TREC)	Shuffled Original	0.728	0.683	0.633	0.599	0.656	0.215
Llama (FSS)	No Duplicates	0.638	0.613	0.583	0.569	0.645	0.170
Llama (FSS)	Original	0.647	0.616	0.581	0.565	0.640	0.190
Llama (FSS)	Shuffled No Duplicates	0.701	0.667	0.627	0.603	0.665	0.082
Llama (FSS)	Shuffled Original	0.740	0.701	0.654	0.620	0.672	0.106
Llama (TREC)	Original	0.633	0.601	0.567	0.552	0.631	0.234
Llama (TREC)	Shuffled Original	0.697	0.664	0.625	0.600	0.661	0.126

G. POS Analysis

In this section we report some analysis regarding the attribution given by the models to the different part of speech. We analysed the distribution of attribution weights across Part-of-Speech (POS) tags and POS bigrams (word pairs) for both GENS and RETs, focusing on the top 30% of attributed tokens (Figures 11 and 12). Our analysis reveals several key similarities and distinctions in how these models attend to linguistic features. To be able to compute so, we first reconstructed the words from the tokens and successively summed the attributions outputted by the models.

Across all four models, NOUNs consistently receive the highest attribution mass. This indicates a shared reliance on substantive content words as the primary elements of meaning and relevance. Furthermore, NOUN + NOUN is overall the most significant bigram, which highlights the importance of nouns and nominal phrases in both generation and retrieval contexts. While both GENS prioritize nouns, their secondary focuses diverge significantly. Gemma exhibits a high sensitivity to punctuation and structural markers. PUNCT is the second most attributed tag, and NOUN + PUNCT values are prominent. This might suggest that Gemma’s attention mechanism relies heavily on sentence boundaries and delimiters to organize context. On the other hand, Llama displays a more syntactic focus, with high attribution to VERBs and function words like ADP (adpositions). Its bigrams, such as NOUN + ADP and DET + NOUN, reflect a deeper engagement with the grammatical structure and prepositional relationships within the text, rather than just isolated keywords or delimiters.

