Grammar Bias: Neurosymbolic Embeddings for Directionally Robust KG-RAG

Victor Jesus Sotelo Chico[1,*,†], André Gomes Regino[2,3,4] and Julio Cesar dos Reis[1]

[1]Institute of Computing, University of Campinas, Campinas, São Paulo, Brazil
[2]Center for Information Technology Renato Archer, Campinas, São Paulo, Brazil
[3]Centro Universitário de Jaguariúna (UniFAJ), Jaguariúna, São Paulo, Brazil
[4]Centro Universitário Max Planck (UniMAX), Indaiatuba, São Paulo, Brazil

Abstract

Knowledge graph triples are symbolic structures, but when embedded for retrieval, they are treated as plain text. Dense encoders learn to match shared words between the text and the triple, rather than exploiting the triple's explicit symbolic structure (subject, predicate, object). This grammar bias stays hidden when retrieving a triple from a text query, since triple tokens often appear verbatim in the source sentence. Retrieving a text from a triple query exposes the encoder's need for genuine relational understanding, since the triple is symbolic and shares no surface vocabulary with candidate texts. We propose a neurosymbolic approach: Masked Triple Modeling (MTM) to address this gap. MTM refers to an auxiliary fine-tuning task that trains sparse encoders to recognize triple structure and slot ordering, independent of surface wording. Combined with sparse, SPLADE-style embeddings, MTM trains the encoder to distinguish semantically valid triples from corrupted ones based on their relational structure rather than their grammar. Experimental results on the Portuguese version of Webnlg show that MTM raises direction-flip robustness by two orders of magnitude over the dense baseline. This comes with a measurable, though not drastic, cost to retrieval ranking. IR NDCG@1 drops from the dense baseline's 0.966 to 0.870 without direction-flip hard negatives, or to 0.630 when hard negatives are used. Earlier structural objectives nearly collapsed retrieval (NDCG@1 ≈ 0.01) to close the same gap. These results indicate that grammar bias and directional robustness can be decoupled from retrieval effectiveness, positioning neurosymbolic sparse embeddings as a building block for more adequate and semantically grounded retrieval in KG-driven RAG pipelines.

Keywords

Sparse Embeddings, Neurosymbolic AI, Retrieval-Augmented Generation (RAG), Grammar Bias

CEUR-WS.org/Vol-4312/paper29.pdf

1. Introduction

Nowadays, knowledge graph (KG) symbolic representations capture human concepts and rules through triple facts. Recent advances in artificial intelligence have increased the demand to consume multiple types of data, ranging from structured to unstructured, with knowledge graphs and text standing out as a prominent example of this kind of interaction. Prior research has shown that combining them is possible by using encoder models to embed triples and text into a shared vector space [1, 2]. Another explored strategy introduces an intermediate step, transforming triples into natural text and then embedding this verbalization with an encoder model [3].

In previous work, we proposed an alignment over fine-tuning pretrained models [4]. Our findings showed that recovering triples from text, or vice versa, was possible using pretrained models. However, we attribute these effects to a shared grammatical bias between text and triples. For example, given the triple (Alan_Bean, birthPlace, Wheeler_Texas) verbalized as "Alan Bean was born in Wheeler, Texas," the subject and object strings appear almost verbatim in both the triple and the text; and the predicate *birthPlace* maps to a fixed, predictable phrase ("was born in"). A model can, therefore, recover the

RAGE-KG 2026: 3rd International Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs, co-located with ISWC 2026, October 25–26, 2026, Bari, Italy
*Corresponding author.
v265173@dac.unicamp.br (V. J. S. Chico); aregino@cti.gov.br (A. G. Regino); jreis@ic.unicamp.br (J. C. d. Reis)
0000-0001-9245-8753 (V. J. S. Chico); 0000-0001-9814-1482 (A. G. Regino); 0000-0002-9545-2098 (J. C. d. Reis)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-04

correct triple–text pairing by matching overlapping surface tokens and the canonical predicate-to-phrase mapping, without ever needing to represent which slot is head and which is tail. We call this lexical shortcut *grammar bias*.

Despite advancements, current approaches to triple-text alignment do not account for the relation-flip problem, in which the order of subject and object determines the semantic meaning of an assertion [5]. For instance, the triple (Barack_Obama, parentOf, Malia_Obama) does not carry the same meaning as its flipped counterpart (Malia_Obama, parentOf, Barack_Obama) — yet an encoder relying on grammar bias alone cannot differentiate them. Because both triples share identical tokens and differ only in slot order, a signal invisible to lexical overlap. As a result, the encoder may retrieve or rank the flipped, factually false triple as confidently as the correct one, producing a fluent but structurally hallucinated assertion. Unlike conventional hallucination, where content is fabricated, here the entities and relation are real; only their roles are swapped.

To address this gap, we propose a neurosymbolic approach that forces the encoder to learn relational structure independently of surface form. We introduce Masked Triple Modeling (MTM), in which one of the three triple slots — subject, predicate, or object — is hidden and the model predicts the missing element from the remaining two. This prediction occurs entirely within the triple space, with no verbalized text involved, so matching shared surface tokens between the text and the triple is unavailable. Success requires the encoder to represent which slot each entity occupies and how it relates to the others. We combine this objective with sparse, SPLADE-style embeddings [6] and fine-tune the encoder on both tasks, weighting a triple-text alignment loss and the MTM loss by a factor λ. In this sense, the resulting representation is shaped as much by relational structure as by alignment with text.

To support reproducibility, we make our code publicly available, including training scripts with their respective hyperparameters, evaluation pipelines, and the full set of ablation configurations used in this work, at the repository[1]

Our experiments conducted on distinct datasets show that this structural learning objective produces measurable improvements, particularly where grammar bias is most harmful: sensitivity to relation direction. While this enhancement slightly decreases retrieval ranking effectiveness, it shows that directional robustness and retrieval effectiveness can coexist. By explicitly modeling triple structures, MTM promotes representations that depend less on lexical shortcuts and more on the symbolic semantics contained in KG triples.