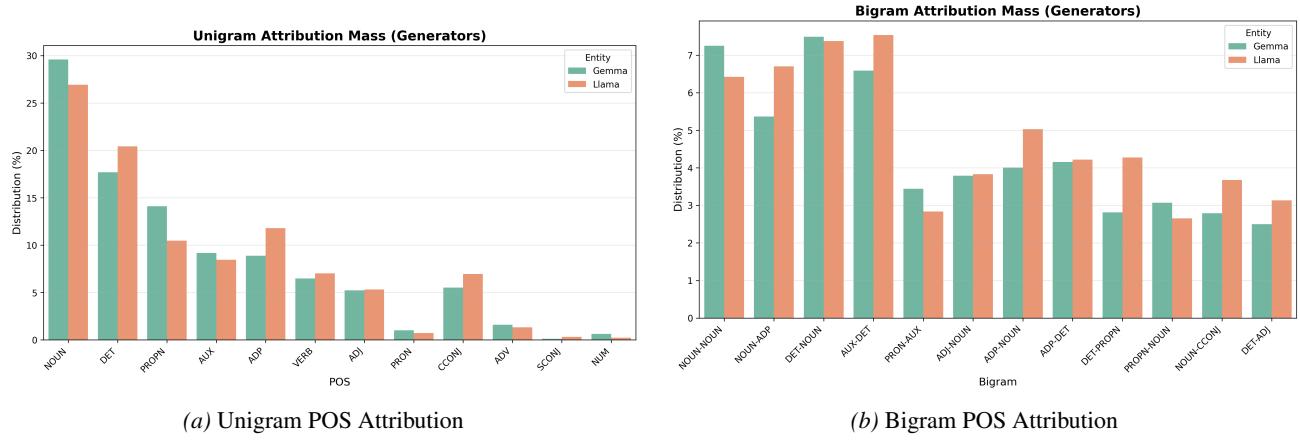
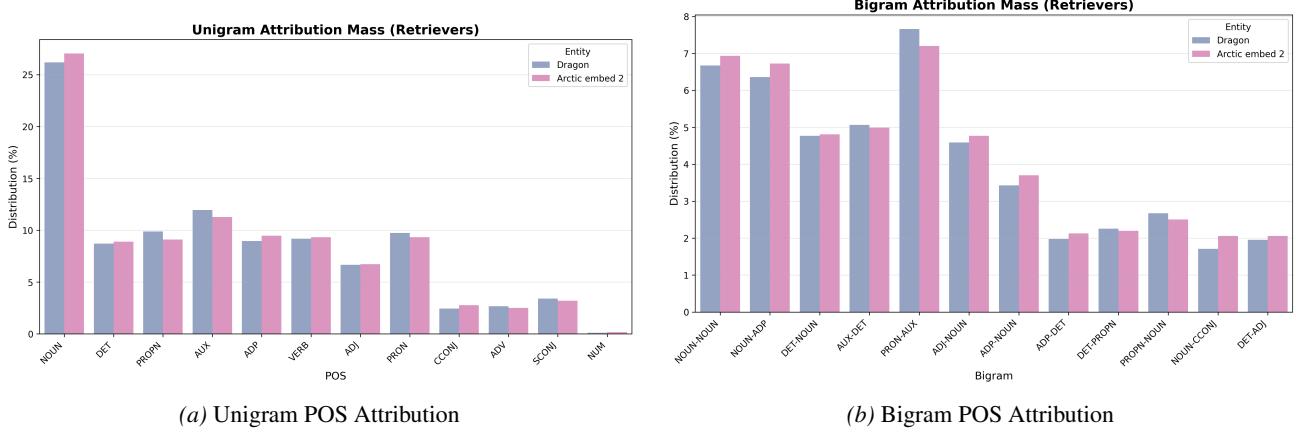


Figure 11. Unigram and Bigram Attribution Mass for LLama and Gemma

RET Comparison: Snowflake vs. Dragon The RETs show a stronger alignment with entity-centric processing compared to the GENS, but with nuanced differences. Arctic-Embed (Snowflake) shows a distinct preference for PROPN (Proper Nouns), which ranks second only to common nouns. Its top bigrams, like PROPN + NOUN, suggest it functions closer to a traditional keyword-based RET, heavily prioritizing named entities and specific terminology to match queries. Dragon (RoBERTa-based) shares the entity focus (high PROPN) but also places significant weight on ADJ (Adjectives) and PRON (Pronouns). This suggests a more semantic understanding of the query, attending to descriptive qualifiers and anaphoric references (e.g., PRON + AUX) which are typical of dense retrieval models that capture “meaning” beyond exact keyword matches.


 Figure 12. POS attribution for *Arctic Embed 2* and *Dragon*.

GEN vs. RET A broad distinction emerges between the two classes of models. On the one hand, GENS (especially Llama) distribute attention more broadly across syntactic elements (VERB, ADP, DET) required to construct coherent sentences and follow narrative flow. On the other hand, RETS concentrate their mass more narrowly on content-bearing classes (NOUN, PROPN, ADJ), filtering out much of the connectors to maximize relevance attribution.

H. Comparison with ContextCite

We compared our perturbation-based attribution method with ContextCite, an attention-based attribution baseline. Table 9 summarizes the agreement (WARG-0.9) and correlation (Spearman) metrics for both methods.

Model (GEN)	Dataset	Our Method (Perturb)		ContextCite (Attn)	
		WARG-0.9 ↓	Spearman ↑	WARG-0.9 ↓	Spearman ↑
Gemma	FSS	0.75	0.01	0.12	-0.04
Gemma	TC	0.72	0.20	0.67	-0.10
Llama	FSS	0.72	0.19	1.65	-0.06
Llama	TC	0.71	0.27	0.62	0.09

Table 9. Comparison of RET-GEN Agreement between our Perturbation-based method and ContextCite. WARG-0.9 (Weighted Agreement Rank Gap) favors lower values; Spearman correlation favors higher values.