The remainder of this article is organized as follows: Section 2 reviews related work on triple-text alignment and retrieval-augmented approaches over KGs; Section 3 presents the problem definition introducing the Grammar Bias problem and present the encoder families; Section 4 presents our neurosymbolic proposal, Masked Triple Modeling (MTM), combined with sparse embeddings; Section 5 reports our experimental methodology; Section 6 reports the obtained results; Section 7 discusses the implications and limitations of our findings; and Section 8 concludes and outlines directions for future work.

2. Related Work

Direct Fact Retrieval (DiFaR) [1] collapses the conventional three-stage fact retrieval pipeline — entity mention detection, entity disambiguation, and relation classification — into a single dense encoder. The encoder maps natural language queries and linearized triples into a shared embedding space, trained with a standard contrastive objective over (text, triple) pairs. Despite requiring no entity-specific or relation-specific supervision, DiFaR outperforms pipeline baselines across four QA tasks and dialogue benchmarks and generalizes zero-shot across knowledge graphs (Wikidata to Freebase) without additional training. However, the examples reported in the paper show the query's entity mention appearing verbatim in the retrieved triple, and the work includes no evaluation of robustness to argument order or to any form of corruption. This leaves open whether its retrieval accuracy indicates true relational understanding or lexical overlap between the query and the triple.

[1] https://github.com/Visot/grammar-bias

Knowledge-Augmented language model PromptING (KAPING) [2] filters knowledge graph facts for LLM prompt injection using off-the-shelf sentence embedding models. Before encoding, triples are converted from the symbolic (s, r, o) form into text via linear verbalization by concatenating the subject, relation, and object into a single string. The question and verbalized triples are then projected into the same representation space using pretrained encoders such as MPNet, with an additional comparison against a trained graph-to-text generator that produces more natural free-form sentences. The authors find that naive linear verbalization performs comparably to or better than the free-form alternative, which they read as evidence that their simple approach is sufficient. We read this is a different perspective: linear verbalization maximizes lexical overlap between the triple string and the query, which is precisely the shortcut grammar bias exploits — a retriever performing well under this condition offers no evidence that it encodes relational structure rather than surface-level matching. Neither this ablation nor the broader KAPING pipeline evaluates robustness to argument order or triple corruption.

A similar study extends the KAPING paradigm to RAG [3] by adding a question decomposition module, while maintaining the original text-triple mapping. It uses standard sentence embeddings applied to linearly verbalized triples and presents its no-decomposition baseline as equivalent to KAPING. The encoder itself is never audited for directional or lexical robustness — precisely the mapping step our work targets directly.

Closest to our diagnostic is Le Scao and Gardent [7], who fine-tune a dense bi-encoder built on *all-mpnet-base-v2* to align RDF graphs and text, targeting referenceless natural language generation evaluation in English rather than retrieval per se. In an ablation restricted to non-symmetrical WebNLG relations, their pretrained model misclassified 38% of argument-inverted triples as equivalent to the original — evidence that dense retrievers can be insensitive to relational direction. Adding inversion-corrupted negatives during pretraining reduces this to 14%, but the fix operates within the contrastive objective via negative resampling, and the test itself remains a single curated ablation rather than a systematic diagnostic. Our work differs by treating directional sensitivity as the central, systematic measurement — contrasted against entity- and relation-swap corruption — and by correcting it through a separate auxiliary objective (masked-slot prediction) rather than negative resampling on a sparse encoder evaluated cross-lingually with an explicit seen/unseen generalization test. Their result stands as independent prior evidence that the phenomenon we formalize as grammar bias is real, not an artifact of our pipeline.

Our own prior work [4] first investigated text-triple alignment via supervised fine-tuning of pretrained sentence encoders, using contrastive and multiple-negatives-symmetric-ranking losses over WebNLG-derived pairs, without any auxiliary structural objective. That study showed that fine-tuning on richer, augmented data (PoRTA dataset [8]) improves text-to-triple retrieval over the original WebNLG-PT baseline [9], but the triple-to-text direction remains harder in both datasets regardless of data volume or paraphrase diversity. This indicates that the bottleneck lies in representation, not data scale. Because the encoder is dense throughout, structural signal (subject, predicate, object roles) and lexical signal compete for the same dimensions, and no explicit mechanism forces the model to learn relational asymmetry (e.g., that a director is never the film). This finding motivated our present study. Rather than adding more paraphrased data to a dense space, we change the representation itself: sparse, SPLADE-style embeddings [6] combined with Masked Triple Modeling (MTM) — our original proposal, an auxiliary objective that directly teaches relational structure independent of surface wording.

3. Problem Definition

We formalize the setting in which grammar bias arises, define the two retrieval directions used throughout the paper, and introduce notation for the two families of encoders considered in our approach.

3.1. Knowledge Graphs and Triples

A knowledge graph is a tuple $KG = \{E, R, T\}$, where E is a set of entities, R is a set of relations, and T is a set of triples[10]. Each triple $t \in T$ is defined as

$$t = (e_h, r, e_t), \quad e_h, e_t \in E,\ r \in R, \tag{1}$$

where e_h and e_t denote the head and tail entities and r the relation connecting them. Each triple is associated with a verbalization $v(t)$, a natural language sentence conveying the same fact in surface form (e.g., $t = (\textit{StarWars}, \textit{directedBy}, \textit{GeorgeLucas})$ verbalizes to $v(t) =$ "*Star Wars was directed by George Lucas*").

3.2. Bidirectional Text-Triple Retrieval

We treat text-triple alignment as a bidirectional retrieval problem across a shared representation space. Given an encoder ϕ that maps both triples and texts into that space, and a similarity function $\text{sim}(\cdot, \cdot)$, we define two retrieval directions:

- **Triple-to-text retrieval:** given a query triple t, retrieve $\hat{v} = \arg\max_{v \in \mathcal{V}} \text{sim}(\phi(t), \phi(v))$ from a candidate pool of texts $\mathcal{V}$;
- **Text-to-triple retrieval:** given a query text $v(t)$, retrieve $\hat{t} = \arg\max_{t' \in \mathcal{T}} \text{sim}(\phi(v(t)), \phi(t'))$ from a candidate pool of triples $\mathcal{T}$.

A properly aligned representation space should support both directions with comparable accuracy, since both express the same underlying fact.

3.3. Grammar Bias

We define *grammar bias* as an encoder's reliance on lexical overlap between a triple and its verbalization, rather than on the relational structure that determines the triple's meaning. This bias is measured by testing whether an encoder can distinguish a triple from a structurally corrupted counterpart that shares identical tokens:

Given a corrupted triple $t^- = (e_t, r, e_h)$, obtained by swapping the head and tail of t, a directionally robust ϕ should satisfy $\text{sim}(\phi(t), \phi(v(t))) \gg \text{sim}(\phi(t^-), \phi(v(t)))$, even though t and t^- share identical tokens and differ only in slot order. An encoder that fails this test, scoring close to chance, exhibits grammar bias regardless of its performance on standard, uncorrupted retrieval.

A grammar bias can also manifest as asymmetry between two retrieval directions. Let $\text{Acc}_{t \to v}(\phi)$ and $\text{Acc}_{v \to t}(\phi)$ denote retrieval accuracy (e.g., NDCG@1) for the triple-to-text and text-to-triple directions, respectively, under encoder ϕ. An encoder ϕ exhibits grammar bias when

$$\text{Acc}_{v \to t}(\phi) \gg \text{Acc}_{t \to v}(\phi). \tag{2}$$

This asymmetry occurs because the tokens of t typically appear verbatim in $v(t)$: text-to-triple retrieval can therefore be driven by lexical overlap between the query text and the candidate triples' surface tokens, without ϕ needing to represent the relational structure (e_h, r, e_t) explicitly. Triple-to-text retrieval offers no such shortcut. A query triple t is compact and symbolic, sharing little surface vocabulary with its candidate verbalizations in $\mathcal{V}$, so an encoder can only retrieve $v(t)$ correctly if it has captured the relation between e_h and e_t independently of grammar. Grammar bias is not just a performance gap; it indicates that ϕ encodes lexical co-occurrence rather than the relational structure of t.

3.4. Encoder Families

We distinguish two families of encoders over the shared representation space, differing in how they distribute information throughout dimensions.

Dense encoders. A dense encoder [11] $\phi_{\text{dense}} : X \to \mathbb{R}^d$ maps an input x (a text or a triple) to a low-dimensional, real-valued vector in $\mathbb{R}^d$, typically $d \in [256, 1024]$, in which every dimension is generally non-zero and no dimension has an intrinsic, human-interpretable meaning. Cosine similarity is used to calculate similarity in this dense space. In this context, all signals in the input—such as linguistic structure, entity identity, and relational structure—compete for the same dimensions. As a result, dense encoders often allow the strongest and most consistent signal, which is lexical overlap, to dominate the learned representation.

Sparse encoders. Sparse encoders map dense embeddings into a higher-dimensional space, where each dimension corresponds to a token in the model's vocabulary [6].

A sparse encoder $\phi_{\text{sparse}} : X \to \mathbb{R}_{\geq 0}^{|\mathscr{V}_{\text{model}}|}$, following the SPLADE paradigm [12], maps an input to a high-dimensional, non-negative vector indexed by the encoder's vocabulary $\mathscr{V}_{\text{model}}$, where each dimension j corresponds to the activation of vocabulary token j and most dimensions are zero. Activations are tied to individual vocabulary entries instead of being distributed throughout an opaque latent space. This allows distinct relations (e.g., *directedBy* versus *bornIn*) to activate separate sets of dimensions rather than becoming mixed with the grammatical tokens of a specific verbalization.

We hypothesize that this property makes sparse encoders better suited to satisfy the directional robustness condition defined above, and we build on ϕ_{sparse} as the basis for our approach in Section 4.2.

4. Countering Grammar Bias with Masked Triple Modeling (MTM)

The sparse encoder backbone provides the foundation for retrieval, but it does not ensure that relational structure, rather than surface lexical overlap, influences the learned representations. We introduce Masked Triple Modeling (MTM), a self-supervised objective that forces the encoder to reconstruct a masked slot in a triple using the other two elements. This is trained alongside a standard text-triple alignment loss. Figure 1 outlines the training pipeline: a triple and its verbalization are encoded by the shared sparse backbone; the triple branch includes slot masking and an MTM prediction head; and the two losses are combined using a weighting coefficient λ. The following subsections will detail each component of this pipeline.

4.1. Sparse encoder backbone

This structure is conceptually similar to classical term-weighting methods such as BM25 [13], which also represent text as sparse, vocabulary-indexed vectors; the key difference is that BM25 weights are computed from term statistics (frequency and inverse document frequency), whereas sparse neural encoders learn contextualized term weights via a masked language model backbone, capturing semantic relevance rather than surface term frequency. This inherited structure means sparse encoders remain compatible with the inverted-index infrastructure long used for BM25-based retrieval, while replacing purely statistical weighting with a learned, semantically aware one.