The results reveal distinct behaviors. On the FSS dataset with Gemma, ContextCite yields an exceptionally low WARG (0.12), implying the attention distribution closely mirrors the retrieval ranking mass. However, this mechanical alignment does not translate to rank correlation (Spearman ≈ 0), suggesting the weights match but the specific ordering might not. Conversely, on Llama with FSS, ContextCite shows very high disagreement (WARG 1.65).

Our perturbation-based method produces more consistent agreement scores ($\text{WARG} \approx 0.7$) across models and datasets, and generally higher Spearman correlations, particularly on TC (0.20–0.27). This suggests that while attention mechanisms (ContextCite) may sometimes align linearly with input order (due to positional bias in attention), perturbation-based causal analysis better captures the relationship between retrieval relevance and generation utility. The high rates of “noise distraction” detected by our method reflect this causal divergence: the GEN often functionally relies on documents that the RET (and potentially simple attention maps) did not prioritize.

I. Comparative Analysis of Attribution Patterns

In this section, we showcase two examples for each dataset, highlighting the extremes of our WARG attribution for both datasets. We report, for every query, the attribution for the query and the retrieved documents, as well as the impact of each document on the generated text (*Assistant*). In Table 10 we also show the percentage of query tokens that appear among the top 50% most attributed tokens as discussed in §5.

Table 10. Percentage of query tokens appearing among the top 50% attributed tokens in retrieved documents.

Dataset	Model (RET)	Query Overlap (%)
TC	Arctic Embed 2	65.23%
	DRAGON	78.23%
FSS	Arctic Embed 2	71.06%
	DRAGON	78.18%

I.1. Case Study 1: Low Agreement

In Figures 13 and 14, we show the attribution patterns for two queries that exhibited low agreement between the retriever and the generator.

As we can observe, in both cases, the most meaningful words in the query are also among the most attributed parts of the documents (*food, flavor*, and *agents* for the FSS example; and *pop, music*, and *education* for the TC example). Interestingly enough, the RET also highly attributes semantically related words (e.g., *lesson based, popular, pedagogy*).

When observing the GEN attributions, we can see that all of the documents contributed to creating the answer when considering Figure 13 with Documents 9 and 10 contributing more than some would expect from the retriever rank. When focusing on the TC example, we clearly see that documents 5,6,7, and 9 contribute the most to the generated text in comparison to the first ranked documents. This could be due to the similarity (complete overlapping in some case) among the retrieved documents

I.2. Case Study 2: High Agreement

Similarly to Case Study 1, in Figures 15 and 16, we show the attributions for both the RET and the GEN. Similarly to what we observed before for the RET side, we observe strong attributions for words that appear in the query. This is particularly evident in figure 16 where the query is short and contains less meaningful words.

When observing the attributions for the generated texts, we can observe again a similar behaviour. When looking at the results computed on the FSS dataset, we show a higher variance in attribution, but we can see that many of the most attributed words (intense color) belong to the first documents. On the other hand, when looking at Figure 16 regarding the TC dataset, we can clearly see that Documents 1 and 3 are among the most attributed.

Query: What are the different types of flavor enhancers and their functions in food products, and how do they compare to other food additives like emulsifiers and bulking agents?

- Document 1:** "...However, substances should not be considered as food additives when they are used for the purpose of imparting flavour and/or taste or for nutritional purposes, such as salt replacers, vitamins and minerals..." Finally, Article

- Document 2:** "...However, substances should not be considered as food additives when they are used for the purpose of imparting flavour and/or taste or for nutritional purposes, such as salt replacers, vitamins and minerals..." Finally, Article

- Document 3:** (5) Curing and pickling agents : Substances imparting a unique flavor and/or color to a food, usually producing an increase in shelf life stability.
(6) Dough strengtheners : Substances used to modify starch and gluten.