For a backbone such as `multilingual-e5-base` [14], this expands the representation from a typical 768-dimensional dense vector to a vocabulary-sized space of approximately 250,000 dimensions. Rather than a drawback, this expansion can enable a reduction in storage footprint. Unlike dense vectors, where every dimension carries a nonzero value, sparse encoders are trained with an L1/FLOPS regularization that pushes the vast majority of dimensions to exactly zero, so only the small subset of active (nonzero) token weights needs to be stored or indexed. The larger nominal dimensionality is therefore not a cost in practice, since sparsity – not the raw dimensionality – determines the effective storage requirement.

The token-aligned structure opens a direct path to neurosymbolic alignment. Given that each active dimension corresponds to a specific vocabulary token rather than an entangled position in a compressed latent space, the components of a triple (s, p, o) can be traced to their own activated tokens, rather than

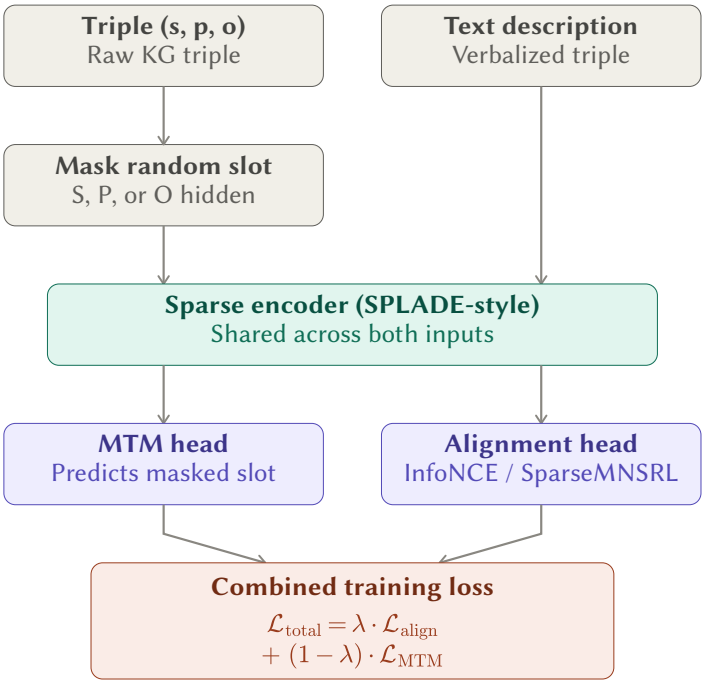


Figure 1: Overview of the MTM training pipeline. A triple and its verbalized text form are encoded by a shared sparse encoder (SPLADE-style); the triple branch is corrupted by masking one slot (subject, predicate, or object) before encoding. The resulting representations support two heads: an MTM head that reconstructs the masked slot, and an alignment head that scores text-triple correspondence via SparseMNSRL. The two objectives are combined into a single training loss, weighted by λ.

being diffused across an uninterpretable dense space. This gives sparse representations a degree of semantic explainability that dense embeddings lack, offering a potential bridge between symbolic KG structure and language-model representations without sacrificing the interpretability that is lost when using dense encoders.

4.2. MTM objetive

Prior work [4], which addresses text-triple alignment in both retrieval directions, shows that fine-tuning encoders on triple-text pairs tends to improve retrieval performance. However, this improvement alone does not confirm that the encoder has learned a correct semantic alignment between triples and text: as discussed in Section 6.1, retrieval gains that stem primarily from surface-level token overlap can reinforce grammar bias rather than genuine relational understanding, and remain invisible under standard, direction-agnostic evaluation.

To address this, we introduce an auxiliary unsupervised task in addition to the text-triple alignment objective. We take inspiration from BERT's Masked Language Model (MLM) task [15] and adapt it to the structure of RDF triples. Instead of masking tokens within a sentence, we mask one element of the triple – subject, predicate, or object – and train the encoder to recover it from the remaining two. We refer to this task as Masked Triple Modeling (MTM), illustrated in Figure 2. By requiring the encoder to reconstruct a missing slot from relational context alone, MTM encourages it to treat triple elements as structured, interdependent components rather than as ordinary vocabulary tokens, offering a path toward the neurosymbolic alignment.

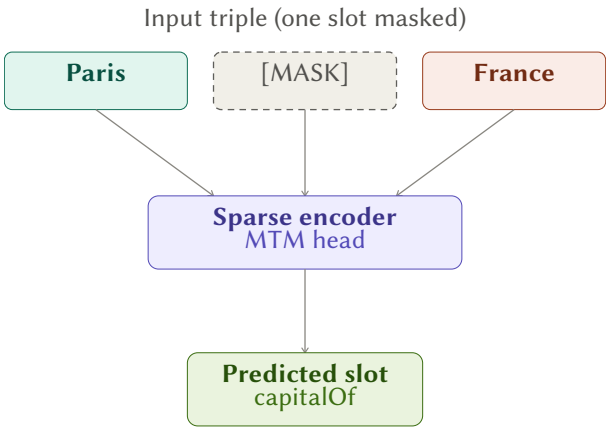


Figure 2: Masked Triple Modeling: one triple slot is masked (here, the predicate), and the sparse encoder's MTM head predicts it from the remaining subject and object.

4.3. Combined training

As we introduce an additional learning task into the encoder, we must combine it with the primary triple-text alignment objective while preserving both signals during training. We formalize the total training loss as a weighted sum:

$$\mathscr{L}_{\text{total}} = \lambda \mathscr{L}_{\text{align}} + (1 - \lambda) \mathscr{L}_{\text{MTM}} \tag{3}$$

where $\mathscr{L}_{\text{align}}$ is the Multiple Negatives Symmetric Ranking Loss (MNSRL), implemented in the Sentence Transformers library [2], used to align text and triple representations, and $\mathscr{L}_{\text{MTM}}$ is the masked-slot prediction loss, computed as cross-entropy over the vocabulary distribution for the masked triple element, analogous to BERT's MLM objective [15] but applied to a single structured slot rather than a full token sequence. The weighting factor $\lambda \in [0, 1]$ controls the relative contribution of each objective: higher values emphasize alignment quality, while lower values emphasize relational structure learning through slot recovery.

Because λ directly trades off between the two objectives, its value is not fixed a priori; instead, we treat it as a hyperparameter and empirically identify an operating point via a sweep over $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, reported in Section 6. This choice is not just a minor adjustment; placing too much emphasis on either loss can undermine the other objective completely instead of creating a balanced trade-off, as discussed in Section 7.

4.4. Corruption strategy

Complementary to MTM's masked-slot prediction, we introduce three corruption types applied to a valid triple, illustrated in Figure 3. Entity swap replaces the head or the tail, chosen with equal probability, with an entity drawn uniformly at random from the full entity vocabulary — the set of all heads and tails observed in the dataset — excluding the original head and tail values. Relation swap replaces the predicate with a relation drawn uniformly at random from the full relation vocabulary, excluding the original predicate; neither corruption enforces type or domain compatibility between the substituted element and its context. Direction flip exchanges subject and object while keeping the predicate fixed, preserving all surface tokens but reversing their relational role. Only direction flip is used during training, optionally added as a document-side hard negative in the symmetric InfoNCE alignment objective (Section 4). Entity swap and relation swap are reserved for the directional-robustness diagnostic (Section 6.1) and are never seen during training.

[2] https://sbert.net/docs/package_reference/sentence_transformer/losses.html

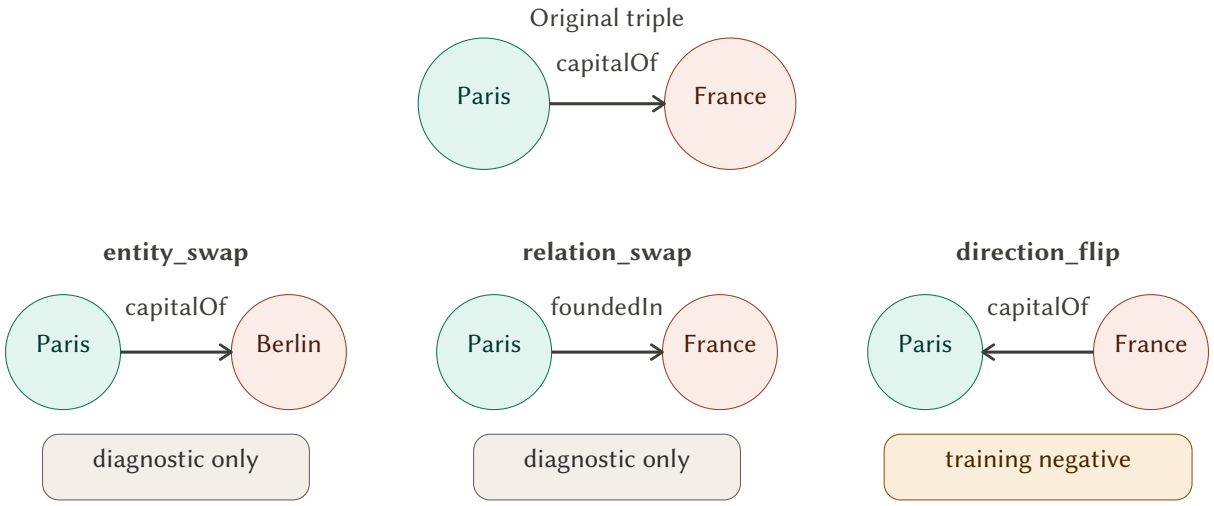


Figure 3: The three corruption types derived from a valid triple (Paris, capitalOf, France). Entity swap and relation swap are used exclusively in the directional-robustness diagnostic; direction flip is optionally used as a hard negative during training.

5. Experimental Methodology

This section describes the experimental configuration to evaluate MTM. We detail the dataset and its splits (cf. Subsection 5.1), the baseline models against which MTM is compared (cf. Subsection 5.2), the evaluation metrics (cf. Subsection 5.3), and the training configuration (cf. Subsection 5.4).

5.1. Dataset

We evaluate MTM on two Portuguese benchmarks, chosen to balance external validity against continuity with our own research line. WebNLG-PT is included because WebNLG is among the most widely adopted benchmarks linking knowledge graph triples to natural language text [9], giving us a point of comparison external to our own data pipeline. PoRTA [8] is our own benchmark, developed to address the WebNLG-PT limitations identified in our earlier work on this task [4]; including it lets us evaluate MTM within the same line of research that motivated its design, using Portuguese verbalizations produced under a controlled generation and curation pipeline rather than machine translation.

WEBNLG-PT: is a translated Portuguese version of WebNLG-v3 [9], covering ten classes extracted from DBpedia. It was generated by machine-translating the original English text-triple pairs via Google Translate and filtering for one-to-many triple-text alignments, resulting in 483 unique properties.

Portuguese RDF-Text Alignment (PoRTA) [8] is a Portuguese benchmark that covers both directional retrieval and triple-to-text generation, inspired by the WebNLG format. It selects 10 classes from DBpedia, resulting in 9 final classes after filtering out those without sufficient associated properties, and uses Llama 3.3 70B for triple verbalization, followed by LLM-based curation and paraphrase augmentation.

Table 1 shows the dataset distribution for Train, Dev, and Test for both WEBNLG-PT and PoRTA used in this research.

Table 1
Benchmark size comparison: WEBNLG-PT (translated WebNLG-v3) and PoRTA (ours). Top values indicate triple-text pairs; values in parentheses indicate unique triples.