- Document 4:** (5) Curing and pickling agents : Substances imparting a unique flavor and/or color to a food, usually producing an increase in shelf life stability.
(6) Dough strengtheners : Substances used to modify starch and gluten.

- Document 5:** (10) **Firming agents:** Substances added to precipitate residual pectin, thus strengthening the supporting tissue and preventing its collapse during processing.
(11) **Flavor enhancers:** Substances added to supplement, enhance, or modify the original taste and/or aroma

- Document 6:** — Strength boosters — Firming agents — Substances that create or maintain hardness and crispness of fruit and vegetable tissues, or that work together with coagulants to create or strengthen a clot — — —

- Document 7:** * For example, when neohesperidin DC (which at high concentrations tastes sweet) is added to a flavouring which is then added to a food, it is able to increase specific characteristics, such as the perceived

- Document 8:** For example, when neohesperidine DC (which at high concentrations tastes sweet) is added to a flavouring which is then added to a food, it is able to increase specific characteristics, such as the perceived

- Document 9:** — A food additive, which adds or Color, decorative pigment, surface colorant

- Document 10:** acids — Acids — Substances that increase the acidity of a food product or give it a sour taste — Acidity regulators — Acidity regulators —

Assistant: Flavor enhancers are substances added to food to supplement, enhance, or modify the original taste and/or aroma (Document 5, 9, 10). They work by amplifying existing tastes and smells, increasing the overall flavor perception, or even balancing flavor profiles (Document 1). For example, monosodium glutamate (E 621) enhances the umami taste in foods like soups and sauces (Document 1). Unlike imparting a characteristic taste of their own, they intensify what's already present. Substances that modify sweetness, sourness, or saltiness can also be considered flavor enhancers. Compared to other food additives, flavor enhancers differ in their primary function. Emulsifiers (Document 9, 10) create or stabilize mixtures of otherwise unmixable substances like oil and water, while bulking agents increase the volume of a food product without significantly affecting its calorie count (Document 10). Essentially, flavor enhancers focus on taste modification, while emulsifiers address texture and consistency, and bulking agents focus on volume.

<i>Legend</i>	<i>Doc. 1</i> (-9%)	<i>Doc. 2</i> (-26%)	<i>Doc. 3</i> (-16%)	<i>Doc. 4</i> (-2%)	<i>Doc. 5</i> (-10%)	<i>Doc. 6</i> (-1%)	<i>Doc. 7</i> (+14%)	<i>Doc. 8</i> (+8%)	<i>Doc. 9</i> (-2%)	<i>Doc. 10</i> (+14%)
---------------	------------------------	-------------------------	-------------------------	------------------------	-------------------------	------------------------	-------------------------	------------------------	------------------------	--------------------------

Figure 13. Visual example of token-level attributions for FSS (FoodSafety) query. Retriever: Snowflake Arctic Embed, Generator: Gemma 3.12B.