Dataset	Train	Dev	Test
WEBNLG-PT	7686 (3115)	961 (402)	1418 (521)
PoRTA (ours)	18 380 (3676)	4600 (920)	5775 (1155)

5.2. Baseline

We compare MTM against three reference configurations. **Dense MNSRL** is a dense bi-encoder trained with symmetric InfoNCE (Multiple Negatives Symmetric Ranking Loss), representing the standard triple-text alignment approach used in prior work [4, 8]. **Sparse MNSRL** uses the same alignment loss but replaces the dense backbone with a SPLADE-style sparse encoder, without the MTM auxiliary objective, isolating the effect of sparsity alone. **Sparse Triplet+MTM** is an earlier multitask configuration combining a triplet loss with MTM; we report it as a negative reference rather than a competitive baseline, since it collapses retrieval performance entirely (Section 6) despite reducing directional bias, and motivated our subsequent switch to a symmetric InfoNCE-based alignment loss.

5.3. Evaluation Metrics

We evaluate models along two axes: retrieval quality and directional robustness.

Retrieval quality is measured with NDCG@1, computed separately for the text-to-triple and triple-to-text directions, following the evaluation protocol of [8].

Directional robustness is measured via a perturbation-based diagnostic (Section 4.4): for each query we construct three corrupted candidates – *entity_swap*, *relation_swap*, and *direction_flip* – and report hit@1 and margin (the score gap between the valid triple and its corrupted counterpart) for each corruption type. Unlike NDCG, which reflects the model's ability to identify the best candidate among many, hit@1 and margin under direction_flip isolate whether the model relies on relational structure rather than lexical overlap, since direction_flip preserves surface tokens while inverting head/tail order.

5.4. Training configurations

All models use multilingual-e5-base [14] as the backbone encoder. Models are trained for 5 epochs with batch size 16 and gradient accumulation of 1, on a single RTX A5500 GPU. For MTM-augmented models, we perform a sweep over the alignment-loss weight $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, where higher λ assigns more weight to the alignment loss and less to the masked-slot prediction loss (Section 4). Because both objectives share the same encoder and only their relative weighting changes across runs, the sweep isolates the contribution of each loss to retrieval quality and directional robustness without confounding the comparison with architectural differences between configurations.

6. Experimental Results

This section reports results from the experimental methodology described in Section 5, organized around four aspects of MTM's behavior: the full configuration space spanning baselines, MTM without directional negatives, and MTM with directional negatives (cf. Subsection 6.1); and results regarding the MTM sensitivity to the alignment–structure weighting λ (cf. Subsection 6.2).

6.1. Baselines and the Full Ablation Space - WEBNLG-PT

Table 2 reports IR NDCG@1, flip_hit@1, and flip_margin across thirteen configurations on webnlg_pt. The dense MNSRL baseline reaches the highest IR NDCG@1 (0.966) but the lowest flip_margin among the well-behaved configurations (+0.003).

Sparse MNSRL without MTM scores lower on both retrieval (0.800) and robustness (flip_hit@1 0.493, flip_margin +0.004). The legacy Sparse Triplet+MTM configuration reaches the highest flip_hit@1 (0.896) and flip_margin (+1.945) of any configuration, but its IR NDCG@1 falls to 0.013. Among the InfoNCE configurations without flip negatives, IR NDCG@1 ranges from 0.804 (λ=0.1) to 0.870 (λ=0.9), while flip_hit@1 ranges from 0.590 (λ=0.9) to 0.717 (λ=0.1).

Adding direction_flip hard negatives raises flip_hit@1 across all four tested λ values (0.944–0.952) and flip_margin (+0.695 to +0.896), while IR NDCG@1 in this group ranges from 0.541 (λ=0.1-flip) to 0.630 (λ=0.5-flip). InfoNCE λ=0.5-flip attains the highest flip_margin (+0.896) among all flip configurations.

Table 2
Retrieval and directional-robustness metrics across the full configuration space on webnlg_pt. IR NDCG@1 is measured on the harder triple-to-text direction; flip_hit@1 and flip_margin measure robustness to `direction_flip` corruption.

Model	MTM	Flip neg.	λ	IR NDCG@1	flip_hit@1 / flip_margin
Reference baselines					
Dense MNSRL	✗	✗	–	**0.966**	0.627 / +0.003
Sparse MNSRL	✗	✗	–	0.800	0.493 / +0.004
Sparse Triplet+MTM	✓	✗	0.5	0.013	0.896 / +1.945
InfoNCE + MTM, no flip negatives					
λ=0.1	✓	✗	0.1	0.804	0.717 / +0.043
λ=0.3	✓	✗	0.3	0.831	0.663 / +0.025
λ=0.5	✓	✗	0.5	0.864	0.657 / +0.017
λ=0.7	✓	✗	0.7	0.856	0.645 / +0.021
λ=0.9	✓	✗	0.9	0.870	0.590 / +0.016
InfoNCE + MTM + direction_flip hard negatives					
λ=0.1-flip	✓	✓	0.1	0.541	0.944 / +0.695
λ=0.3-flip	✓	✓	0.3	0.589	**0.952** / +0.708
λ=0.5-flip†	✓	✓	0.5	0.630	0.950 / +**0.896**
λ=0.7-flip	✓	✓	0.7	0.557	0.951 / +0.865

† Pareto-optimal configuration.

6.2. Weighting Alignment against Structure: The λ Sweep - PoRTA

Table 3 reports the λ sweep on PoRTA across five values ($\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$), alongside the dense baseline and a sparse MNSRL configuration without MTM. The dense baseline scores highest on entity_swap (0.999), relation_swap (0.995), and overall hit@1 (0.849), and reaches NDCG@1 of 0.986/0.987 in the triple-to-text and text-to-triple directions, respectively. On direction_flip, the dense baseline's hit@1 (0.553) and margin (0.0012) are close to those of a random classifier. Sparse MNSRL without MTM scores lower than the dense baseline on every corruption type (direction_flip hit@1 0.393, relation_swap 0.675, entity_swap 0.958) and on bidirectional NDCG@1 (0.601–0.602).