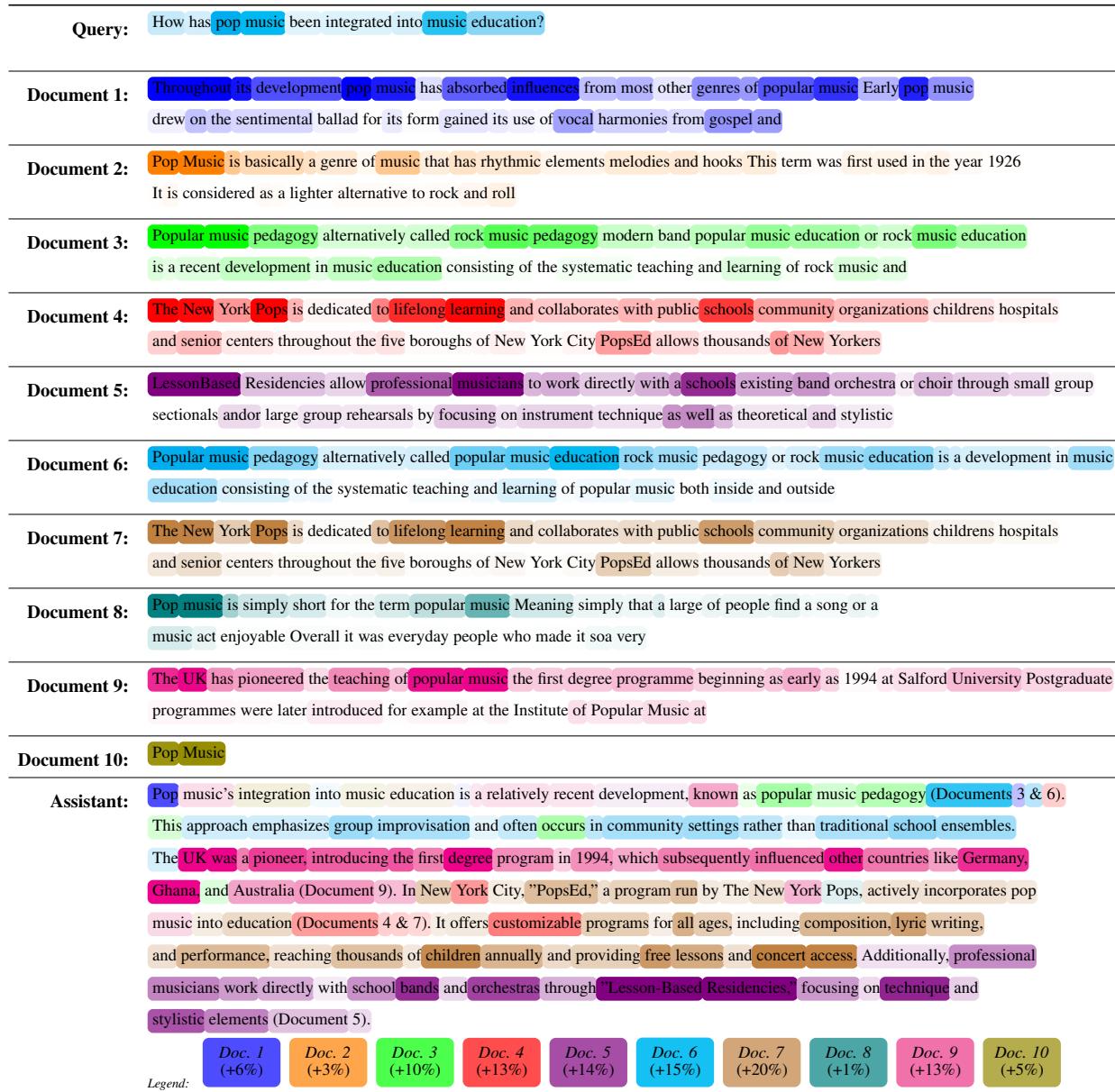


Figure 14. Visual example of token-level attributions for TREC query. Retriever: Snowflake Arctic Embed, Generator: Gemma 3 12B.



Figure 15. Visual example of token-level attributions for FSS Low WARG query. Retriever: Snowflake Arctic Embed, Generator: Gemma 3 12B.

Query:	What are lung cancer's symptoms?										
Document 1:	The most common symptoms of lung cancer are 1 A cough that does not go away or gets worse 2 Chest pain that is often worse with deep breathing coughing or laughing 3 Hoarseness 4										
Document 2:	The most common symptoms of lung cancer are 1 A cough that does not go away or gets worse 2 Chest pain that is often worse with deep breathing coughing or laughing 3 Hoarseness 4										
Document 3:	The most common symptoms of lung cancer are 1 A cough that does not go away or gets worse 2 Coughing up blood or rustcolored sputum spit or phlegm 3 Chest pain that is often										
Document 4:	The most common lung cancer symptoms include 1 Appetite loss 2 Fatigue 3 Weight loss 4 A persistent cough 5 A change in a long standing cough 6 Breathlessness or shortness of breath 7 Coughing										
Document 5:	The most common lung cancer symptoms include 1 Appetite loss 2 Fatigue 3 Weight loss A persistent 1 cough A change in a long standing cough Breathlessness or shortness of 1 breath Coughing up blood										
Document 6:	Common symptoms of lung cancer include 1 A cough that doesnt go away and gets worse over time 2 Constant chest pain 3 Coughing up blood Shortness of breath wheezing or 1 hoarseness Repeated problems										
Document 7:	Signs and symptoms of lung cancer may include 1 A new cough that doesnt go away 2 Changes in a chronic cough or smokers cough 3 Coughing up blood even a small amount Shortness of										
Document 8:	Signs and symptoms of lung cancer typically occur only when the disease is advanced Signs and symptoms of lung cancer may include A new cough that doesnt go away Changes in a chronic cough or										
Document 9:	Signs and symptoms of lung cancer typically occur only when the disease is advanced Signs and symptoms of lung cancer may include A new cough that doesnt go away Coughing up blood even a small										
Document 10:	Common symptoms of lung cancer include A cough that doesnt go away and gets worse over time Constant chest pain Coughing up blood Shortness of breath wheezing or hoarseness Repeated problems with pneumonia or bronchitis										
Assistant:	Lung cancer symptoms can vary, but several are commonly reported. A persistent or worsening cough is a primary indicator (Documents 1, 2, 3, 4, 6, 7, 8, 9, 10). Many sources also mention chest pain, often exacerbated by breathing, coughing, or laughing (Documents 1, 2, 3, 6, 8). Other frequent symptoms include hoarseness (Documents 1, 2, 3, 6, 7), weight loss, and loss of appetite (Documents 1, 2, 3, 4, 5, 6, 10). Coughing up blood or rust-colored sputum is a concerning sign (Documents 2, 4, 5, 7, 10). Shortness of breath or breathlessness is also common (Documents 3, 4, 5, 7, 8, 10). Finally, fatigue, aches or pains when breathing/coughing, and repeated chest infections can occur (Documents 4, 5, 6, 7, 8). **Please note:** Symptoms often appear when the disease is advanced (Documents 8,										
<i>Legend:</i>	<table style="width: 100%; text-align: center;"> <tr> <td>Doc. 1 (+16%)</td> <td>Doc. 2 (+12%)</td> <td>Doc. 3 (+14%)</td> <td>Doc. 4 (+9%)</td> <td>Doc. 5 (+13%)</td> <td>Doc. 6 (+2%)</td> <td>Doc. 7 (+8%)</td> <td>Doc. 8 (+12%)</td> <td>Doc. 9 (+7%)</td> <td>Doc. 10 (+7%)</td> </tr> </table>	Doc. 1 (+16%)	Doc. 2 (+12%)	Doc. 3 (+14%)	Doc. 4 (+9%)	Doc. 5 (+13%)	Doc. 6 (+2%)	Doc. 7 (+8%)	Doc. 8 (+12%)	Doc. 9 (+7%)	Doc. 10 (+7%)
Doc. 1 (+16%)	Doc. 2 (+12%)	Doc. 3 (+14%)	Doc. 4 (+9%)	Doc. 5 (+13%)	Doc. 6 (+2%)	Doc. 7 (+8%)	Doc. 8 (+12%)	Doc. 9 (+7%)	Doc. 10 (+7%)		

Figure 16. Visual example of token-level attributions for TREC Low WARG query. Retriever: Snowflake Arctic Embed, Generator: Gemma 3 12B.