Across the λ sweep, direction_flip hit@1 peaks at λ=0.3 (0.634) and declines at higher λ values, reaching 0.360 at λ=0.9. Direction_flip margin follows the same pattern, peaking at λ=0.3 (0.0522) and falling to 0.0025 at λ=0.9. NDCG@1 in both retrieval directions also peaks at λ=0.3 (0.784 triple-to-text, 0.780 text-to-triple) and declines at higher λ, reaching 0.466 and 0.481 at λ=0.9. Relation_swap and entity_swap hit@1 remain above 0.62 and 0.94, respectively, across all five λ values.

Table 3
λ sweep on PoRTA, evaluated against three corruption strategies (`entity_swap`, `relation_swap`, `direction_flip`) and bidirectional retrieval NDCG@1. Cells report hit@1 / margin for direction_flip and NDCG@1 (triple→text / text→triple) for retrieval. *Dense baseline NDCG@1 may be inflated by a small evaluation pool

Run	dir_flip (hit@1 / margin)	rel_swap hit@1	ent_swap hit@1	overall hit@1	NDCG@1 (t→x / x→t)
Dense (baseline)	0.553 / 0.0012	**0.995**	**0.999**	**0.849**	0.986* / **0.987***
Sparse-MNSRL (no MTM)	0.393 / 0.0059	0.675	0.958	0.675	0.602 / 0.601
λ=0.1	0.563 / 0.0164	0.736	0.981	0.762	0.817 / 0.841
λ=0.3	**0.634** / **0.0522**	0.707	0.976	0.774	0.784 / 0.780
λ=0.5	0.457 / 0.0188	0.646	0.955	0.686	0.517 / 0.526
λ=0.7	0.394 / 0.0113	0.668	0.958	0.677	0.523 / 0.536
λ=0.9	0.360 / 0.0025	0.624	0.946	0.641	0.466 / 0.481

7. Discussion

We organize this discussion in two stages, following the same progression as the experiments themselves: webnlg_pt first, as the dataset where the full configuration space was ablated end-to-end (Table 2), and PoRTA second, as the higher-fidelity dataset where the λ sweep was evaluated against three independent corruption strategies (Table 3).

In the webnlg_pt analysis, the ablation identifies the source of directional robustness. The dense MNSRL baseline achieves the highest IR NDCG@1 (0.966) in Table 2 but the lowest flip_margin among the well-performing configurations (+0.003). This aligns with the grammar bias identified in our previous work [4]. The model performs well due to the overlap of triple tokens with the source text, not because it captures relational structure. The introduction of MTM changes this scenario, but not consistently. Without direction_flip hard negatives, InfoNCE improves flip_hit@1 compared to the sparse baseline (up to 0.717 at λ=0.1) while retrieval remains within the sparse baseline's range (0.804–0.870). When direction_flip hard negatives are incorporated into training, flip_hit@1 rises across all four tested λ values (0.944–0.952), but IR NDCG@1 declines to a lower range (0.541–0.630) than both the dense baseline and the no-flip configurations. The legacy Sparse Triplet+MTM configuration exemplifies this trend: it achieves the highest flip_hit@1 and flip_margin of any tested configuration, but its IR NDCG@1 falls to 0.013 — two orders of magnitude below the dense baseline. We treat this specific result with caution, since a model that fails to retrieve the correct candidate from a realistic pool cannot be described as directionally robust, regardless of its flip metrics in isolation.

This raises a question that webnlg_pt alone cannot resolve: is the retrieval collapse under Sparse Triplet+MTM specific to that loss, or is it an unavoidable outcome of training MTM alongside any sparse contrastive objective? The λ sweep on PoRTA (Table 3) addresses this with a revised training objective (SparseMNSRL instead of SparseTripletLoss) and a larger, more linguistically diverse dataset. None of the five λ-swept configurations approach the webnlg_pt collapse. IR NDCG@1 across the sweep ranges from 0.466 to 0.784, never falling below 0.4. This supports the notion that the retrieval collapse stems from the training objective rather than in MTM itself.

Within this corrected setup, λ=0.3 stands out as the configuration that best balances both objectives. It slightly exceeds the dense baseline's direction_flip hit@1 (0.634 vs. 0.553). The retrieval performance (NDCG@1) declines from 0.986/0.987* to 0.784/0.780*, which represents a real but moderate cost, and is much smaller than the joint collapse seen at higher λ values. For $\lambda \geq 0.5$, this trade-off is lost: both direction_flip hit@1 and NDCG@1 decrease sharply, reaching 0.360 and 0.466/0.481 respectively at $\lambda = 0.9$.

The entity_swap and relation_swap columns in Table 3 clarify the source of this robustness. The dense baseline achieves scores of 0.999 and 0.995 on these two perturbations, which change surface tokens that the model identifies through lexical overlap. In constrast, on direction_flip — where every token remains intact while only the head/tail order is altered — the same baseline scores 0.553, indistinguishable from chance. This gap between token-level and order-level perturbations is present across all configurations in Table 3, not only the dense baseline, though it narrows considerably under MTM at λ=0.3.

Finally, it is worth noting that the dense baseline achieves the highest overall NDCG@1 across both datasets, even though it suffers from the grammar bias discussed throughout this work. While the baseline performs well on global NDCG, this performance is tied to the grammar bias issue. Its poor results on direction flips indicate that it does not capture true relational associations during training, reinforcing our findings about the underlying problem.

8. Conclusion

Prior work identified grammar bias as an obstacle to knowledge-grounded retrieval. Dense encoders exploit lexical overlap between verbalized triples and their source text rather than the triples' relational structure. Enriching the training data alone did not resolve this gap; a larger, more varied benchmark improved alignment without closing the directional asymmetry between text-to-triple and triple-to-text

retrieval [8]. Our present study investigated whether the gap is instead a representation problem, addressed through sparse embeddings trained with an explicit structural objective, namely MTM. Our results support this direction. On PoRTA, replacing SparseTripletLoss with SparseMNSRL removed the retrieval collapse seen in our earlier multitask configuration, and λ=0.3 improved direction_flip hit@1 slightly exceeds the dense baseline (0.634 vs 0.553), with a moderate retrieval cost (NDCG@1 0.784/0.780 vs 0.986/0.987). This is a much better trade-off than the collapse seen for higher λ values. The gap between token-level corruptions, which the dense baseline already handles well, and direction_flip, which requires structural sensitivity, offers a reusable diagnostic for isolating grammar bias in future triple-text alignment work. These results reduce the gap identified in our previous studies rather than eliminate it: IR NDCG@1 under MTM still does not meet the dense baseline, generalization to unseen triples needs further assessment, and the relationship between λ and the alignment-structure trade-off remains unclear. Future work will focus on completing the seen/unseen evaluation, testing robustness under a corruption strategy different from training, and normalizing the two loss terms to determine if λ serves as a genuine trade-off knob. We also plan to use the encoder model in a KG-RAG end-to-end pipeline to uncover the benefits of our vector representations.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. This work was also supported by CNPq, Brazil (grant #301337/2025-0).

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) and Grammarly in order to perform grammar and spelling checks. The authors also used Claude to assist in writing the SVG code used to render Figures 1 and 2, whose design and content were specified by the authors. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. Baek, A. F. Aji, J. Lehmann, S. J. Hwang, Direct fact retrieval from knowledge graphs without entity linking, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 10038–10055. URL: https://aclanthology.org/2023.acl-long.558/. doi:10.18653/v1/2023.acl-long.558.

[2] J. Baek, A. F. Aji, A. Saffari, Knowledge-augmented language model prompting for zero-shot knowledge graph question answering, in: E. Hruschka, T. Mitchell, S. Rahman, D. Mladenić, M. Grobelnik (Eds.), Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023), Association for Computational Linguistics, Toronto, ON, Canada, 2023, pp. 70–98. URL: https://aclanthology.org/2023.matching-1.7/. doi:10.18653/v1/2023.matching-1.7.

[3] J. Linders, J. M. Tomczak, Knowledge graph-extended retrieval augmented generation for question answering, Applied Intelligence 55 (2025) 1102. URL: https://doi.org/10.1007/s10489-025-06885-5. doi:10.1007/s10489-025-06885-5.

[4] V. J. S. Chico, J. Reis, Learning knowledge representation by aligning text and triples via finetuned pretrained language models, in: Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 2: KEOD, INSTICC, SciTePress, 2024, pp. 51–62. doi:10.5220/0013015100003838.

[5] R. T. McCoy, E. Pavlick, T. Linzen, Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3428–3448. URL: https://aclanthology.org/P19-1334/. doi:10.18653/v1/P19-1334.

[6] T. Formal, B. Piwowarski, S. Clinchant, Splade: Sparse lexical and expansion model for first stage ranking, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 2288–2292. URL: https://doi.org/10.1145/3404835.3463098. doi:10.1145/3404835.3463098.

[7] T. Le Scao, C. Gardent, Joint representations of text and knowledge graphs for retrieval and evaluation, in: J. C. Park, Y. Arase, B. Hu, W. Lu, D. Wijaya, A. Purwarianti, A. A. Krisnadhi (Eds.), Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings), Association for Computational Linguistics, Nusa Dua, Bali, 2023, pp. 110–122. URL: https://aclanthology.org/2023.findings-ijcnlp.10/. doi:10.18653/v1/2023.findings-ijcnlp.10.

[8] V. J. S. Chico, A. G. Regino, J. C. dos Reis, Bench4t3: A framework to create benchmarks for text-to-triples alignment generation, Journal of the Brazilian Computer Society 32 (2026) 85–101. URL: https://journals-sol.sbc.org.br/index.php/jbcs/article/view/5809. doi:10.5753/jbcs.2026.5809.

[9] C. Gardent, A. Shimorina, S. Narayan, L. Perez-Beltrachini, The WebNLG challenge: Generating text from RDF data, in: J. M. Alonso, A. Bugarín, E. Reiter (Eds.), Proceedings of the 10th International Conference on Natural Language Generation, Association for Computational Linguistics, Santiago de Compostela, Spain, 2017, pp. 124–133. URL: https://aclanthology.org/W17-3518/. doi:10.18653/v1/W17-3518.

[10] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Comput. Surv. 54 (2021). URL: https://doi.org/10.1145/3447772. doi:10.1145/3447772.

[11] W. X. Zhao, J. Liu, R. Ren, J.-R. Wen, Dense text retrieval based on pretrained language models: A survey, ACM Trans. Inf. Syst. 42 (2024). URL: https://doi.org/10.1145/3637870. doi:10.1145/3637870.

[12] T. Formal, B. Piwowarski, S. Clinchant, Splade: Sparse lexical and expansion model for first stage ranking, 2021. URL: https://arxiv.org/abs/2107.05720. arXiv:2107.05720.

[13] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Found. Trends Inf. Retr. 3 (2009) 333–389. URL: https://doi.org/10.1561/1500000019. doi:10.1561/1500000019.

[14] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual e5 text embeddings: A technical report, 2024. URL: https://arxiv.org/abs/2402.05672. arXiv:2402.05672.

[15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/. doi:10.18653/v1/N19-1423.